

Institute for Logic, Language and Computation
University of Amsterdam, 1992

Proceedings of the
Eighth Amsterdam Colloquium
December 17 — 20, 1991

Paul Dekker & Martin Stokhof (eds.)



Proceedings of the
Eighth Amsterdam Colloquium

Institute for Logic, Language and Computation
University of Amsterdam, 1992

Proceedings of the
Eighth Amsterdam Colloquium
December 17 — 20, 1991

Paul Dekker & Martin Stokhof (eds.)

Preface

The 1991 edition of the Amsterdam Colloquium was the eighth in a series which started in 1976. Originally an initiative of the Department of Philosophy, the colloquium is now organized by the Institute for Logic, Language and Computation (ILLC), in which the Departments of Mathematics and Computer Science, the Department of Philosophy and the Department of Computational Linguistics of the University of Amsterdam cooperate.

For the organization of the Eighth Amsterdam Colloquium additional financial support was received from the Foundation for Philosophical Research (SWON), which is subsidized by the Netherlands Organization for Scientific Research (NWO), from the Faculty of Arts (Faculteit der Letteren) and from the Centre for Mathematics and Computer Science (CWI), which is gratefully acknowledged.

Published here are 34 of the 40 contributed talks that were delivered at the Colloquium. They appear as the authors prepared them. The copyright resides with the individual authors.

Paul Dekker
Martin Stokhof
Amsterdam, April 1992

Institute for Language, Logic and Computation
Department of Philosophy
Nieuwe Doelenstraat 15
1012 CP Amsterdam, The Netherlands

Contents

Preface	v
Contents	vi
Pieter Adriaans	
A domain theory for categorial language learning algorithms	1
David Beaver	
The kinematics of presupposition	17
Filippo Beghelli	
Comparative quantifiers	37
Patrick Blackburn	
Tense, temporal reference and tense logic	57
Patrick Blackburn & Edith Spaan	
On the complexity of attribute value logics	73
Michael Böttner, Lin Liang & Patrick Suppes	
Comprehension grammars generated from machine learning of natural languages	93
Paul Dekker	
An update semantics for dynamic predicate logic	113
Jan van Eijck & Fer-Jan de Vries	
A sound and complete calculus for update logic	133
Frank van Eynde	
Towards a dynamic and compositional treatment of temporal expressions	153
Dov Gabbay & Ruth Kempson	
Natural language content: a proof-theoretic perspective A preliminary report	173
Jonathan Ginzburg	
A quasi-naïve semantics for interrogatives and its implications.	197
Herman Hendriks	
Lambek semantics	
Normalization, spurious ambiguity, partial deduction and proof nets	213
Petra Hendriks	
Subdeletion and the Lambek calculus	233
Mark Hepple	
Command and domain constraints in a categorial theory of binding	253
Zhisheng Huang & Peter van Emde Boas	
Belief dependence, revision and persistence	271
Pauline Jacobson	
Bach-Peters sentences in a variable-free semantics	283
Theo M.V. Janssen	
On properties of the Zielonka-Lambek calculus	303

Edward L. Keenan & Edward P. Stabler	
Language invariants	309
Jean-Yves Lerner & Manfred Pinkal	
Comparatives and nested quantification	329
Márta Maleczki	
Bare common nouns and their relation to the temporal constitution of events in Hungarian	347
J.-J.Ch. Meyer	
Free choice permissions and Ross's paradox: internal vs external non- determinism	367
Wilfried P.M. Meyer Viol	
Partial objects and DRT	381
Michael Moortgat	
Labelled deductive systems for categorial theorem proving	403
John Nerbonne	
Constraint-based semantics	425
Valeria de Paiva	
A dialectica model of the Lambek calculus	445
Jaroslav Peregrin	
The role of variables in the formal description of natural language	463
Tanya Reinhart	
Wh-in-situ: an apparent paradox	483
Uwe Reyle	
Dealing with ambiguities by underspecification: A first order calculus for unscoped representations	493
Mark Ryan	
Belief revision and ordered theory presentations	513
Kjell Johan Sæbø	
The generation of noun phrases: adequacy and anaphora	531
Roger Schwarzschild	
<i>Together</i> as a non-distributivity marker	547
Hans Smessaert	
Duality distribution with nominal and aspectual quantifiers in Dutch ...	567
Marten Trautwein	
On the complexity of feature structures in categorial unification grammar	587
Alessandro Zucchi	
Existential sentences and predication	601
Adresses of the authors	623

A Domain Theory for Categorial Language Learning Algorithms

Pieter Adriaans
Syllogic. B.V.

Abstract

We study the Lambek calculus from the perspective of semantic learning. In this context we identify some (as far as we can judge) comparatively new problems:

1. The traditional priority of syntax over semantics has to be relativized in a learning situation since we can deduce syntactic information from semantic data and vice versa.
2. We need a domain theory to deal with partial models of a Lambek calculus.
3. We must have a theory about the influence of intensional information on the learning process.

The current approaches to semantics for the Lambek calculus are not feasible in these respects. We present a definition of the relation between a language, its syntax and semantics in terms of adjoint functors between Lambek calculi enriched with Boolean operators. This approach enables us to study the interplay between semantic and syntactic information in a new way that may be of interest in a wide area of semantic research.

1 Introduction

In our research, motivated by problems in knowledge engineering and systems analysis, we are interested in semantical learning of categorial languages in a dialogical context. To give a formal analysis we introduced the notion of an abstract speaker:

Definition: An abstract speaker is a set $A = \langle D, L, S, E_P, O \rangle$ where:

- D is a dictionary,
- L is a language which is a subset of the Kleene closure D^* of D ,
- S is the set of true sentences of L ,
- E_P is an examples routine that produces an element of S according to the probability distribution P over L , i.e. the probability that $L(X) = s$ is $P(s)$ and O is an oracle routine that tells us whether or not a certain element of D^* is member of L and S .

Buskowsky and Penn (1989) described an algorithm to deduce a categorial grammar from linguistic data by unification. Their method presupposes that we know the syntactic structure of the expressions. For real life learning situations this is an unrealistic presupposition. In (Adriaans, 1991) we presented a class of clustering learning algorithms to overcome this drawback. We proved that it is possible to learn a grammar effectively under the following conditions:

- The language has a categorial grammar.
- Each syntactic rule has an example with a complexity and length that is logarithmic in the complexity of the whole grammar.
- The distribution P is close to the Solomonoff-Levin distribution (see: Li & Vitany, 1989).

We also argued that these conditions are quite natural and give an indication for the effective learnability of categorial grammars in real life environments. In this paper we intend to go somewhat deeper into the formal background of these learning algorithms and we will argue that we need to develop a special domain theory to understand the learning process properly. The formal system corresponding to a categorial grammar is the Lambek calculus. The most current version is system $\mathcal{L}$ with Application, Composition, Associativity, Lifting and Division as its basic rules (See: Moortgat: 1988). Van Benthem (1986, 1991) describes a semantics based on the lambda calculus. He observes that the lambda calculus is too rich, since many of its terms do not correspond to admissible type changes in $\mathcal{L}$. Buszkowski (1986) has shown that the Lambek calculus is complete for a semigroup semantics I that assigns to every atomic type a subset of a semigroup $\mathcal{M} = (M, \bullet)$. The categorial counterpart of a Lambek calculus is a so-called biclosed monoidal category (Lambek, 1988, Lambek & Scott, 1986). Lambek (1988) showed that such a syntactic category can be interpreted in any cartesian closed category with a distinguished object of truth values, preferably a topos. To analyse the learning process itself we need a version of the Lambek calculus that is enriched with Boolean operators: we will call this version the Lambek Calculus extended with Boolean operators $\mathcal{L}_E$. We can use the Boolean operators to perform the clustering operations with rules like:

$$\cup_l \frac{A \rightarrow C \quad B \rightarrow C}{A \cup B \rightarrow C} \text{ and } \cap_r \frac{A \rightarrow B \quad A \rightarrow C}{A \rightarrow B \cap C}$$

Roorda (1991) has shown that it is indeed possible to create a version of the Lambek calculus enriched with Boolean operators and has obtained completeness and decidability results. He also showed that $\mathcal{L}$ is faithfully embedded in $\mathcal{L}_E$.

2 The need for a domain theory

There are several motivations for developing a domain theory for learning:

1. If we want to learn a language we do not know its syntactic structure. In a learning situation however we get syntactic and semantic information at the same time. Sometimes we are able to derive semantic information from syntactic information and vice versa, e.g. if a sentence is true then it is also well formed. Therefore the traditional priority of syntax over semantics has to be relativised in a learning situation. We cannot work with a theory that assigns a constituent structure to a sentence before it assigns a semantics. If we want to learn the associative variant $\mathcal{L}$ of the Lambek calculus we have to use an associative semantic calculus. The current semantic approaches to semantics for the Lambek calculus are not feasible in such learning environments.
2. If we want to learn an interpretation of a calculus we need a domain theory to deal with partial models of this interpretation. This means a shift from models for $\mathcal{L}$ to

models for $\mathcal{L}_E$. An interpretation of $\mathcal{L}_E$ however is a much more general object than an interpretation of $\mathcal{L}$. It doesn't define a grammar, but describes the free monoid of the underlying alphabet. A functor from $\mathcal{L}$ to $\mathcal{L}_E$ forgets the structure of $\mathcal{L}$. We need to furnish $\mathcal{L}_E$ with an additional structure that 'remembers' the original interpretation of $\mathcal{L}$.

3. In the learning process there may be intentional information about the (semantic) domain to be learned, e.g.

Cardinality Constraints We may know in advance that our language is two valued.

Universal Laws Semantic category A is contained in B .

In some cases this intentional information enables us to derive local completeness results, e.g. a complete definition of a logical operator. In the traditional semantic proposals it is difficult to analyse the influence this kind of information has on the learning process.

We want to develop a domain theory that can cope with these three problems. For the moment we will not bother about the internal structure of $\mathcal{L}$ and $\mathcal{L}_E$. We will only suppose that they have a categorical definition.

2.1 Types and Functors

We begin with an example. Suppose we want to learn a function $f : S \rightarrow T$. We define two new functions the direct image $f_* : \mathcal{P}(S) \rightarrow \mathcal{P}(T)$ and the inverse image $f^{-1} : \mathcal{P}(T) \rightarrow \mathcal{P}(S)$ as:

$$f_*(S_0) = \{f(s) \mid s \in S_0\}$$

$$f^{-1}(U) = \{s \in S \mid f(s) \in U\}$$

Suppose S is a set of expressions and T is a set of types. We have

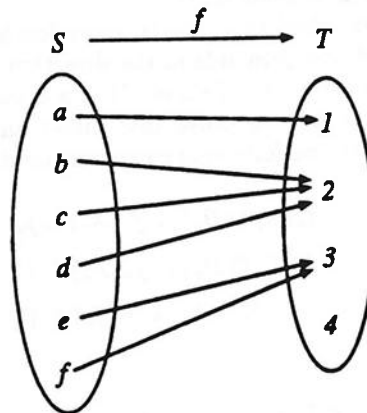


Figure 1: Example of a function.

$$f_*({c}) = \{2\}$$

$$f^{-1}(\{2\}) = \{b, c, d\}$$

$$f^{-1}(\{4\}) = \emptyset$$

and we have

$$f^{-1}f_*(\{c\}) = \{b, c, d\}$$

$$f^{-1}f_*(\{b\}) = \{b, c, d\}$$

$$f^{-1}f_*(\{a, b, c\}) = \{b, c, d\}$$

There is a one to one correspondence between f and f_* ; f_* is constructed out of f and f^{-1} uniquely determines f . For every $x \in S$ we have $f(x) = y \leftrightarrow f_*(\{x\}) = \{y\}$. There is also a one to one correspondence between f and f^{-1} ; f^{-1} is constructed out of f and f^{-1} uniquely determines f . For every $y \in T$ we have $f^{-1}(\{y\}) = X \leftrightarrow X = \{x | f(x) = y\}$. Consequently there is a one to one correspondence between f^{-1} and f_* . This partly explains the usefulness of these notions for functions learning. If we learn either f^{-1} or f_* then we can reconstruct the original f . There is also a more interesting correspondence:

Definition 1 Let S and T be two posets. We shall say that a pair (g, d) of functions $g : S \rightarrow T$ and $d : T \rightarrow S$ is a Galois connection or an Adjunction between S and T provided that:

1. both g and d are monotone and

2. the relations $g(t) \leq s$ and $t \leq d(s)$ are equivalent for all pairs of elements $(t, s) \in S \times T$.

In an adjunction (g, d) , the function g is called the left adjoint to d and d the right adjoint to g .

We give the following theorem:

Theorem 1 The pair (f_*, f^{-1}) defines a Galois correspondence between $\mathcal{P}(S)$ and $\mathcal{P}(T)$.

We won't give the proof here which is quite straightforward.¹ We have $f_*(s) \leq t$ is equivalent to $s \leq f^{-1}(t)$. The transformation $f^{-1}f_*$ is a closure operation that projects any set of expressions on the extension of their types.

In this example we have described this closure operation for a simple function. We are interested in a generalisation of this principle in the direction of the extended Lambek Calculus $\mathcal{L}_E$. As we have seen this calculus is hybrid. It is a classical boolean algebra, as well as a Lambek calculus. Now it is easy to prove that direct and inverse image preserve Boolean operations. This means that the following equations hold:

$$f_*(A_s \cup B_s) = f_*(A_s) \cup f_*(B_s)$$

$$f_*(A_s \cap B_s) = f_*(A_s) \cap f_*(B_s)$$

$$f_*(\neg A_s) = \neg(f_*(A_s))$$

$$f^{-1}(A_t \cup B_t) = f^{-1}(A_t) \cup f^{-1}(B_t)$$

$$f^{-1}(A_t \cap B_t) = f^{-1}(A_t) \cap f^{-1}(B_t)$$

$$f^{-1}(\neg A_t) = \neg(f^{-1}(A_t))$$

For any function we get this result for free. If the domain and the range of the function are concatenative structures we could also enforce the conservation of Lambek operators. Suppose that S and T are free monoids and that f_* and f^{-1} preserve concatenation and slashes:

$$f_*(A_s \circ B_s) = f_*(A_s) \circ f_*(B_s)$$

$$f_*(A_i \setminus B_i) = f_*(A_i) \setminus f_*(B_i)$$

$$f_*(A_i / B_i) = f_*(A_i) / f_*(B_i)$$

$$f^{-1}(A_i \bullet B_i) = f^{-1}(A_i) \bullet f^{-1}(B_i)$$

$$f^{-1}(A_i \setminus B_i) = f^{-1}(A_i) \setminus f^{-1}(B_i)$$

$$f^{-1}(A_i / B_i) = f^{-1}(A_i) / f^{-1}(B_i)$$

This approach gives us a scala of possibilities to describe the relation between a language and its syntax and semantics. It also gives us an understanding of the structure of partial models. A simple example will make this clear. Suppose we have a set of words of a certain language:

$$L = \{John, Derek, walks\}$$

There is also a set of (syntactic) types:

$$S = \{e, t\}$$

Of course these two sets do not describe a language. If we want to describe a language we could give an intensional description in the form of a dictionary function:

$$f(John) = e$$

$$f(Derek) = e$$

$$f(walks) = e \setminus t$$

This functions defines a language with two sentences. In a learning environment in general we do not know the dictionary. We have to reconstruct the types from examples we get. So we are interested in the exact relation between the intensional and the extensional representations of the types. We need a theory in which both notions of a types have a representation. This can be done along the following lines. We form two monoids:

$$Lang = (L, \bullet)$$

$$Syn = (S, \bullet)$$

And we form the powersets of these monoids.

$$Language = \mathcal{P}(Lang)$$

$$Syntaz = \mathcal{P}(Syn)$$

Now we have:

$$f_* : Language \rightarrow Syntaz$$

$$f^{-1} : Syntaz \rightarrow Language$$

We will suppose that these functions preserve the concatenative structure. Now we have a domain in which we can say anything we would like about our language. Of course we have the dictionary as a projection of the singletons:

$$f_*(\{John\}) = \{e\}$$

$$f_*(\{Derek\}) = \{e\}$$

$$f_*(\{walks\}) = \{e\} \setminus \{t\}$$

Concatenations of singletons give sentences:

$$f_*(\{John\} \bullet \{walks\}) = \{t\}$$

The inverse image of the singleton of sentence type defines the whole language:

$$f^{-1}(\{t\}) = \{John\} \bullet \{walks\} \cup \{Derek\} \bullet \{walks\}$$

This means that the closure of a singleton is the extension of its type:

$$f^{-1}f_*(\{John\}) = f^{-1}(\{e\}) = \{John, Derek\}$$

$$f^{-1}f_*(\{walks\}) = (f^{-1}(\{e\})) \setminus (f^{-1}(\{t\})) = \{walks\}$$

We even have a neat representation for ungrammatical structures:

$$f_*(\{John\} \bullet \{John\}) = \{e\} \bullet \{e\}$$

This illustrates how we can lift the dictionary function f to f^{-1} and f_* to get a domain description that gives us all the flexibility we need in learning algorithms.

2.2 What about semantics?

These ideas also enable us to express the exact relation between syntax and semantics from a learning perspective. The semantic function has from a mathematical point of view exactly the same structure as the syntactic function. One could speak of a semantic dictionary function. What is interesting is the interplay of these two structures. The semantic domain is an *intermediate layer* between the language and its syntax. An example will make this clear. Suppose we have the syntactic dictionary showed in figure2. A possible semantics for this

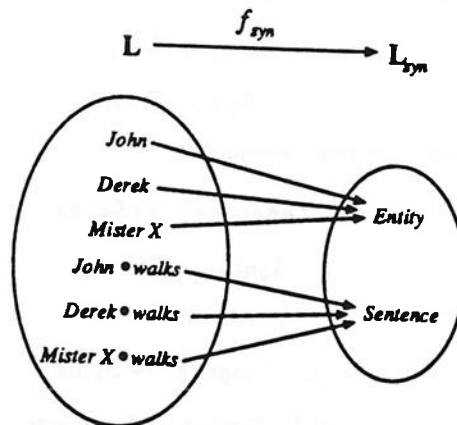


Figure 2: Example of a syntactic function

language is showed in figure3. One can proof that this function defines a semantics for this language is there is a function f_{dif} such that for all s in L we have $f_{syn}(s) = f_{dif} f_{sem}(s)$ as

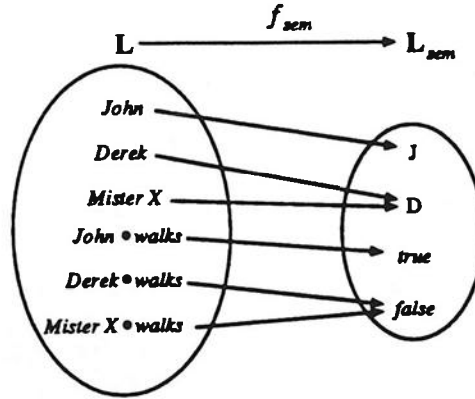


Figure 3: Example of a semantic function

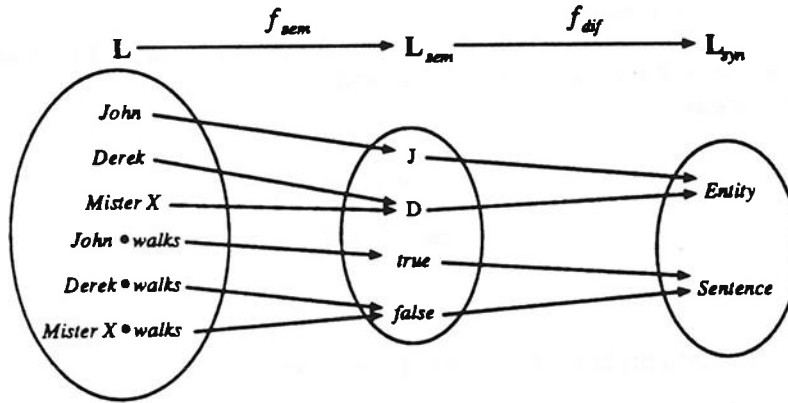


Figure 4: Example of a composition of a syntactical and a semantical function

shown in figure 4. This shows how the semantic calculus is an **intermediate layer** between language and syntax. The value of these ideas becomes clear if we lift the discussion to the complex set of interrelations between sets of expressions, sets of syntactic types and sets of semantic types defined by $f_{syn,*}$, $f_{sem,*}$, $f_{dif,*}$, f_{syn}^{-1} , f_{sem}^{-1} and f_{dif}^{-1} . These functions give us a wealth of possibilities to study the effects of partial information in language learning.

2.3 Monoidal Categories

It is possible to lift the whole discussion about the relation between syntax and semantics to the level of category theory. The categorial version of the Lambek Calculus is a so-called biclosed monoidal category. A biclosed monoidal category C is a syntactic calculus which satisfies the equations of a category as well as certain additional equations which assure the following conditions:

1)

$$\bullet : C \times C \rightarrow C$$

$$/ : C \times C^{op} \rightarrow C$$

$$\backslash : C^{op} \times C \rightarrow C$$

are bifunctors.

2) The categorial rules are natural isomorphisms.

3) The following composite arrows are identities:

$$A \bullet B \rightarrow (A \bullet I) \bullet B \rightarrow A \bullet (I \bullet B) \rightarrow A \bullet B$$

$$\begin{aligned} ((A \bullet B) \bullet C) \bullet D &\rightarrow (A \bullet B) \bullet (C \bullet D) \\ &\rightarrow A \bullet (B \bullet (C \bullet D)) \\ &\rightarrow A \bullet ((B \bullet C) \bullet D) \\ &\rightarrow (A \bullet (B \bullet C)) \bullet D \\ &\rightarrow ((A \bullet B) \bullet C) \bullet D \end{aligned}$$

The conditions under 3) are called coherence conditions. Cartesian Closed Categories are biclosed monoidal categories.²

If we want to extend this discussion to cover the notion of an extended Lambek calculus also we have to add functors and isomorphisms:

1) The bifunctors:

$$\begin{aligned} \& : C \times C \rightarrow C \\ \vee : C^{op} \times C^{op} &\rightarrow C^{op} \\ \Rightarrow : C^{op} \times C &\rightarrow C \end{aligned}$$

2) The functor:

$$\neg : C^{op} \rightarrow C$$

3) The natural isomorphisms for the categorial rules.

4) An initial object $\perp$ and a terminal object T .

This discussion shows that it is possible to give a categorial definition of $\mathcal{L}_E$.

3 A domain theory

We give an informal account of the theory. We begin with a preliminary example. Let TYPE be an alphabet of basic types and let the set LANG be an alphabet of some language. $\mathcal{L}_{\text{TYPE}}$ will be the lambek calculus freely generated from TYPE . $\mathcal{L}_{E\text{-TYPE}}$ will be the boolean extension of $\mathcal{L}_{\text{TYPE}}$ and $\mathcal{L}_{E\text{-LANG}}$ will be the boolean extension of the lambek calculus based on LANG . $\mathcal{M}_{\text{TYPE}} = (\mathcal{M}_{\text{TYPE}}, \bullet)$ will be the free monoid determined by TYPE and $\mathcal{M}_{\text{LANG}} = (\mathcal{M}_{\text{LANG}}, \bullet)$ will be the free monoid determined by LANG . A Buszkowski interpretation of $\mathcal{L}_{\text{TYPE}}$ is a functor $I : \mathcal{L}_{\text{TYPE}} \rightarrow P(\mathcal{M}_{\text{LANG}})$, i.e. a functor determined by an assignment of subsets of $\mathcal{M}_{\text{LANG}}$ to the atomic types of $\mathcal{L}_{\text{TYPE}}$. If we want to learn the language determined by I from an abstract speaker we are confronted with elements from $\mathcal{M}_{\text{LANG}}$. From this information we want to reconstruct the relation between $P(\mathcal{M}_{\text{LANG}})$ and $\mathcal{L}_{\text{TYPE}}$. So in fact we are interested in the inverse of I . As it stands I is not a very handy functor to have an inverse. Lets try to imagine the qualities we would like such a new functor to have. We want to have partial models of the language to be learned and partial

models of its syntax. A natural candidate is a functor $F : \mathcal{L}_{E_LANG} \rightarrow \mathcal{L}_{E_TYPE}$. This would correspond to a function from $P(M_{LANG})$ to $P(M_{TYPE})$ assigning a set of types to a set of expressions of the language to be learned.

$$\begin{aligned} F(I) &= I & F(T) &= T \\ F(\perp) &= \perp & F(a \setminus b) &= F(a) \setminus F(b) \\ F(a/b) &= F(a)/F(b) & F(a \bullet b) &= F(a) \bullet F(b) \\ F(a \&b) &= F(a) \& F(b) & F(a \vee b) &= F(a) \vee F(b) \end{aligned}$$

Note that F defines a complete syntax for a language since it assigns a set of types to every singleton element of $P(LANG)$, just like a traditional 'dictionary' for a categorial language would do. In fact F gives an intensional definition of the syntax of a language $\mathcal{L}$. But this is exactly one of the problems we wanted to solve: F specifies precisely the sort $\mathcal{L}_E$ with additional 'lambek' structure we are interested in!

We have identified a syntax with a functor from $\mathcal{L}_{E_LANG}$ to the Extended Lambek Calculus of the types $\mathcal{L}_{E_TYPE}$. From another point of view types are nothing but sets of expressions in $\mathcal{L}_{E_LANG}$. How do these two views relate? Can we get back from the types as expressions in $\mathcal{L}_{E_TYPE}$ to the types as sets in $\mathcal{L}_{E_LANG}$? To realize this we define a new functor $U : \mathcal{L}_{E_TYPE} \rightarrow \mathcal{L}_{E_LANG}$ as:

Definition 2 $U(B)$ is the supremum of all objects $A \in \mathcal{L}_{E_LANG}$ such that there is an arrow $b : F(A) \rightarrow B \in \mathcal{L}_{E_TYPE}$: i.e. $U(B) = (A_1 \vee A_2 \vee \dots \vee A_n)$ for all A_i such that there is an arrow $b : F(A_i) \rightarrow B$.

In our interpretation this will be $U(B) = \bigcup \{A \mid F(A) \subseteq B\}$. The value of U applied to an object B of $\mathcal{L}_{E_TYPE}$ is the union of all objects $A \in \mathcal{L}_{E_LANG}$ that have an image in B . It is easy to show that according to this definition we also have:

$$\begin{aligned} U(I) &= I & U(T) &= T \\ U(\perp) &= \perp & U(a \setminus b) &= U(a) \setminus U(b) \\ U(a/b) &= U(a)/U(b) & U(a \bullet b) &= U(a) \bullet U(b) \\ U(a \&b) &= U(a) \& U(b) & U(a \vee b) &= U(a) \vee U(b) \end{aligned}$$

Suppose each expression has only one syntactical type, then we have $F(\{d\}) = \{t\}$, where t is the type of d . Now U gives us all expressions that have the same type. So we have $d \in UF(d)$. In fact UF is a closure operation in $\mathcal{L}_{E_LANG}$ that projects each expression into the set of expressions that correspond to its syntactical type. FU is the dual interior operation that projects each expression of $\mathcal{L}_{E_TYPE}$ to, what one might call its *cotype*. The functor FU applied to an object $B \in \mathcal{L}_{E_TYPE}$ gives us the set of all the elements of B that are in use as a type of $\mathcal{L}_{E_LANG}$.

Before we proceed we will give a usefull lemma:

Lemma 2 If $U(B)$ is the supremum of all objects A in $\mathcal{L}_{E_LANG}$ such that there is an arrow $b : F(A) \rightarrow B \in \mathcal{L}_{E_TYPE}$: i.e. $U(B) = (A_1 \vee A_2 \vee \dots \vee A_n)$ for all A_i such that there is an arrow $b : F(A_i) \rightarrow B$ then there will be an arrow $b : A \rightarrow U(B) \in \mathcal{L}_{E_LANG}$ which is unique up to equivalence.

Proof The arrow $a : A \rightarrow U(B) \in \mathcal{L}_{E_LANG}$ is equivalent to

$$k_{A, (A_1 \vee A_2 \vee \dots \vee A_n)} : A \vee (A_1 \vee A_2 \vee \dots \vee A \vee \dots \vee A_n)$$

All the arrows of this form are equivalent under all permutations of elements of $(A_1 \vee A_2 \vee \dots \vee A_n)$ so $a : A \rightarrow U(B)$ is unique up to equivalence.

We now prove a theorem that states the existence of a solution to a so-called *universal mapping problem* for $U : \mathcal{L}_{E_TYPE} \rightarrow \mathcal{L}_{E_LANG}$.

Theorem 3 For each object A of $\mathcal{L}_{E_LANG}$ there is an object $F(A)$ of $\mathcal{L}_{E_TYPE}$ and an arrow $type(A) : A \rightarrow UF(A)$ such that for each arrow $f : A \rightarrow U(B) \in \mathcal{L}_{E_LANG}$ there exist a unique arrow $f^* : F(A) \rightarrow B \in \mathcal{L}_{E_TYPE}$ such that $U(f^*)type(A) = f$

Proof Suppose we have an arrow $f : A \rightarrow U(B)$, now $U(B) = \text{Sup}\{X | F(X) \rightarrow B\}$, so we have $F(A) \rightarrow B$. By 2 this gives a unique arrow $f^* : F(A) \rightarrow B \in \mathcal{L}_{E_TYPE}$. Since $UF(A) = U(F(A))$ we can combine $U(f^*) : U(F(A)) \rightarrow U(B)$ and $type(A) : A \rightarrow UF(A)$ to $U(f^*)type(A) : A \rightarrow U(B)$.

This theorem has some interesting consequences:

1. There is a natural transformation $type : 1_{\mathcal{L}_{E_LANG}} \rightarrow UF$; i.e. the diagram in figure 5 commutes. For every $a : A \rightarrow A' \in \mathcal{L}_{E_LANG}$ we have

$$\begin{array}{ccc} \mathcal{L}_{E_LANG}(A) & \xrightarrow{type(A)} & UF(A) \\ \downarrow \mathcal{L}_{E_LANG}(a) & & \downarrow UF(a) \\ \mathcal{L}_{E_LANG}(A') & \xrightarrow{type(A')} & UF(A') \end{array}$$

Figure 5: Type transformation

$$UF(a)type(A) = type(A')1_{\mathcal{L}_{E_LANG}}(a)$$

2. There is a natural transformation $cotype : FU \rightarrow 1_{\mathcal{L}_{E_TYPE}}$ defined by $cotype(B) = (1_{U(B)})^*$; i.e. the diagram in figure 6 commutes. For every $b : B \rightarrow B' \in \mathcal{L}_{E_TYPE}$ we

$$\begin{array}{ccc} FU(B) & \xrightarrow{cotype(B)} & \mathcal{L}_{E_TYPE}(B) \\ \downarrow FU(b) & & \downarrow \mathcal{L}_{E_TYPE}(b) \\ FU(B') & \xrightarrow{cotype(B')} & \mathcal{L}_{E_TYPE}(B') \end{array}$$

Figure 6: Cotype transformation

$$\text{have } 1_{\mathcal{L}_{E_TYPE}}(b)cotype(B) = cotype(B')FU(b)$$

3. $(F, U, type, cotype)$ is an adjointness between $\mathcal{L}_{E_LANG}$ and $\mathcal{L}_{E_TYPE}$, i.e. $(Utype) \circ (cotypeU) = 1_U$, $(type F) \circ (Fcotype) = 1_F$. F can be defined using $*$: for each $f : A \rightarrow A'$ put $F(f) = (type(A')f)^*$.

We have seen that F has an adjoint functor $U : \mathcal{L}_{E_TYPE} \rightarrow \mathcal{L}_{E_LANG}$. This functor gives us exactly the appropriate sets of expressions in $\mathcal{M}_{LANG}$ that correspond with singletons in $P(TYPE)$, i.e. the atomic types of $\mathcal{L}_{TYPE}$. This means that our original interpretation I is faithfully embedded in U . We don't lose information if we lift I to U . The adjointness between F and U gives us a lot of very welcome insight in the structure of our definitions. UF is a closure operation that projects each singleton expression in $\mathcal{L}_{E_LANG}$ on the set of expressions that corresponds to the extensional definition of its syntactical type. FU is the dual interior operation that projects each expression of $\mathcal{L}_{E_TYPE}$ to, what one might call, its cotype. The functor FU applied to an object p gives us the set of all elements that are in use as a type of some element of $\mathcal{L}_{E_LANG}$. Of course, we now have two natural transformations: $\text{type}: 1_{\mathcal{L}_{E_LANG}} \rightarrow UF$ and $\text{cotype}: FU \rightarrow 1_{\mathcal{L}_{E_TYPE}}$. Now what do we gain by these definitions? It appears that the discussion about the relation between syntax and semantics can be lifted to the level of morphisms between Extended Lambek Calculi. Suppose we have

$$F(\{\text{john}\}) = N$$

and

$$F(\{\text{walks}\}) = N \setminus S$$

where N and S are expressions of $\mathcal{L}_{E_TYPE}$ then

$$F(\{\text{john}\} \bullet \{\text{walks}\}) = F(\{\text{john}\}) \bullet F(\{\text{walks}\}) = N \bullet (N \setminus S) = S$$

The expressions S , $N \setminus S$ and N are types. The corresponding extensions are $U(S)$, $U(N \setminus S)$ and $U(N)$, which are sets of expressions in $\mathcal{L}_{E_LANG}$.

4 Semantics

The most interesting aspect of this approach is that we can give a very elegant treatment of semantics that overcomes our objections to the current proposals and partly incorporates them. The basic idea is this. If a syntax for a Boolean extension of a Lambek calculus is a functor $F_{\text{syn}} : \mathcal{L}_{E_LANG} \rightarrow \mathcal{L}_{E_TYPE}$, then a compositional semantics is nothing but an intermediate layer between $\mathcal{L}_{E_LANG}$ and $\mathcal{L}_{E_TYPE}$:

Definition: Suppose $\mathcal{L}_{E_LANG}$ and $\mathcal{L}_{E_TYPE}$ are Boolean extension of Lambek calculi then $F_{\text{syn}} : \mathcal{L}_{E_LANG} \rightarrow \mathcal{L}_{E_TYPE}$ defines an interpretation for $\mathcal{L}_{E_LANG}$ if and only if $(F_{\text{syn}}, U_{\text{syn}}, \text{type}, \text{cotype})$ is an adjointness between $\mathcal{L}_{E_LANG}$ and $\mathcal{L}_{E_TYPE}$.

Definition: Suppose $\mathcal{L}_{E_LANG}$, $\mathcal{L}_{E_TYPE}$ and $\mathcal{L}_{E_SEM}$ are extended Lambek calculi and there are functors $F_{\text{syn}} : \mathcal{L}_{E_LANG} \rightarrow \mathcal{L}_{E_TYPE}$ and $F_{\text{sem}} : \mathcal{L}_{E_LANG} \rightarrow \mathcal{L}_{E_SEM}$ are interpretations of $\mathcal{L}_{E_LANG}$, then F_{sem} is a semantics for the interpretation F_{syn} iff there is a functor $F_{\text{dif}} : \mathcal{L}_{E_TYPE} \rightarrow \mathcal{L}_{E_SEM}$ such that F_{dif} is an interpretation for $\mathcal{L}_{E_TYPE}$ and $F_{\text{syn}} = F_{\text{dif}} F_{\text{sem}}$. See figure 7

F_{syn} and F_{sem} generate a syntax for $\mathcal{L}_{E_LANG}$, F_{dif} generates a syntax for $\mathcal{L}_{E_SEM}$. There are adjoint functors $U_{\text{syn}} : \mathcal{L}_{E_TYPE} \rightarrow \mathcal{L}_{E_LANG}$, $U_{\text{sem}} : \mathcal{L}_{E_SEM} \rightarrow \mathcal{L}_{E_LANG}$ and $U_{\text{dif}} : \mathcal{L}_{E_TYPE} \rightarrow \mathcal{L}_{E_SEM}$. There are natural transformations

$$\text{syntype}: 1_{\mathcal{L}_{E_LANG}} \rightarrow U_{\text{syn}} F_{\text{syn}}$$

$$\text{semtype}: 1_{\mathcal{L}_{E_LANG}} \rightarrow U_{\text{sem}} F_{\text{sem}}$$

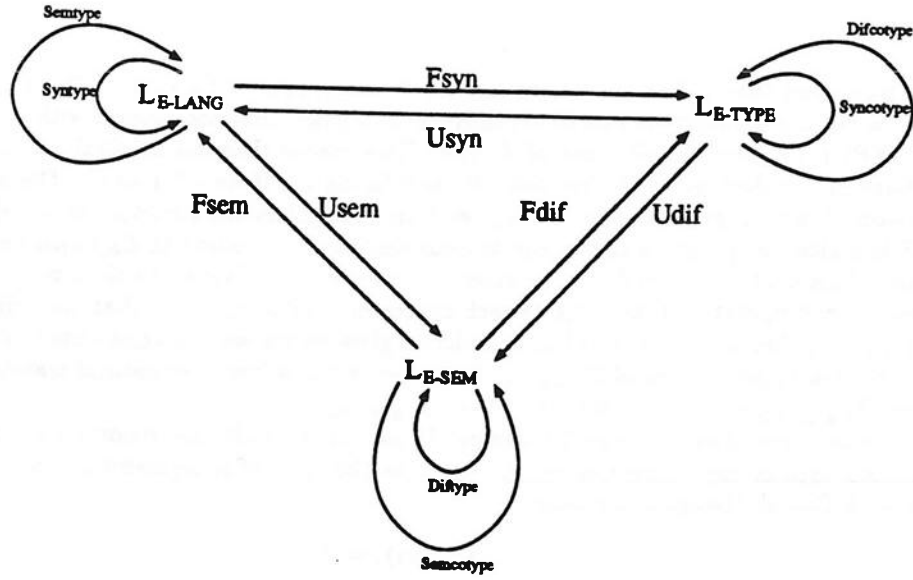


Figure 7: All functors and Transformations involved in a Semantics

$$\text{diftype}: 1_{\mathcal{L}_{E-SEM}} \rightarrow U_{\text{dif}} F_{\text{dif}}$$

$$\text{syncotype}: F_{\text{syn}} U_{\text{syn}} \rightarrow 1_{\mathcal{L}_{E-TYPE}}$$

$$\text{semcotype}: F_{\text{sem}} U_{\text{sem}} \rightarrow 1_{\mathcal{L}_{E-SEM}} \text{ and}$$

$$\text{difcotype}: F_{\text{dif}} U_{\text{dif}} \rightarrow 1_{\mathcal{L}_{E-TYPE}}$$

The transformation diftype defines the 'difference' between syntactic and semantic types. Now we have:

$$U_{\text{syn}} F_{\text{syn}}(l) = U_{\text{sem}} U_{\text{dif}} F_{\text{dif}} F_{\text{sem}}(l)$$

$$U_{\text{sem}} F_{\text{sem}}(l) = U_{\text{syn}} F_{\text{dif}} U_{\text{dif}} F_{\text{syn}}(l)$$

consequently we have:

$$U_{\text{syn}} F_{\text{syn}}(l) = U_{\text{sem}} \text{diftype} F_{\text{sem}}(l)$$

$$U_{\text{sem}} F_{\text{sem}}(l) = U_{\text{syn}} \text{difcotype} F_{\text{syn}}(l)$$

The fact that this creates a compositional semantics is illustrated by the following set of equations:

$$\begin{aligned} F_{\text{syn}}(a \circ b) &= F_{\text{dif}} F_{\text{sem}}(a \circ b) = \\ F_{\text{syn}}(a) \circ F_{\text{syn}}(b) &= F_{\text{dif}}(F_{\text{sem}}(a) \circ F_{\text{sem}}(b)) = \\ F_{\text{dif}} F_{\text{sem}}(a) \circ F_{\text{dif}} F_{\text{sem}}(b) \end{aligned}$$

It is clear that this semantic theory solves the three problems stated before. We solved the problem of specifying additional structure for Boolean extensions of Lambek Calculi. Because of the adjointness the diagram above commutes in two ways and generates some

usefull local natural transformations. This enables us to understand the subtle interplay between syntactic and semantic information in a learning process. Note that both $\mathcal{L}_{E_TYPE}$ and $\mathcal{L}_{E_SEM}$ are associative, so we have an interpretation of semantics that preserves the original idea of a semantic assignment by way of a faithful embedding, and at the same time enables us to distribute semantic information associatively over partial interpretations of the calculus. The theory also enables us to incorporate intensional information in the learning process. An intensional cardinality constraint for two valuedness is an arrow $u_{dif1} : U_{dif}(\{s\}) \rightarrow \{t, f\}$ together with arrows $f_{dif1} : F_{dif}(\{t\}) \rightarrow \{s\}$ and $f_{dif2} : F_{dif}(\{f\}) \rightarrow \{s\}$. They state that the cotype of the syntactic type s is at most one element of the set $\{t, f\}$. This information enables us to develop a strategy for obtaining local completeness results, e.g. when learning Boolean connectives. A codification of a universal law is just an arrow in $\mathcal{L}_{E_SEM}$.

4.1 Some examples

We will give an example that illustrates how our domain theory works in the context of our learning algorithm. Suppose we have the following arrows in Syn :

$$f_{dif1} : F_{dif}(\{t\}) \rightarrow \{s\}$$

$$f_{dif2} : F_{dif}(\{f\}) \rightarrow \{s\}$$

These arrows say that $t(ruth)$ and $f(false)$ are of type $s(sentence)$. Another arrow in Sem :

$$u_{dif1} : U_{dif}(\{s\}) \rightarrow \{t, f\}$$

states that sentences can only be true or false. We will suppose the sentences can only have one semantic value. The semantic value of a sentence will be isomorphic to a singleton value; for each $\{x\} \in L$ we have $f_{sem} x : F_{sem}(x) \rightarrow z$ if and only if $f_{sem} x' : t \rightarrow F_{sem}(z)$. Suppose that we have to analyse a simple set of true sentences:

- john walks
- pete walks
- john walks and pete walks

In our representation this will be something like:

$$f_{sem1} : F_{sem}(\{john\} \bullet \{walks\}) \rightarrow \{t\}$$

$$f_{sem2} : F_{sem}\{pete\} \bullet \{walks\}) \rightarrow \{t\}$$

$$f_{sem3} : F_{sem}(\{john\} \bullet \{walks\} \bullet \{and\} \bullet \{pete\} \bullet \{walks\}) \rightarrow \{t\}$$

By isomorfism we have:

$$f_{sem1'} : \{t\} \rightarrow F_{sem}(\{john\} \bullet \{walks\})$$

$$f_{sem2'} : \{t\} \rightarrow F_{sem}\{pete\} \bullet \{walks\})$$

$$f_{sem3'} : \{t\} \rightarrow F_{sem}(\{john\} \bullet \{walks\} \bullet \{and\} \bullet \{pete\} \bullet \{walks\})$$

We can derive that:

$$\begin{aligned} f_{sem4} : F_{sem}(\{john\} \bullet \{walks\} \cup \\ \{pete\} \bullet \{walks\} \cup \\ \{john\} \bullet \{walks\} \bullet \{and\} \bullet \{pete\} \bullet \{walks\}) \rightarrow \{t\} \end{aligned}$$

By compositionality we have:

$$\begin{aligned} f_{sem1} : F_{dir}(F_{sem}(\{john\} \bullet \{walks\})) \rightarrow F_{dir}(\{t\}) \\ = \\ f_{sem1} : F_{dir}(F_{sem}(\{john\} \bullet \{walks\})) \rightarrow \{s\} \end{aligned}$$

The fact that these structures have a truthvalue guarantees that they are sentences. By associativity of concatenation we can rewrite f_{sem3} like:

$$\begin{aligned} f_{sem3''} : F_{sem}(\{john\} \bullet \{walks\}) \bullet \\ F_{sem}(\{and\}) \bullet \\ F_{sem}(\{pete\} \bullet \{walks\}) \rightarrow \{t\} \end{aligned}$$

By isomorphy we have:

$$f_{sem3''} : \{t\} \bullet F_{sem}(\{and\}) \bullet \{t\} \rightarrow \{t\}$$

So we have:

$$f_{sem3'''} : F_{sem}(\{and\}) \rightarrow \{t\} \setminus \{t\} / \{t\}$$

This gives part of the definition of $\{and\}$. Note that:

$$F_{dir}(\{t\} \setminus \{t\} / \{t\}) = \{s\} \setminus \{s\} / \{s\}$$

but that:

$$U_{dir}(\{s\} \setminus \{s\} / \{s\}) = \{t, f\} \setminus \{t, f\} / \{t, f\}$$

so that:

$$difttype(\{t\} \setminus \{t\} / \{t\}) = \{t, f\} \setminus \{t, f\} / \{t, f\}$$

Given the fact that semantic values of singletons are isomorphic to singleton values there are only sixteen consistent semantic definitions of $\{and\}$ under $\{t, f\} \setminus \{t, f\} / \{t, f\}$. When we have derived the following 4 arrows:

$$\begin{aligned} f_{semr} : F_{sem}(\{and\}) \rightarrow \{t\} \setminus \{t\} / \{t\} \\ f_{sems} : F_{sem}(\{and\}) \rightarrow \{f\} \setminus \{f\} / \{t\} \\ f_{semt} : F_{sem}(\{and\}) \rightarrow \{t\} \setminus \{f\} / \{f\} \\ f_{semu} : F_{sem}(\{and\}) \rightarrow \{f\} \setminus \{f\} / \{f\} \end{aligned}$$

we can derive:

$$\begin{aligned} f_{semv} : F_{sem}(\{and\}) \rightarrow \\ \{t\} \setminus \{t\} / \{t\} \cap \\ \{f\} \setminus \{f\} / \{t\} \cap \\ \{t\} \setminus \{f\} / \{f\} \cap \end{aligned}$$

$$\{f\} \setminus \{f\} / \{f\}$$

we know that we have a maximal definition of {and} given our constraints. If we add another arrow e.g.

$$f_{sem} w : F_{sem}(\{and\}) \rightarrow \{f\} \setminus \{t\} / \{t\}$$

we can derive that some elements of type {s} will have {t, f} as a semantic value and this is in contradiction with the constraint that semantic values of sentences are isomorphic to singleton values.

This gives an illustration of the possibilities of our domain theory in a learning environment. There are other possibilities. It is for instance possible to define a compositional semantics for associative languages. Here is an illustration of an associative version of the propositional calculus:

$$F_{sem}(\{a\}) = \{1\}$$

$$F_{sem}(\{\neg a\}) = \{0\}$$

$$F_{sem}(\{a \vee a\}) = \{1\}$$

$$F_{sem}(\{\neg a \vee a\}) = \{0, 1\}$$

$U_{sem}(F_{sem}(\{a\}))$ is the set of all true formulae and $U_{sem}(F_{sem}(\{\neg a \vee a\}))$ is the set of all formulae.

4.2 Conclusion

Perhaps the most interesting feature of this theory is structural resemblance between a language, its syntax and semantics. We can generalize the study of categorial syntax and semantics to the study of adjoint functors from $\mathcal{L}_{E,N}$ to $\mathcal{L}_{E,N}$, where $\mathcal{L}_{E,N}$ is the extended Lambek calculus of the natural numbers. In this object we will find any thinkable categorial relation between a categorial grammar, its syntax and semantics. Of course a language with a syntax could have a lot of different semantics, some of which are themselves compositional with respect to other semantics. At the same time every semantics is also a syntax and vice versa. This enables us to model the learning process as a form of deduction in this special functor category associated with $\mathcal{L}_{E,N}$, i.e. we have constructed a suitable domain to understand the learning process.

5 Notes

¹ For a discussion see Barr & Wells 1990 pg. 280.

² See Lambek (1988) for a brief discussion.

6 References

- Adriaans, P.W.:1991, Algoritmen voor het effectief leren van categoriale en contextvrije talen, *Proceedings NAIC 1991*, Amsterdam
M. Barr and G. Wells: 1990, *Category Theory for Computing Science*, Prentice Hall int. inc., Hertfordshire
Benthem, Johan van: 1986, *Essays in Logical Semantics*, Reidel, Dordrecht.

- Benthem, Johan van: 1991, *Language in Action*, North Holland Publishing Company, Amsterdam
- Buszkowski, W.: 1986, Completeness Results for Lambek Syntactic Calculus, *Zeitschrift für mathematischen Logik und Grundlagen der Mathematik*, **32**, 13-28
- Buszkowski, W. and Penn, G.: 1989, *Categorical Grammars Determined from Linguistic Data by Unification*, Technical Report 89-05, The University of Chicago
- Li, M. and Vitanyi, P.M.B.: 1989, A Theory of Learning Simple Concepts under Simple Distributions and Average Case Complexity for the Universal Distribution, *ITLI Prepublication series for Computation and Complexity Theory CT-89-07*, Universiteit van Amsterdam
- Li, M. and Vitanyi, P.M.B.: 1990, Kolmogorov Complexity and its Applications, in J. van Leeuwen, (ed.), *Handbook of Theoretical Computer Science*, Elsevier Science Publishers, 188-254
- Lambek, J, P.J. Scott: 1986, *Introduction to Higher Order Categorical Logic*, Cambridge UP, Cambridge
- Lambek, J.: 1988, Categorical and Categorical Grammars, in R.T. Oehrle, E. Bach, D. Wheeler, (eds.), *Categorical Grammars and Natural Language Structures*, Reidel, Dordrecht and Boston
- Moortgat, M.: 1988, *Categorical Investigations, Logical and Linguistic Aspects of the Lambek Calculus*, dissertatie Universiteit van Amsterdam, Amsterdam
- Oehrle R.T. et al. (eds.): 1988, *Categorical Grammars and Natural Language Structures*, Reidel, Dordrecht en Boston
- Roorda, D.: 1991, *Resource Logics: Prooftheoretical Investigations*, dissertatie Universiteit van Amsterdam, Amsterdam

The Kinematics of Presupposition*

David I Beaver

Universities of Edinburgh and Amsterdam

1 Introduction

Russell's analysis of definite descriptions provides us with celebrated propositions like "There is one and only one King of France, and he is bald". Strawson, however, took the Fregean line that sometimes referring expressions could fail to denote, and was not prepared to accept that "The King of France is bald" *asserted* the existence or uniqueness of a King, preferring to believe that these were necessary conditions for the sentence to have a meaning at all: *presuppositions* as we would now call them.

The intuition that a Russellian analysis cannot be right is perhaps strongest in the case of imperatives. If the conditions imposed by definite descriptions were part of the propositional content of a sentence, then we should expect the command "Shave the king of France's head!" to be, in part, an instruction to restore the French monarchy, or at least to restore the king's head. (Further still, if it turns out that the restored head is hairless, we should presumably have to restore the hair as well, since this is, arguably, a precondition for removing it.) Yet the desire for such restoration plays no part in any natural interpretation of the sentence.¹

Suppose we go along with such reasoning, and say that there is more to meaning than what is asserted, there additionally being a component of *presupposition*: then our problems are just beginning. The Russellian analysis involves both a perspicuous notation for the meaning of a sentence, and a well understood logic. Strawson's approach suggests neither. If we want to even begin the task of reasoning about Strawsonian meanings we will have to notate those meanings in such a way that both assertive and presuppositional aspects of meaning are transparent, define a suitable semantics for the resulting structures, and ascertain the relevant (non-standard) logic.

Permit me to make a slightly jarring leap, to an oddity of modern dynamic semantics. In a dynamic logic, formulae are typically given a relational interpretation, and may be thought of as instructions for an agent to move across some space: in this space the formula relates points at which the instructions are valid to points which the agent may end up at by following the instructions. For our purposes the space will be a space of information states, and the agent will tend to move in the direction of increasing information. Yet not all formulae will denote movement: those that do not are what Veltman (1990) has termed *tests*.

A good example of a test arises with Veltman's *might* operator, which he uses to model epistemic possibility: the intuition is that you can say "possibly X" just in case at that point in the discourse X has not been ruled out. He defines a dynamic semantics for a simple extension of a propositional logic, in which the semantics for ordinary formulae are naturally viewed in terms of Stalnaker's assertion perspective (Stalnaker 1978). An information state is modelled as a set of worlds, sometimes called the *context set*, intuitively those worlds which the agent still considers as possible candidates for the real world, and an ordinary formula is an instruction to narrow down that set by throwing out worlds which are not compatible with some new information. A formula of the form *might* ϕ is a test: it is an instruction to maintain the current context set provided some of its constituent worlds support ϕ , and otherwise move to the absurd information state, here identified with the empty context set.

Let us say that an assertion is a formula that can help us reach a maximal non-contradictory information state, potentially narrowing down the context set to a single world. In that case a formula of the form *might* ϕ is non-assertive: it cannot productively inform the agent. In terms of the spatial analogy it says: stand still, take a look around, and if you cannot see ϕ on any of the paths ahead, commit suicide. After hearing such an instruction, an agent is at best in the same place as before, and at worst nowhere. The conundrum is this: how can formulae which can never productively inform be useful in modelling natural language, where the whole point is to productively inform? Whatever the successes of Veltman's analysis of "might" (which anyway he only provides as an initial example of his framework, the main applications being within default reasoning) it cannot be the final answer, since telling somebody what you regard as possible may be informative.

In fact I believe that whilst a test operator cannot adequately capture the meaning of "might", there is a place for test operators in our analysis of meaning. I will claim that whilst the main

*I am particularly grateful to Robin Cooper, Paul Dekker, Willem Groeneveld, Lex Holt, Laszlo Kalman, Ewan Klein and Henk Zeevat for valuable feedback on earlier versions of these notes.

¹Grice (1989) credits this sort of objection to the Russellian analysis to Searle.

purpose of language is to assert information, so that we should not expect the translation of any sentence to be a pure test, there is a non-assertive component to meaning which can be usefully modelled using a test operator within a dynamic logic. We might say that the result is a logic for Strawsonian meanings. Incidentally, thinking about the behavior of presuppositions will also provide us with a natural resolution to the problem of defining an assertive *might* operator, showing that from an appropriate perspective Veltman's test can provide information after all.

I should stress that a dynamic perspective on presupposition is not in itself novel. Most of the hard work is already to be found in papers by Karttunen, Karttunen & Peter's (hence K&P), and Heim. (I will sometimes refer collectively to the model they have developed as the CCP model, from Heim's description of sentence meanings in terms of *Context Change Potentials*). Indeed, presupposition phenomena formed one of the main motivations for Stalnaker's assertion model. However, the last few years have seen a flowering in the understanding and technical elaboration of all things dynamic, and in the remainder of this paper I will attempt to show how our understanding of the major problems of presupposition theory can benefit from application of the tools of dynamic semantics.

A Handy Guide to this Paper

Presupposition being a fairly well circumscribed area, there is a forest of problems of both a philosophical and a linguistic nature that any theory must tackle. I have divided the problems into three groups: the projection of presuppositions from truth-functionally connected compounds, the behavior of presuppositions in quantificational contexts, and the accommodation of presupposed information. There is a section devoted to each of these topics, each section starting with an empirically oriented subsection concerning the chief problem(s) to be dealt with, followed by further subsections containing the mixture of technicality and speculation that constitutes the theory. The summary below should help to make clear the relationship between the different sections, and provide an overview of the different motivations for the approach adopted.

- Section 2 is devoted to making explicit the links between existing context change models of presupposition and Veltman's *Update Semantics*. This serves to introduce the framework which is used throughout the remainder of the paper.
- Section 3 concerns the interaction between presupposition, quantification and anaphora. K&P have presented a Montague style treatment of presupposition which is *heterogeneous* in the sense that meanings are split up into two parts, one containing the presupposition, and the other containing the assertion. Unfortunately such an approach cannot adequately account for the projection of presuppositions from quantificational contexts. It is shown how this motivates the sort of approach developed in Section 2, which is essentially *homogeneous* and uses a unary operator to embed presuppositions within the primary assertion.
- The homogeneous account improves over that presented by K&P through providing a simple mechanism for quantifiers in the assertion to bind variables in the presupposition. Considering some more data makes it clear that it is also advantageous to allow quantifiers in the presupposition to bind variables in the assertion, and this inspires the introduction of a Groenendijk & Stokhof (hence G&S) style dynamic approach to variable binding. Utilising such an approach also makes the model appropriate to the demands of standard cases of discourse and donkey anaphora.
- It is shown how the resulting account can make sensible predictions on a range of examples that have troubled existing CCP theories without invoking any non-compositional mechanism of *local* or *intermediate accommodation*. These examples concern the strength of the presupposition that emerges from a quantified context. It also becomes clear that the *kinematic* account developed in Section 3 can provide some overview regarding what the options are within a CCP theory and where the existing accounts fit in.
- A theory of presupposition that involved no notion of accommodation would be inadequate, since it would fail to account for the informativeness of presuppositions. In Section 4 it is shown how *global accommodation* can be modelled within a CCP account.
- The model developed to account for global accommodation provides an interesting perspective on a range of issues on the border between semantics and pragmatics. Section 4.3 is devoted to a speculative exploration of some of these issues, including the definition of an assertive *might* operator, the strengthening of the typical CCP predictions regarding the presuppositions of the conditional, and the licensing of definite descriptions.
- The final parts of Section 4, are devoted to a comparison with some other types of account. Firstly I consider the relationship between the kinematic model developed in these notes and default accounts of presupposition, showing that presuppositions can be defeasible within a CCP theory, and showing how some of the classic cases of presupposition cancellation could be interpreted in terms of accommodation. I then turn briefly to van der Sandt-ian accounts involving accommodation of representations into discourse structure, suggesting a way in which the kinematic model could enable a crucial distinction to be made between different types of accommodation.

2 Update Semantics and Presupposition Projection

2.1 The Projection Problem

As originally conceived by Langendoen and Savin (1971), the projection problem consists of determining the presuppositions of complex sentences in terms of the presuppositions of their parts. To cast the problem in this way is to assume that there is some monolithic class of presuppositional constructions, presumably including clefts, definite descriptions, factive verbs and many more, all of which behave the same way under embedding. Whilst I am by no means committed to that assumption, for the most part I will concentrate on factive constructions like "regret", "know" and "is annoyed that", in the hope that they are representative of the remainder of the class. At any rate, factive constructions certainly form a class worth study in their own right.

To simplify the analysis, I will assume a translation of English sentences into a formal language, and cast the projection problem in terms of inference relations that must hold in a logic for this formalism. Although the translation function itself will not be formally specified in this paper, it should be clear that the function envisaged in some sense preserves the structure of the original English: the translation will at least be compositional in spirit.

For the purposes of this section a propositional language will be used, consisting of some atomic propositions, the normal truth-functional connectives connecting in their normal way, and two additional unary operators $\Diamond$ and ∂ . The $\Diamond$ will play the role of Veltman's *might* operator, and $\partial\phi$ will be interpreted as "the presupposition that ϕ holds", and the translation of a simple sentence involving a factive verb, such as "John regrets that Bill is happy" will assume a division of the sentence's meaning into a presupposition and an assertion.² Let us say that in this case the presupposition is the atomic proposition that Bill is happy, which will be written BIH , and the assertion is another atomic proposition, which we can imagine is concerned with John's mental state, written $\text{R}(\text{J}, \text{BIH})$. Then the translation of the sentence will be the conjunction $\partial(\text{BIH}) \wedge \text{R}(\text{J}, \text{BIH})$.

Let me make some observations on this translation. It is important that the apparent complexity of $\text{R}(\text{J}, \text{BIH})$ is purely a notational convenience: as an atomic proposition, semantically $\text{R}(\text{J}, \text{BIH})$ has no internal structure, and in particular it has nothing to do with the atomic proposition BIH : this is clearly inadequate. I do not pretend to have given a suitable semantics for "regrets", or given any explanation of the source of its presupposition. A full account of the semantics of "regrets" would demand an understanding of the nature of intensionality and the folk psychological notions implicit in attitude reports: I would not know where to begin with these topics! I will merely say that I find it useful to abstract across what presuppositional constructions have in common by assuming that they have a definite presuppositional component and regarding this component as simply conjoined with the primary content. Following this approach the logic of presupposition becomes quite transparent.

I note in passing that, so far as I am aware, nobody else has chosen to represent presuppositions using a distinct unary operator. One might say that whilst other notations tend to hide away the presuppositions within the meaning of the atomic propositions, or else keep the presuppositions as some formal aside, explicit but separate from the principal logical form, in the current paper translations wear their presuppositions on their sleeves. Given that there is no commitment to a psychologically real level of logical form here, it is hard to see any empirical significance in this decision. Yet the notation chosen here will simplify some of the technicalities, especially when, in Section 3, we come to the issue of scope and binding relations between presupposition and assertion.

We are now in a position to state the projection problem in terms of inferences we would like to have between formulae of the given propositional language. These inferences are summarised in Table 1. The first three inference patterns express the standard properties of presupposition, namely emergence from atomic contexts, embedding under negation, and embedding in the antecedent of a conditional: this behavior is often used as a test for the presence of a presupposition. The fourth pattern is more contentious, giving a type of restricted emergence of presuppositions from the consequent of a conditional: later we will consider how this pattern can be strengthened. The final pattern represents emergence from recursive embedding within presuppositional contexts. Of course, these five inference patterns do not represent the only desiderata for our logic. These are merely additional inference patterns above and beyond all the classical entailments that we would expect to hold between non-presuppositional formulae.

It has been my intention to present in Table (1) an abstract characterisation of the projection problem that is, in so far as this is possible, theory neutral. I would hope that even proponents of defeasible accounts of presupposition (see eg. Gazdar 1979; Mercer 1988) would not object too strongly: after all, the ' $\models$ ' in the table could be interpreted as a form of defeasible entailment. There is some discussion of defeasibility in Section 4.4, below.

²Unlike K&P, I will not be assuming such a simple division for complex sentences.

If we know ...	Then we know ...	Inference Pattern
John regrets that Bill is happy. $\partial(\text{BIH}) \wedge \text{R}(\text{J}, \text{BIH})$	Bill is happy BIH	$\partial(A) \wedge B \models A$
John doesn't regret that Bill is happy. $\neg(\partial(\text{BIH}) \wedge \text{R}(\text{J}, \text{BIH}))$	Bill is happy BIH	$\neg(\partial(A) \wedge B) \models A$
If John regrets that Mary is singing, then he should buy ear-plugs. $(\partial(\text{MIS}) \wedge \text{R}(\text{J}, \text{MIS})) \Rightarrow \text{JSBE}$	Mary is singing MIS	$(\partial(A) \wedge B) \Rightarrow C \models A$
If Mary is in the bath, then John regrets that she is singing. $(\text{MIB} \Rightarrow (\partial(\text{MIS}) \wedge \text{R}(\text{J}, \text{MIS})))$	Mary's singing if bathing $\text{MIB} \Rightarrow \text{MIS}$	$A \Rightarrow (\partial(B) \wedge C) \models A \Rightarrow B$
Bill regrets that John regrets that Bill is happy $\partial(\partial(\text{BIH}) \wedge \text{R}(\text{J}, \text{BIH})) \wedge \text{R}(\text{B}, \text{R}(\text{J}, \text{BIH}))$	Bill is happy BIH	$\partial(\partial(A) \wedge B) \wedge C \models A$

Table 1: The Projection Problem

2.2 The Context-Change-Potential Model

The Context Change Potential model has developed from Robert Stalnaker's work on *assertion* and *pragmatic presupposition* (Stalnaker 1974; Stalnaker 1978) and a series of papers by Lauri Karttunen (eg. Karttunen 1973; Karttunen 1974) culminating in a joint paper with Stanley Peters (K&P 1979), which still represents the only serious attempt at a strictly compositional account of presupposition. After taking some battering in the late seventies and early eighties, their ideas were given a new lease of life by Irene Heim (Heim 1983), and up to now her short paper has remained the definitive statement on what can be achieved in the CCP approach.

In Definition 2, below, I present a renotation of the CCP model in the form of a logic over formulae in the propositional language introduced earlier. The resulting system is strikingly close to the Update Semantics (US) introduced in Veltman (1990), also discussed above, the main difference being my introduction of the ∂ operator, which brings a new type of partiality into the logic. Whereas in Veltman's semantics formulae are interpreted as total functions between partial models of reality (context sets), in the semantics below formulae are interpreted as partial functions between partial models of reality.³

Definition 1 (Symbols)

ϕ, ψ are (possibly atomic) propositions; a world, w , assigns a boolean to every elementary proposition; a context set, σ , is a set of worlds; $X \doteq Y$ means " X is Y if Y is defined, else X is undefined"; $\setminus$ is set subtraction.

Definition 2 (Semantics of Partial Update Logic)

$$\begin{aligned}
\sigma[\phi] &\doteq \{w \in \sigma \mid w(\phi)\} \quad (\text{for atomic } \phi) \\
\sigma[\phi \wedge \psi] &\doteq \sigma[\phi][\psi] \\
\sigma[\neg\phi] &\doteq \sigma \setminus \sigma[\phi] \\
\sigma[\partial\phi] &= \sigma \text{ iff } \sigma[\phi] = \sigma \text{ else undefined} \\
\sigma[\Diamond\phi] &\doteq \sigma \text{ iff } \sigma[\phi] \neq \emptyset \\
&\quad \emptyset \text{ iff } \sigma[\phi] = \emptyset \\
\sigma[\phi \Rightarrow \psi] &\doteq \sigma[\neg(\phi \wedge \neg\psi)]
\end{aligned}$$

In all of these clauses a formula is interpreted as a function from context sets to context sets, that is to say, a CCP. Let us consider some of the clauses individually.

The base case interprets an atomic proposition as an instruction to remove from the context set any worlds that are not compatible with the new proposition, just as was discussed in the introduction to this paper.

The clause for conjunction defines the local context at a given point in a discourse in terms of the initial context of utterance incremented with the information provided by the discourse up to that point. Thus conjunction becomes functional composition: we update with the first conjunct to get to an intermediary state, and then with the second to take us to the final output. But why should conjunction be asymmetric?⁴ Here we can find some new motivation beyond that cited by

³ The relation between the CCP model and US is also clear in Zeevat (1991).

⁴ There is a case to be made for also having a symmetric (ie. static) conjunction in the logic, which could be defined by $\sigma[\phi \& \psi] \doteq \sigma[\phi] \cap \sigma[\psi]$. It would be natural to use this, for example, within complex lexical translations, where there is no reason to say that there is any privileged order to the different components of lexical meaning.

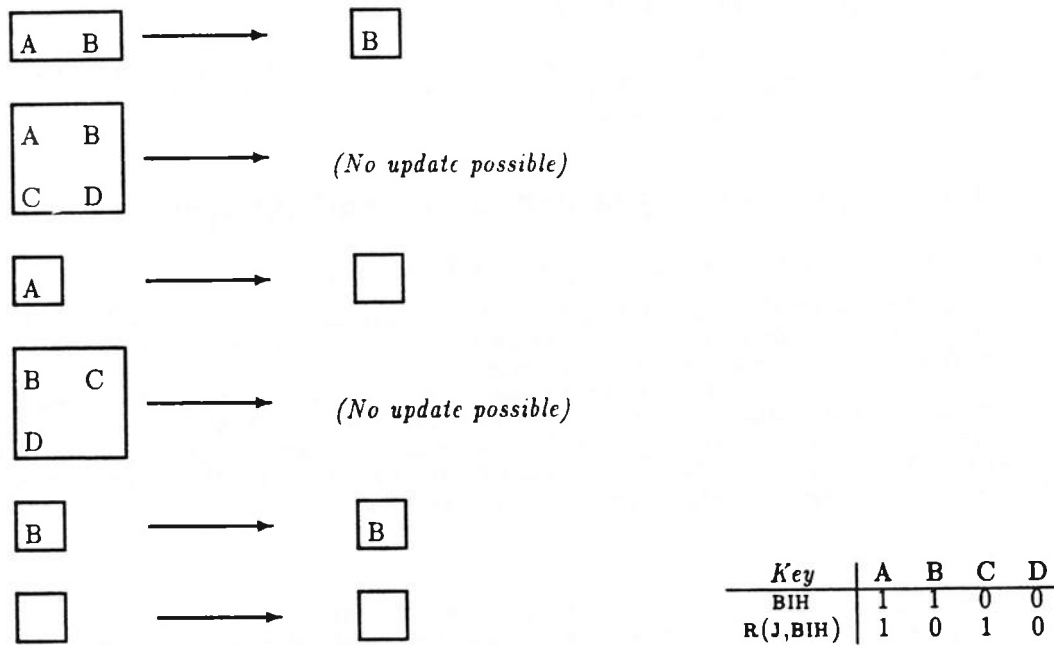


Figure 1: Updates associated with $\neg(\partial\text{BIH} \wedge \text{R}(\text{J},\text{BIH}))$

Karttunen or Heim, with Veltman-esque examples like the following:

- (1)
 - a. Maybe this is Frank Veltman's example. It isn't his example!
 - b. $\Diamond\text{FVE} \wedge \neg\text{FVE}$
- (2)
 - a. ?This is not Frank Veltman's example! Maybe it's his example.
 - b. $\neg\text{FVE} \wedge \Diamond\text{FVE}$

As has been indicated, the $\Diamond$ is essentially Veltman's *might* operator, apart from some syntactic restrictions that he places on its occurrence. The first example gives us a function from contexts in which the origin of the example may be uncertain to contexts in which the example is known not to be Frank's. By contrast, the second example gives us a function which only has the empty set (that is, the absurd information state) in its range, this presumably accounting for its oddity. So, the asymmetry of conjunction can be motivated not only by presupposition data, but, at least in some cases, also by independent principles of textual coherence.

Negation is defined as a simple set complement operation: to update a context with $\neg\phi$, try updating with ϕ : the resulting subset of worlds contains only those members of the original context that are compatible with ϕ , and since we are seeking the set of worlds that is incompatible with ϕ , we must take the subset away from the original context.

The crucial clause is that for ∂ . A formula of the form $\partial\phi$ is interpreted as a fixed-point function that is only defined on a context if ϕ is already accepted in that context. The facts of presupposition projection are then explained in terms of the definedness of a formula's subparts within the local contexts where the subparts occur. Consider the projection of presuppositions from within a negation, as in example (3).

- (3)
 - a. John doesn't regret that Bill is happy
 - b. $\neg(\partial\text{BIH} \wedge \text{R}(\text{J},\text{BIH}))$

The argument of the negation is simply evaluated in the same context as that where the negation itself occurs, so a negated formula will be defined on a context only when the argument is defined — in other words the presuppositions of a negative will be the same as for a corresponding positive.

It might be helpful to consider the sort of partial function associated with the translation of example (3): Figure 1 shows some of the relevant updates. In this example there are only two relevant propositions, concerning Bill's and John's mental states respectively, and so it is only necessary to consider contexts made up of four worlds which differ with respect to the value they assign to these two propositions (here labelled A–D.) It can be seen that the function associated with the formula in (3b) is not defined on contexts where either Bill's happiness is in doubt or Bill is known to be unhappy. On the other hand, in contexts that only contain worlds where Bill is happy, the effect of updating with this formula is simply to remove any worlds where John has regrets.

Finally, we need a notion of semantic entailment for the system above:

Definition 3 (Entailment in Partial Update Logic)
 $\phi_1, \dots, \phi_n \models \psi$ iff $[\phi_1] \dots [\phi_n] = [\phi_1] \dots [\phi_n][\psi]$

This is the second of the three entailment notions considered by Veltman, and essentially it says that one thing entails another if once you know the first you have nothing to learn from the second. This yields all the entailments in Table 1.⁵

3 Presupposition, Quantification and Anaphora⁶

3.1 The Quantificational Projection Problem

Once again I will simplify by assuming a translation of English sentences into a formal language, this time a first order predicate language augmented with the unary operator ∂ , and expressing the problem in terms of inferences between sentences of this language.

It is fair to say that there is significant disagreement about the facts of the projection of presuppositions from quantified contexts, but by considering the behavior of existing accounts it should be possible at least to gather some minimal constraints on the logic: these can be strengthened later, according to taste. In what follows I have divided the constraints into three dimensions of variation, *sensitivity*, *dynamism* and *weakness*.⁷

Sensitivity

K&P (1979) define a Montague style fragment in which the meaning of a sentence is divided into two components. One of these components contains the primary assertion of the sentence, and the other contains the presuppositions. Each syntactic composition in the grammar is accompanied not just by a rule for combining components of the primary assertion, but also by a rule for putting together the presuppositions. It might seem that this provides a completely general framework in which to state how presuppositions behave, and in fact there have been suggestions that the system is *ad hoc* since there are so few constraints on what projection rules can be defined.

Yet in spite of the apparent generality there is a serious restriction inherent in the paired-meanings approach: there is no obvious way to define scope and binding relations between the presupposition and the assertion. In particular, the presupposition is *insensitive* to variable bindings introduced in the assertion, and as K&P point out in a footnote, this causes problems with cases like (4).

- (4) a. ?Somebody managed to succeed George IV
 b. $\exists x \text{ succeeded_g4}(x) \wedge \partial(\text{difficult_succeeding_g4}(x))$ ⁸

This sentence is odd since it suggests that the person who succeeded George IV had some difficulty ascending to the throne when in fact we know that the next in line had no difficulty. However, the K&P system only gives this sentence the presupposition that it was difficult for *somebody* to succeed George IV. Since this presupposition is obviously satisfied, there being many people for whom it was difficult to succeed George IV, the K&P system does not predict that (4) is anomalous.

In the suggested translation of example (4) above, the presupposition contains a free variable. For the formula to make sense, the presupposition must be *sensitive* to the external binding of that

⁵ Whilst the entailment relation in definition 3 has its merits, it also has its failings. For instance, as Willem Groeneveld has pointed out to me, it does not yield *substitutivity*: two formulae which mutually entail each other may behave differently when embedded within other formulae. In particular we have the unintuitive result that for any formula ϕ , $\partial\phi \models \phi$ and $\phi \models \partial\phi$, but that formulae containing ϕ may have different entailments to formulae containing $\partial\phi$. However, there are alternatives to definition 3. Suppose we were to define a notion of entailment $\models^*$ as follows:

$\phi_1, \dots, \phi_n \models^* \psi$ iff $[\phi_1] \dots [\phi_n] = [\phi_1] \dots [\phi_n][\psi]$ and $\mathcal{D}([\phi_1] \dots [\phi_n]) \supseteq \mathcal{D}([\psi])$

The additional restriction on domains means that one formula entails another if whenever you are in a position to learn something from the second, you could learn the same (and possibly more besides) from the first. Now we have $\phi \models^* \partial\phi$ but $\partial\phi \not\models^* \phi$. On this notion we would in fact lose all the entailments in Table 1, but this is not as serious as it might seem. We would at least retain weaker versions of all the inference patterns where the conclusions are embedded under a ∂ operator. At any rate, I do not offer $\models^*$ as a final solution, but merely as a pointer for further research.

⁶ Much of the technical development in this section reflects joint work with Paul Dekker whose reformulation of DPL (Dekker 1992) is crucial to the current paper.

⁷ The reader should be wary of taking my word on what existing accounts predict, as I have occasionally used some latitude in interpreting what has been said and attempting to apply it to specific examples that may not have been considered by the original authors. Having said that, I do not expect my characterisation of these accounts to be particularly contentious.

⁸ Note that in all the translations below I have not explicitly delimited the scope of existential quantification. It will turn out to be useful to assume a non-standard, dynamic notion of scope, in which we have associativity, so that $(\exists x\phi) \wedge \psi \equiv \exists x(\phi \wedge \psi)$ and we can simply ignore the bracketing.

variable. Expressing the need for sensitivity in terms of an inference pattern gives us something like: $\exists x a(x) \wedge \partial(b(x)) \models \exists x a(x) \wedge b(x)$. (We will attend later to the question of whether there is a stronger universal presupposition associated with the sentence.)

K&P have at least two options in order to cure their system's *insensitivity*. Firstly they could define the semantics of their system so as to achieve the necessary binding: this simply seems unnatural given their notation. Secondly they could include more of the asserted information in the presupposition, in the case of example (4) the information that the relevant person with difficulty was in fact the successor to the throne. But following this path would lead to the duplication of all the asserted information in the presupposition, and, at least from a notational point of view, this also seems awkward. It should be clear that I prefer a third option, namely choosing a notation in which the problem does not arise, for in defining a logic over the language used in this section, *sensitivity* seems not so much a dimension of variation as plain common sense.

Dynamism

That K&P's system does not allow binding of variables in the presupposition by quantifiers in the assertion is one side of a coin. The other side is that quantifiers in the presupposition cannot bind variables in the assertion. Consider examples (5) and (6):

- (5) a. I know that a thief broke in. He stole my thesis.
 b. $\partial(\exists x \text{thief}(x) \wedge \text{broke_in}(x)) \wedge \text{know}(me, \exists y \text{thief}(y) \wedge \text{broke_in}(y)) \wedge \partial(\exists z \text{my_thesis}(z)) \wedge \text{stole}(x, z)^9$
- (6) a. If John is married then his wife is very tolerant.
 b. $\text{married}(j) \Rightarrow (\partial(\exists x \text{wife_of}(j, x)) \wedge \text{tolerant}(x))$

I do not want to make any great claims for the treatment of definite descriptions implicit in the translations above: for a start I have completely left aside the issue of uniqueness. What does seem right to me is that these translations make definites presuppose rather than assert the existence of an object, and that the object thus introduced can later fill argument slots in asserted predications. Whether or not the filling of the slot is mediated by a pronoun, it seems to me that the simplest explanation of the slot filling will be, as usual, in terms of bound variables. The same holds for the coreference of an indefinite introduced within a presupposition (eg. "a thief" in (5)) and a later pronoun: a bound variable solution would be nice. This means that the ∂ -operator must be externally dynamic, in the sense of G&S (1990). In terms of entailment this will yield patterns like: $\partial(\exists x a(x)) \wedge b(x) \models \exists x a(x) \wedge b(x)$. We will return to the treatment of definite descriptions in Section 4, where the issue of *licensing* will be briefly discussed.

Note that making presuppositions dynamic would achieve the same results as van der Sandt's DRT-based model (van der Sandt 1990) achieves non-compositionally, by invoking *accommodation*. When a presupposition is not satisfied during the construction of a DRS, the DRS is repaired by addition of the presupposed information in some appropriate place. As a default the information is added to the top level of the DRS — so-called *global accommodation*. This is what would happen in example (5). However, in example (6), where some form of accommodation is necessary to provide the discourse marker for the wife, global accommodation would produce the unwelcome inference that John really does have a wife, whereas it is clear in this example that the existence of a wife is in doubt. Thus van der Sandt is forced to invoke *intermediate* or *local accommodation*, where the existence of a wife is only added to the antecedent or consequent of the conditional. Calculating the exact landing site of the presupposition requires quite a complicated chain of pragmatic reasoning in this case: it would be nice if we could predict the right result in a CCP account without local or intermediate accommodation. Ideally the two premises $\text{married}(j) \Rightarrow \exists x \text{wife_of}(j, x)$ and $\text{married}(j) \Rightarrow (\partial(\exists y \text{wife_of}(j, y)) \wedge \text{tolerant}(y))$ should not entail $\exists y \text{wife_of}(j, y)$, and this should be determined in purely semantic and declarative terms.

Weakness

Consider examples (7–9):

- (7) a. Everyone who serves his king will be rewarded.
 b. $\forall x ((\partial(\exists y \text{king_of}(x, y)) \wedge \text{serves}(x, y)) \Rightarrow \text{rewarded}(x))$
- (8) a. No nation cherishes its king.
 b. $\neg(\exists x \text{nation}(x) \wedge \partial(\exists y \text{king_of}(x, y)) \wedge \text{cherishes}(x, y))$
- (9) a. A fat man was pushing his bicycle.
 b. $\exists x \text{fat_man}(x) \wedge \partial(\exists y \text{bike_of}(x, y)) \wedge \text{pushing}(x, y)$

⁹Here I have not stuck to the language of FOPL+ ∂ , in order to give a rough and ready translation. Note that the higher order part, the "know" clause is by assumption non-presuppositional, and so does not affect the main point which is to establish the behaviour of presuppositions.

Heim's raw CCP account (Heim 1983), without the addition of an accommodation mechanism, predicts that these examples have the presuppositions that everyone has a king, every nation has a king, and every fat man has a bike respectively.¹⁰

Intuitions vary in the case of (7). Neither of the accounts in Cooper (1983) and van der Sandt (1990) predict any presupposition at all for this sentence, both yielding the meaning that everyone who has a king and serves him will be rewarded: deriving such a meaning seems a good minimal constraint on a theory, whether or not there is an additional universal presupposition.

As far as examples (8) and (9) are concerned, intuitions seem much clearer: the presuppositions predicted by Heim's raw CCP account are too strong. Cooper gives example (8) the same universal presupposition as Heim, whereas van der Sandt predicts merely that the sentence means that no nation which has a king cherishes him. Both van der Sandt and Cooper only attribute to example (9) an existential presupposition, namely that some fat man has a bike.

Van der Sandt's theory makes sensible predictions not only with regard to the dimension of strength, where his predictions are uniformly weak, but also regarding sensitivity and dynamism. The question remains as to whether a sensitive, dynamic and weak account of presupposition can be given entirely within the CCP model, or whether, as one might gather from a reading of Heim (1983), a successful account will inevitably depend on a van der Sandt-ian mixture of local, intermediate and global accommodation. In what follows I shall attempt to show that all the relevant desiderata, summed up in Table 2, can be satisfied within a pure CCP model.

Desiderata	Inference		Examples
Sensitivity	$\exists x a(x) \wedge \partial b(x)$	$\models \exists x a(x) \wedge b(x)$	(4)
Dynamism	$\partial(\exists x a(x)) \wedge b(x)$	$\models \exists x a(x) \wedge b(x)$	(5)
Weakness	$\forall x(a(x) \wedge \partial b(x)) \Rightarrow c(x)$	$\neq \forall x(a(x) \Rightarrow b(x))$	(7)
		$\neq \forall x((a(x) \wedge b(x)) \Rightarrow c(x))$	
	$\neg(\exists x a(x) \wedge \partial b(x))$	$\neq \forall x(a(x) \Rightarrow b(x))$	(8)
	$\exists x a(x) \wedge \partial b(x)$	$\neq \neg(\exists x a(x) \wedge b(x))$	
		$\neq \forall x(a(x) \Rightarrow b(x))$	(9)

Table 2: The Quantificational Projection Problem

3.2 Kinematic Predicate Logic

Lewis (1979) refers collectively to all the things we have to keep track of in a conversation as the *conversational score*. In this section we will be concerned with what Lewis calls the *kinematics* of such a multi-component score, although we will be restricting our attention to two components, one keeping track of the propositions that have been established (as in Section 2), and the other keeping track of the individuals that have been mentioned. The resulting *Kinematic Predicate Logic* (KPL) must satisfy not only presuppositional desiderata, but also the demands placed by discourse and donkey anaphora, and all in the spirit of aesthetic compositionality, where the form of the meaning evokes the same feelings as the form of the utterance.¹¹

To achieve this the Propositional Update Logic developed in Section 2 must be extended with the ideas in the Dynamic Predicate Logic (DPL) of G&S (1990): but we must tread carefully. As noted in G&S (1991b) there are some technical problems in effecting such a unification. These can be overcome if DPL is reformulated so that all formulae define updates, instead of the situation in G&S where requantification over a variable causes a downgrade, loss of information about the variables previous value. One way of removing downgrading is to adopt Paul Dekker's suggestion of switching from total variable assigns as in DPL to the partial variable assigns in his EDPL.¹² This will leave us with more or less what Heim (1983) had in the first place: a context is defined as a set of pairs of worlds and partial variable assignments, and a formula denotes a function from contexts to contexts. However, the combination of recent dynamic ideas will yield a more uniform treatment of worlds and variables, and also give us more of an overview of the options for a CCP treatment of presupposition. Let us begin by defining a logic (definitions 4, 5 and 6) which meets all the given desiderata, including all those in Table 2, and then briefly try to see where it diverges from Heim's account, as well as Cooper's and K&P's.

¹⁰I refer to the predictions of Heim's "raw", unaccommodating, CCP account because in Heim (1983) she discusses but does not fully explicate the accommodation mechanism she has in mind, so it is difficult to be sure what the predictions would be like. One imagines that she would want to derive more or less the same results as van der Sandt does in his theory, which contains a much more fully developed account of accommodation.

¹¹Kinematics may be just "dynamics for the upper classes" (an aspersion cast by Johan van Benthem during the colloquium), but at least it avoids terminological conflict with the predominant use of *dynamic semantics* to refer to G&S style treatment of anaphoric information and *update semantics* to refer to Veltman style treatment of worldly information.

¹²The interested reader is referred to Dekker's paper in this volume (Dekker 1992).

Definition 4 (Symbols for KPL)

A world, w , is a total function from n -ary predicates, R , to sets of n -tuples of individuals from some fixed domain; An assignment, f or g , is a partial function from variables, x_i , to individuals; A state, S , is a set of pairs of worlds and assignments such that all the assignments have the same domain of variables, V_s ; $g <_x f$ means f extends g wrt x but is otherwise identical; $g \geq f$ means g is an extension of f .

Definition 5 (KPL Semantics)

$$\begin{aligned}
 S[R(x_1, \dots, x_n)] &= \{ \langle w, f \rangle \in S \mid \langle f(x_1), \dots, f(x_n) \rangle \in w(R) \} \\
 &\quad \text{iff } \{x_1, \dots, x_n\} \subseteq V_s \text{ else undefined} \\
 S[\phi \wedge \psi] &\doteq S[\phi][\psi] \\
 S[\neg\phi] &\doteq S \setminus \{ \langle w, f \rangle \in S \mid \exists g \geq f, \langle w, g \rangle \in S[\phi] \} \\
 S[\exists x \phi] &\doteq \{ \langle w, f \rangle \mid \exists g <_x f, \langle w, g \rangle \in S[\phi] \} \text{ iff } x \notin V_s \text{ else undefined} \\
 S[\partial\phi] &= S[\phi] \text{ iff } \forall \langle w, f \rangle \in S, \exists g, \langle w, g \rangle \in S[\phi] \text{ else undefined} \\
 S[\phi \Rightarrow \psi] &\doteq S[\neg(\phi \wedge \neg(\psi))] \\
 S[\forall x \phi] &\doteq S[\neg(\exists x \neg\phi)]
 \end{aligned}$$

Definition 6 (Entailment in KPL)

$$\begin{aligned}
 \phi_1, \dots, \phi_n \models \psi \text{ iff } \forall S, w, f \quad &\text{if } \langle w, f \rangle \in S[\phi_1] \dots [\phi_n] \\
 &\text{then } \exists g \geq f, \langle w, g \rangle \in S[\phi_1] \dots [\phi_n][\psi]
 \end{aligned}$$

The definitions of predication and conjunction should be straightforward. Note that a predication is only defined on a context in which all of the predicated variables have already been introduced, so now undefinedness can be introduced in two ways: conventional presupposition failure and abuse of variables.

Negation is basically a set complement operation as before, but complicated by the possibility that the formula within the negation introduces new variables. The definition essentially says update the context with the formula, restrict all the assignment functions to their original values, and take the resulting context away from the original.

The existential definition is intuitively molecular, consisting of $\exists x$, which increments the context adding to every assignment function all possible x -extensions, composed with ϕ . It would be quite easy to define $\exists x$ as a formula in its own right, which could be conjoined with other formulae (eg. ϕ) compositionally. It is important that the domain of the function defined by $[\exists x \phi]$ contains only states which do not already provide valuations for x : implicit in the definition is the idea that existential quantification is treated as the introduction of a new discourse referent. Given that a discourse will start off without any referents having been introduced, it is natural to think of the state of zero information as the set of all pairs consisting of a world and an empty assignment.

Finally we consider the clause for the ∂ -operator. Firstly, note that the ∂ is made *dynamic* in a very simple way: provided certain conditions are met the update of a context with $\partial\phi$ is obtained by updating with ϕ . In a sense we *always* locally accommodate, rather than umm-ing and ah-ing about local accommodation as in van der Sandt or Heim's account. Note that accommodation itself does not challenge aesthetic compositionality: the challenge arises either from arbitrary application of local accommodation, or from intermediate accommodation which allows modification of bits of discourse representation arbitrarily far from the presupposition trigger.

The conditions on the ∂ -operator provide a check that when the context is updated with ϕ no worlds are lost. Consider the interpretation of $\exists x \text{ fat-man}(x) \wedge \partial(\exists y \text{ bike_of}(x, y)) \wedge \text{pushing}(x, y)$, from example (9). By the time $\text{fat-man}(x)$ has been interpreted, the context may contain several alternative values of x (corresponding to different assignment functions) for each remaining world. The ∂ clause will be undefined if any of the worlds have no value of x such that x is a bike owner. That is, for each world in the context set, there must be at least one fat man (value of x) who owns a bike. It is this condition that gives the definition its necessary *weakness* and *sensitivity*. Note that the "always locally accommodate" principle means that the outgoing context for the entire formula will only contain values of x which are bike owners: not only is it checked that there is at least one suitable fat man, but all the others are discarded.

The net effect of this definition is that a presupposition cannot carry any new factual information, which would involve discarding worlds from the context set, but can carry information about which individuals the speaker had in mind, and can introduce new individuals.

Definition 6 is a modified version of the first entailment definition in this paper (definition 3). It says one thing entails another if updating an arbitrary context with the first will take you to a context where updating with the second provides no new factual information and provides no further restrictions on the variables already introduced. The only complication is that the entailed formula can introduce new variables, which allows such inferences as $\exists x a(x) \models \exists y a(y)$.

3.3 Choices

What are the options in a CCP based account of the interaction between quantification and presupposition? To answer this question in full generality would obviously be difficult, since the options include not only different accounts of the semantics of presupposing constructions, but also any number of different treatments of quantification and connectives. Let us simplify, by assuming fixed definitions of quantifiers and logical connectives, namely those in the clauses of definition 5 apart from that for " ∂ ". Now we can ask the simpler question, what are the options for a definition of the ∂ -operator? In what follows I will sketch roughly where I believe the accounts of K&P, Cooper and Heim fit in to the KPL framework in terms of alternative definitions of ∂ . Regrettably I do not have the space here to explain their work in as much detail as it deserves, and for the moment I must refer the interested reader to the relevant papers in the bibliography.

Heim's account is very close to that presented here: she has essentially identical semantics for the quantifiers and junctions, although the technical elaboration differs somewhat. This makes it very easy to see how her treatment of presupposition fits in to the KPL picture. She defines presupposition not in terms of a unary operator, but as a relation between sentences and the propositions they presuppose: "S presupposes p iff all contexts that admit S entail p", where S is some sentence and p is a proposition. She then goes on to consider the interpretation of variables, and defines a context to be a set of pairs of worlds and partial assignment functions, as for KPL above. The immediate consequence of this is that a presupposition is only satisfied if updating the context with the presupposition would not remove any world-assignment pairs. Put another way, not only must the hearer know all the relevant facts, but she must also know which individuals the speaker had in mind, a strong condition.

Suppose we make the following definitions of relations between contexts S, S' :

$$S \sqsubseteq_w S' \text{ iff } \forall (w, f) \in S (\exists g (w, g) \in S'),$$

$$S \sqsubseteq_s S' \text{ iff } \forall (w, f) \in S (\exists g \geq f, (w, g) \in S').$$

Then the presupposition operator in definition 5 is given by: $S[\partial\phi] = S[\phi] \text{ iff } S \sqsubseteq_w S[\phi] \text{ else undefined}$. Heim's presupposition relation is equivalent to the definition of the following operator:

Definition 7 (A Heimian presupposition operator)
 $S[\partial_{\text{heim}}\phi] \doteq S \text{ iff } S \sqsubseteq_s S[\phi], \text{ else undefined.}^{13}$

The operator ∂_{heim} , unlike the ∂ of KPL, is static as there is no mechanism for a presupposition to introduce new variables. Of course, this fails to reflect the mechanism of accommodation that Heim envisages, which should introduce a sort of dynamism into the system by allowing global accommodation of discourse markers.

Both K&P and Cooper assume a treatment of quantifiers as binary relations rather than as one place predicates. This increases their range of choices for how quantifiers and presuppositions and quantifiers interact, allowing a presupposition to behave differently according to whether it occurs in the restrictor or scope of a quantifier, and making it difficult to capture their accounts as precisely as Heim's within the KPL framework.

Still, defining $W(S) = \{(w, \emptyset) \mid \exists f, (w, f) \in S\}$, a function which effectively strips a context of all its information about variables, we can parody the situation in K&P as being loosely equivalent to the definition of a unary presupposition operator: $S[\partial_{\text{K\&P}}\phi] = S \text{ iff } W(S)[\phi] = W(S) \text{ else undefined}$. This definition, by not allowing any flow of information about the values of variables between the presupposition and the assertion is static and insensitive as we should expect of an account where presuppositions and assertions are kept separate.

One of the main motivations of Cooper's account is to overcome the insensitivity of the K&P account. Strikingly, he observes regarding the George IV example, (4) above: "We need a semantics that will have the effect represented by... $\exists x [x \text{ succeeds George IV} \wedge \text{PRESUPPOSED difficult for } x \text{ to succeed George IV}]$ ". This, of course, amounts to a statement of the programme adopted in the current paper of defining a unary presupposition operator.

However, Cooper did not in fact pursue this line any further, and instead adopted an approach whereby predicates are given what he terms *paired intensions* in the model. The meaning of a predicate is defined as a function from worlds to a positive and a negative extension. This effectively yields a four valued logic in which he is able to reproduce the essential CCP inheritance properties while still allowing for a sensible account of binding of variables in the presupposition by

¹³It can be seen that ∂_{heim} is *strong* in the sense used to define the quantificational projection problem, whilst ∂ is *weak*. Note that we need not be committed to a unitary account of different presupposing constructions in terms of one or the other sort of operator: conceivably some constructions might have strong and others weak presuppositions, though this poses more questions than it answers. Which constructions should have which presuppositions and why? One can even conceive of single constructions having multiple types of presupposition. For instance, we could attribute to "managed to X" a weak presupposition that some effort had been made to do X, and a strong presupposition that there was difficulty involved in X. In that case "Somebody managed to sit on the bench" would presuppose that it would be difficult for anybody to sit on the bench, and that the person who sat on the bench made some effort in order to do so. I have not given this enough thought to make it a serious analysis of "manage" in its own right, I am merely trying to exemplify the fact that the choices available go beyond simply picking the single analysis of presupposition that produces the most sensible predictions in a majority of cases. The question of whether all this extra choice is actually relevant must await further empirical investigation.

quantifiers in the assertion. The binary quantifiers approach allows him, for example, to define the semantics of a universal “every X Ys” as presupposing that everything in the positive extension of X is either in the positive or negative extension of Y, effectively excessively weak regarding presuppositions of the restrictor, and strong regarding presuppositions of the scope. This has some interesting consequences: for instance the weakness of the restrictor means that the sentence “every person who read my PhD thesis laughed for days” would presumably not presuppose that I have a PhD thesis.

We could, again very loosely, view Cooper’s account in terms of some unary presupposition operator (although to do it justice we would need to define binary quantifiers): $S[\partial_{\text{cooper}}\phi] = S$ iff $S \sqsubseteq S[\phi]$, else undefined. Here I have not committed myself to exactly what $\sqsubseteq$ is, since it seems to flip between something like $\sqsubseteq_S$ and $\sqsubseteq_W$ according to the position of the presupposition trigger. At least it should be clear in this definition that ∂_{cooper} is static, there still being no way for a quantifier in the presupposition to bind a variable in the assertion, and sensitive.

It is important to realise that no claim is being made regarding the overall expressive power of the logics resulting from the alternative definitions of “ ∂ ”, although the logics could conceivably differ. The point is to see which definitions make for the most *natural* treatment of presupposition: more aesthetics!

4 Accommodation

4.1 The Accommodation Problem

Although it is not normally the primary function of presuppositions to inform, neither is it the case that we are incapable of learning anything from them. It is well known that presuppositions can be used with the obvious intention of providing new information. (If you did not already believe that, then you should do now that you have read the previous sentence.)

More generally, whatever the speaker’s actual intentions, when a speaker presupposes something of which a hearer is unaware, the hearer is not necessarily at a dead end. Lewis (1979) suggests that the hearer may make running repairs on his knowledge of the discourse context, by *accommodating* any missing information needed for interpretation to proceed smoothly. In what follows I will make some suggestions about how to formalise this process, but to simplify I will only consider accommodation at the propositional level, as a development of the Partial Update Logic presented in section 2 rather than the Kinematic Predicate Logic of section 3.

Yet even having allowed that accommodation occurs, there is a difficulty caused by the fact that people do not seem to accommodate the logically weakest proposition needed to make sense of an utterance. To understand this we must go beyond the simple principles of contextual entailment considered so far in this paper, and consider pragmatic issues determining the hearer’s model of the speaker.

4.2 The Logic of Epistemic Alternatives

Accommodation as Filtering

The *common ground* of the participants in a discourse situation can be seen as a partial model of reality, and we have seen how this can be modelled as a set of worlds which is gradually narrowed down. The speaker, unless possessed of extrasensory powers, can never be sure precisely what is the common ground. A certain amount of the common ground is established by the discourse itself, by a process like that described above. But it would be laborious to start every conversation with an assumption of zero common knowledge, and build up from scratch each time. Limitations of mortality and attention span mean that a certain amount of common ground must be assumed by the speaker.

But how does a hearer know what the speaker has assumed on a given occasion of utterance? He does not know, so he is forced to take into account his own uncertainty about the common ground. Thus we must model the hearer’s knowledge state not as one single partial model, but as a set of such partial models. The speaker may have assumed the common ground to be any one of these partial models. I will call them *epistemic alternatives*, and refer to propositions entailed by an epistemic alternative over and above those explicitly asserted in a discourse as *presumptions*.¹⁴

An utterance may or may not provide the hearer with information about the speaker’s presumptions. In the simplest case, when the utterance carries no awkward presuppositions, the hearer must update each of the epistemic alternatives with the new proposition. But when there is a presupposition, it may become clear to the hearer that some of the alternatives do not correspond to the speaker’s view of the common ground, and these alternatives must be discarded.

¹⁴Some — the die-hard semanticists — may like to think of these knowledge states as generalised quantifiers over worlds. It is interesting to observe that in G&S (1991a) a knowledge state is a generalised quantifier over assignments, but I shall not attempt to interpret this similarity here.

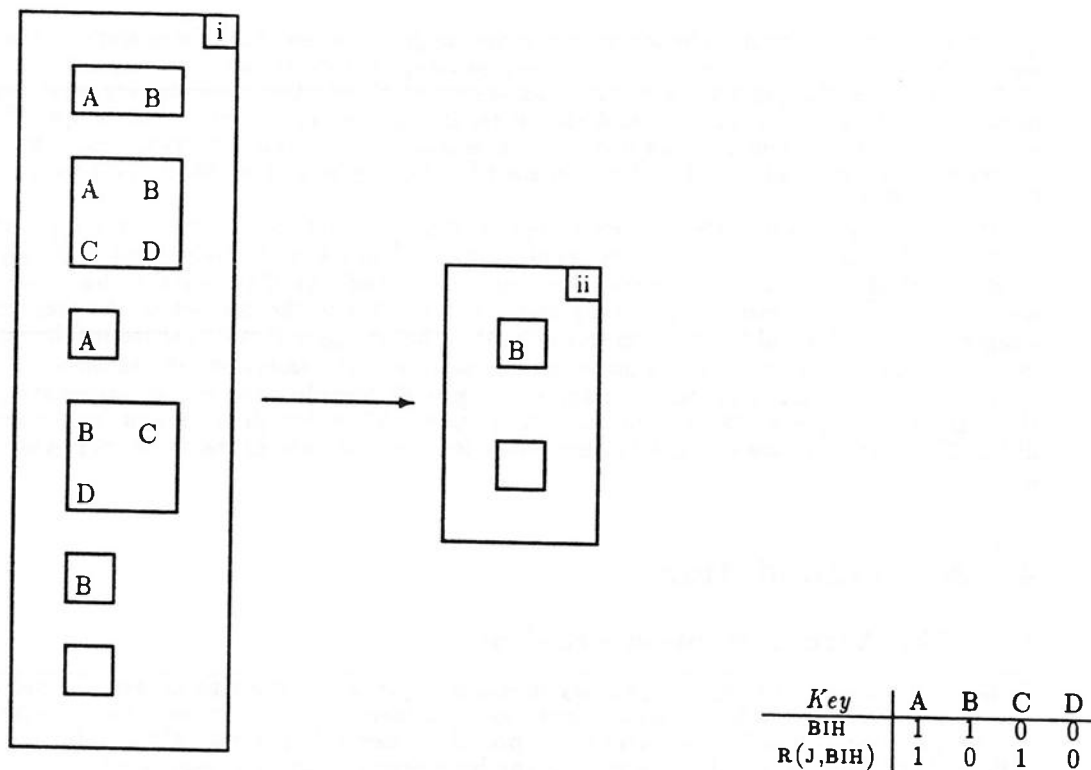


Figure 2: An accommodating update for $\neg(\partial\text{BIH} \wedge \text{R}(\text{J},\text{BIH}))$

Thus we arrive at a slightly different perspective from Lewis: *accommodation* is a form of filtering operation on epistemic alternatives, which perhaps this gives it less of the feeling of a repair strategy, and makes it seem more like part of the orderly process of communication.

In Figure 2 an initial hearer information state (i) is shown updated with the utterance "John doesn't regret that Bill is happy" to produce a new state (ii). State (i) consists of a fairly arbitrary set of epistemic alternatives, the six small boxes, each of which is characterised by a set of worlds. To simplify, the epistemic alternatives are restricted to combinations of four worlds, represented by the letters A-D, which differ only with respect to the two propositions BIH and R(J,BIH).

The topmost alternative, consists only of worlds where Bill is happy, so the hearer believes it possible that the Bill's happiness is being assumed to be common knowledge. This alternative is compatible with the presuppositions of the utterance, and updating with the utterance simply removes from this alternative any worlds where John has a negative state of mind regarding Bill's happiness. The result is the topmost epistemic alternative in state (ii).

The second alternative contains all the worlds under consideration, so before processing the utterance the hearer thought it plausible that there were no relevant presumptions. However, this alternative is incompatible with the presuppositions of the utterance, and is filtered out. A hearer who was certain that nothing had been taken for granted would soon run out of options.

The third alternative shows that before the utterance the hearer thought it possible that both Bill being happy and John having regrets were believed by the speaker to be common knowledge. This is incompatible with the assertion of the utterance, and after updating yields the empty, or contradictory, alternative. This takes us to a slight quirk in the formalisation I will present: the contradictory alternative is persistent, no matter what you learn, if you ever wondered whether the speaker thought the common ground was absurd, then you still wonder it. Furthermore it may be that you considered alternatives which in the light of what the speaker later said lead you to believe that maybe the speaker thought that the common ground was contradictory all along.¹⁵

¹⁵It would be possible to alter the model so as to remove the absurd alternative at every step, yet I think that this would miss an important problem which results from some simplifications I am making. The first simplification is one of logical omniscience. Although the common ground is partial, it is defined in terms of worlds which are total and consistent. There is no means to represent a common ground which is flawed in some domains, but consistent in others, characteristics which presumably typify the common ground of resource bounded agents like ourselves. One direction to take in order to remedy this would be to view propositions instead of worlds as the basic objects of the common ground, and not assume completion under logical closure, but from a technical point of view this would considerably complicate matters. The second simplification is that there is no mechanism for down-dating, and it is this that means that the common ground can never be repaired once it has reached a contradictory state. In a real discourse situation it may well be that the speaker recognizes that a hearer has contradictory beliefs and, for the sake of argument, goes along with those beliefs just so as to demonstrate their inconsistency. We will consider such a case later.

Rather than going through the remaining epistemic alternatives, here is a summary of what is happening in Figure 2: when the hearer updates with the new proposition, all the alternatives that include worlds where Bill is not happy (C or D worlds) are immediately filtered out, leaving only alternatives where Bill is happy, whilst the main propositional content of the sentence, concerning John's psychological state, removes from these remaining alternatives any worlds where John has regrets about Bill's happiness (that is, the A worlds).

Below a semantics is presented for the same propositional language as in section 2, but with sentences interpreted as functions between sets of epistemic alternatives.

Definition 8 (Symbols for Lifted Update Logic)

A world, w , assigns a boolean to every elementary proposition; an epistemic alternative, σ, τ , is a set of worlds; a state Σ is a set of epistemic alternatives; $\downarrow$ defines a function from a singleton set to its only element, otherwise undefined.

Definition 9 (Lifted Update Logic)

$$\begin{aligned}\Sigma[\phi] &= \{\sigma \mid \exists \tau \in \Sigma, \sigma = \{w \in \tau \mid w(\phi)\}\} \quad (\text{for atomic } \phi) \\ \Sigma[\phi \wedge \psi] &= \Sigma[\phi][\psi] \\ \Sigma[\neg\phi] &= \{\sigma \mid \exists \tau \in \Sigma, \sigma = \tau \downarrow (\{\tau\}[\phi])\} \\ \Sigma[\partial\phi] &= \{\sigma \in \Sigma \mid \{\sigma\}[\phi] = \{\sigma\}\} \\ \Sigma[\phi \Rightarrow \psi] &= \Sigma[\neg(\phi \wedge \neg\psi)]\end{aligned}$$

Such a semantics can be derived straightforwardly from the earlier semantics in definition 2, using a simple lift operation. For atomic propositions, the update is defined intersectively on each set of worlds which is an epistemic alternative within an incoming state Σ , and the set containing as its elements all the resulting sets of worlds is the output. I have simplified conjunction by defining it as functional composition at the level of sets of alternatives, rather defining it pointwise on each alternative. However, negation is defined as a lift from the earlier definition, becoming a set complement operation on each epistemic alternative.

The definition for ∂ shows how a test at the level of partial models becomes assertive at the level of epistemic alternatives — below this is put to use in the definition of an assertive *might* operator. $\partial\phi$ provides an eliminative update, removing all those epistemic alternatives which do not support the proposition ϕ . An immediate consequence of this is that there is no source of partiality in Lifted Update Logic: all formulae define total functions.

Our earlier notion of entailment for update logic would still make sense in this system, but we can also choose Veltman's third type of dynamic entailment. This says that updating the minimal information state (here the power set of the set of worlds) with the antecedent leaves us in a state to which the consequent adds nothing. This definition will preserve all the presupposition projection entailments listed in Table 1.

Definition 10 (Entailment in lifted update logic)

$$\phi_1, \dots, \phi_n \models \psi \text{ iff } \mathcal{P}(W)[\phi] \dots [\phi_n] = \mathcal{P}(W)[\phi] \dots [\phi_n][\psi]$$

The lifted semantics provides a new perspective on Veltman's *might* operator. Raising the operator to the new level yields:

Definition 11 (Lifted semantics of *might*)

$$\Sigma[\Diamond\phi] = \{\sigma \in \Sigma \mid \{\sigma\}[\phi] \neq \emptyset\}$$

When a speaker expresses $\Diamond\phi$ he is telling you (at least) that he doesn't know to the contrary of ϕ . The definition above says that in this situation the hearer is expected to update his knowledge state by discarding all models of the common ground which do not allow for the possibility of ϕ . At the level of partial models, there was simply no way of capturing the informational content of "epistemic-might", so perhaps the level of sets of sets of worlds is not quite so bizarre as it first seems.¹⁶

We still have not exhausted the information content of an utterance expressing epistemic possibility, for when a speaker says "Perhaps X", he conversationally implicates that he does not know X. This too may have an effect on the common ground, which could be modelled in the current system:

Definition 12 (Pragmatics of *might*)

$$\Sigma[\Diamond\phi] = \{\sigma \in \Sigma \mid \sigma \supset \downarrow (\{\sigma\}[\phi]) \supset \emptyset\}$$

¹⁶This is not to claim that the real meaning of "epistemic-might" should be in terms of the common ground: when somebody expresses $\Diamond\phi$ he is talking not about the common ground but about his own belief state. So the real semantics of $\Diamond\phi$ should be in terms of the speaker's attitude towards ϕ , not the speaker's attitude towards whether the common ground contains ϕ . None the less, an utterance of $\Diamond\phi$, if accepted, will also have the effect of constraining the common ground to contain some ϕ -worlds.

This definition says that when the speaker expresses $\Diamond\phi$, the hearer should discard not only the alternatives which contradict ϕ , but also those alternatives which positively affirm it.

One can imagine a similar treatment for the conversational implicatures associated with disjunction and conditionals. When, for instance, a conditional is uttered, it is implicated that the speaker is not certain about the truth or falsity of either the antecedent or the consequent, so perhaps epistemic alternatives which confirm or deny one or other of these should be discarded. Such a treatment will be assumed in Section 4.3, although in Section 4.4 a more sophisticated alternative will be considered.

4.3 What We Really Accommodate

One complaint that has been raised against the CCP theories of presupposition is that they frequently predict presuppositions that are too weak. Consider the contrast between the first and third of the following sentences:

- (10) a. If the walls are thin then it'll be annoying Bill that Mary is singing.
b. $TW \Rightarrow \partial MIS \wedge A(B, MIS)$
- (11) a. If the walls are thin then Mary is singing.
b. $TW \Rightarrow MIS$
- (12) a. If Mary is in the bath then it'll be annoying Bill that she is playing with his rubber duck.
b. $in_bath(m) \Rightarrow (\partial(\exists x duck(x) \wedge of(b, x)) \wedge \partial(play(m, x)) \wedge a(b, play(m, x)))$

On hearing the first sentence, I think I would normally conclude that Mary is singing. Similarly, on hearing the third sentence I would conclude that Bill had a rubber duck. However, regarding this latter example, I do not think I would come to the conclusion that Mary is definitely playing with Bill's rubber duck, although I would conclude that if she is in the bath then she is playing with his rubber duck. The problem with standard CCP models is that they lead us to expect a presupposition of a weak conditional in all these cases, so that Mary might not be singing and Bill might not have a duck. What is wrong?

I believe that the CCP predictions are, in a certain sense, correct. Belief in the conditional in the second sentence is necessary and sufficient for interpretation of the first sentence. If we knew that in fact Mary sings whenever she knows that the walls are thin — say because she is an operatic extrovert — then hearing the first sentence might *not* lead us to conclude that she is singing. Such a logically weakest precondition for interpretation I shall call a *linguistic presupposition*, “linguistic” because one has the feeling that it can be generated just by looking at the sentences linguistic form, and without knowing too much about what the non-logical parts of the sentence really mean. The real bits of information that people seem to accommodate when faced with a presupposition failure I shall call *cognitive presuppositions*, “cognitive” because their calculation can require arbitrary amounts of world knowledge. If there is such a thing as a grammar module in the brain, then it does not on its own calculate cognitive presuppositions. Having made this distinction we might, admittedly on somewhat shaky ground, go on to claim that Karttunen consistently predicts the correct linguistic presupposition.¹⁷

But is it enough that the CCP model gets the right linguistic presuppositions: surely it is the cognitive presuppositions in which we are interested? K&P offer an informal explanation of how extra implicatures can yield the requisite strengthening. I will suggest that the apparatus I have described, in the form of a lifted update semantics, has the potential to provide an adequate and formal version of this explanation.

Figure 3 shows how two states, (i) and (ii) evolve when updated with (10) and (11), where the epistemic alternatives under consideration are characterised in terms of the eight worlds A–H which differ with respect to the three propositions TW , MIS and $A(B, MIS)$. I assume that epistemic alternatives are filtered out not only because of presupposition failure but also as a result of conversational implicatures arising from the use of the conditional. In particular, saying “if A then B” typically implicates that neither A nor B is definitely known or definitely known to be false, so epistemic alternatives containing only A-worlds, or no A-worlds, or only B-worlds or no B-worlds, should be filtered out.¹⁸

State (i) contains four alternatives (discounting the absurd alternative, which I ignore here), corresponding to there either being no presumptions, or the presumption that Mary is singing if the walls are thin, or the presumption that Mary is singing, or the presumption both that the walls are not thin and that Mary is singing.

¹⁷My *cognitive presupposition* seems to be just what Stalnaker (1974) has in mind when he uses the term *pragmatic presupposition*: the reason I have refrained from using his terminology is that it would immediately suggest that what I have called *linguistic presupposition* was in fact *semantic presupposition*. Yet this identification might be misleading: the calculation of *linguistic presupposition* already involves what could be thought of as a partly pragmatic context-change based account of meaning. Having said that, I must confess that I find the identification with pragmatic and semantic presupposition rather appealing.

¹⁸For the case considered the implicature filtering could be achieved by defining the conditional as follows:
 $\Sigma[\phi \Rightarrow \psi] = \{\sigma \mid \{\emptyset\} \neq \{\sigma\}[\phi] \wedge \{\sigma\} \neq \{\sigma\}[\phi] \wedge \{\emptyset\} \neq \{\sigma\}[\psi] \wedge \{\sigma\} \neq \{\sigma\}[\psi]\} \cap \Sigma[\neg(\phi \wedge \neg\psi)]$

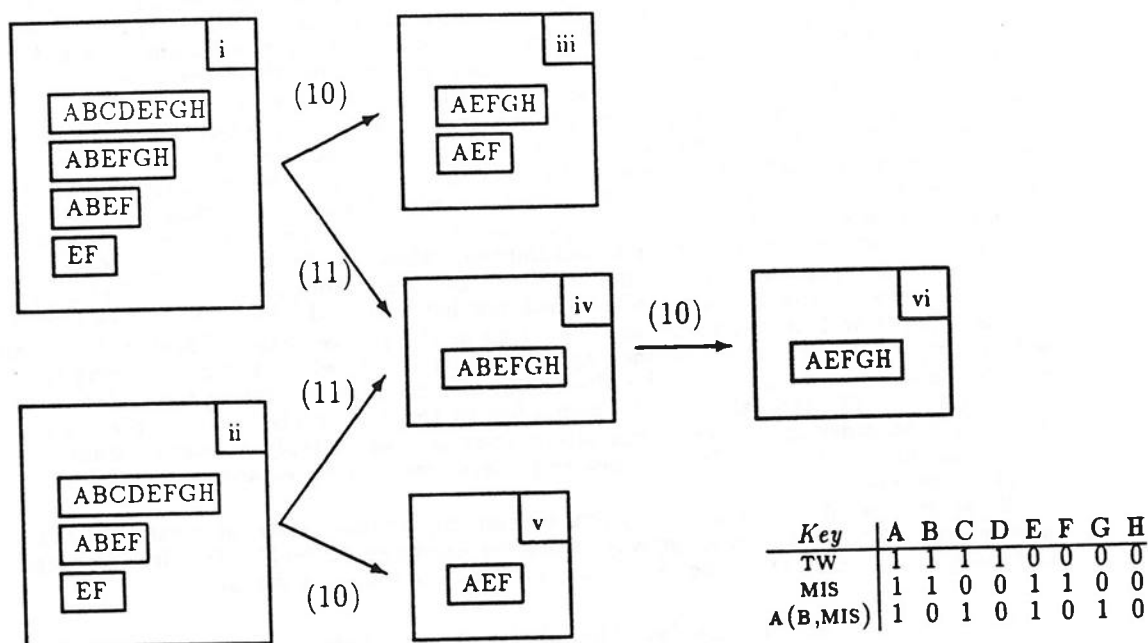


Figure 3: Updates with (10) $TW \rightarrow \partial(MIS) \wedge A(B, MIS)$ and (11) $TW \rightarrow MIS$

The topmost arrow in Figure 3 shows this state being updated with (10) to produce state (iii). This update removes alternatives where nothing is assumed (because of presupposition failure), and where the walls are known not to be thin (because of the implicature that the truth of antecedent of a conditional must be a live option.) This results in a state which is ambiguous between the speaker having assumed that everyone knew that Mary is singing, and having assumed that everyone knew that Mary is singing if the walls are thin.

A second path of two arrows shows state (i) being updated firstly with (11), yielding state (iv), then with (10) to produce state (vi). The only significant new element in this combined update is that the implicature from (11) that Mary's singing is in doubt removes the alternative where she is known to be singing. Thus the end state, (vi), shows unambiguously that the conditional $TW \rightarrow MIS$ is in the common ground, and this conclusion is arrived at without any presupposition triggered accommodation.

For a hearer to be in state (i) he must think it reasonable for a speaker to assume that it is common knowledge that Mary is singing if the walls are thin. Yet, for whatever reasons, Mary singing if the walls are thin is an odd thing to take for granted. State (ii) in the figure is the same as state (i) except the speaker is not allowing for the possibility that Mary singing if the walls are thin is being taken for granted. That is not to say that the proposition $TW \rightarrow MIS$ has been ruled out in state (ii). Updating (ii) with this unlikely fact is a perfectly consistent thing to do, and, as shown, results in state (iv) which, as was seen above, can itself be updated with $TW \rightarrow \partial(MIS) \wedge A(B, MIS)$ to produce state (vi). However, if state (ii) is updated directly with $TW \rightarrow \partial(MIS) \wedge A(B, MIS)$, as shown by the lowermost arrow in the diagram, the result is state (v) in which global accommodation of MIS has occurred.

To sum up, the states of lifted update logic can be used to distinguish propositions not only according to whether they are accepted, refuted or open, but also according to whether they are plausible presumptions. Furthermore, I believe it to be essential that plausibility rather than entailment is the central notion of an account of global accommodation, as is shown by (12). It seems clear that a hearer whose previous domain knowledge does not entail the proposition that Mary is playing with Bill's rubber duck if she is in the bath, can accept (12), and accept it without accommodation of the proposition that Mary is playing with Bill's rubber duck. And if the same hearer did not know that Bill had a duck, he can be expected to accommodate that proposition, and do so without even considering the possibility that Bill's duck ownership is conditional on Mary's bathing.¹⁹

This, of course, does not tell us much about why one proposition may be plausible as a presumption but another not. In its full generality, that would be a difficult problem to say the least, but perhaps an update approach can contribute at least something. One type of information that helps determine plausibility may be held in the form of default rules about the world, a subject

¹⁹It is worth pointing out that doubts have been cast on the factivity of the verb "to annoy", but I do not think that this is relevant. Firstly, even if it were not factive, there would still be a problem of explaining the contrast between what is globally accommodated in (10) and in (12). Secondly, the effects discussed above are quite robust, and still seem to hold in a case like "If Mary is in the bath, then her playing with the rubber duck is distracting her from singing." where the playing event is introduced as a definite description.

that is discussed in Veltman (1990). He defines a system in which an information state is a pair (S, ϵ) of a set of worlds and an ordering over those worlds. S encodes factual information in the same way as in his "might" system, or the related Partial Update Logic presented in section 2 of this paper; ϵ is used to encode default rules. For instance, if Martin Stokhof has a default rule to the effect that person X's paper will not be finished on time, then worlds in which X finishes on time will be lower in his ordering than worlds where X does not, and in the absence of factual information to the contrary, he presumes that come the deadline X's paper will not be finished.

How does this relate to the presence or absence of epistemic alternatives in the hearer's model of the common ground? Well, imagine that Jeroen is talking to Martin about X. Martin knows that Jeroen has recently seen X, but does not know what Jeroen is assuming Martin knows about X. Martin is entertaining a number of epistemic alternatives concerning Jeroen's picture of the common ground, and given Martin's presumptions about X, he will certainly favour epistemic alternatives in which the paper is not finished. So, even though the truth of this has not yet been established in the discourse, Martin will not bat an eyelid when Jeroen says that it's a pity that X's paper will not be ready in time to go in the proceedings. If Martin had maintained any epistemic alternatives where the paper was finished, or where the paper being finished was still in doubt, these would simply be filtered out as a result of processing the factive construction "pity that...". The general principle exemplified by this scenario is that our ordering over worlds will induce an ordering over epistemic alternatives, and so partially determine what information we can accommodate. In effect the hearer projects his own expectations onto his model of the speaker's beliefs.

However, it would be wrong to say that we can only accommodate information that could have been presumed to be the case anyway. Consider examples (13)–(15), the first of which is from Goldberg *et al* (1990), the paper which has inspired the following discussion:

- (13) Joe got married yesterday. The rabbi spoke very harshly.
- (14) Joe got married yesterday. The priest spoke very harshly.
- (15) ?Joe got married yesterday. The rabbi spoke very harshly. So did the priest.

On hearing that some person called Joe got married, I for one would not assume that there was a rabbi present, nor assume that there was a catholic priest. Yet both examples (13) and (14) are perfectly coherent pieces of discourse. In this case my default rules lead me to expect that there was just one person present who was in some way licensed to perform the ceremony, although even this might be over-ridden by an explicit statement to the contrary. But, in Veltman's terms, there will remain a number of *optimal* worlds, worlds in which there was a rabbi, a catholic priest, a captain of a ship, and so on. On the basis of my default rules alone, I will not be able to choose between these *default extensions* to the known information. So, I maintain epistemic alternatives corresponding to all the default extensions of the current knowledge state. Note, however, that as soon as I come across the definite description "the rabbi", I filter out all the other epistemic alternatives, and cannot easily process a discourse like that in example (15).

- (16) ?Seamus Murphy and Mary O'Docherty got married yesterday. The rabbi spoke very harshly.
- (17) Seamus Murphy and Mary O'Docherty, who are both Jewish, got married yesterday. The rabbi spoke very harshly.

This sort of reasoning may also explain why some people will find example (16) strange. If you, like me, do find it odd, then that could be because you have a default rule to the effect that people with Irish sounding names are not Jewish, so that after the first sentence you do not maintain an epistemic alternative involving a rabbi, and at the point you meet the definite your processing is interrupted. At this point the sort of accommodation you must perform in order to understand the discourse must be much more like a Lewis-style repair strategy than like the orderly process of narrowing down the assumptions that have been made we have been describing so far. Regarding this example, it is important to realise that the oddity does not arise simply from our being unable to accept that Seamus and Mary are Jewish, for then example (17) would also be an awkward discourse, which it is not. We can accept that Seamus and Mary are Jewish, but, without explicit information to that effect, they will not be Jewish in any of our default extensions, and this may explain why some speakers will find (16) strange but (17) acceptable.

4.4 Defeasibility and Changing Perspectives

It has often been suggested that presuppositions are defeasible. This might be motivated by cases like the following, where the presupposition is explicitly cancelled:

- (18) John doesn't *know* that Bill is happy. In fact Bill is not happy.

The linguistic community has to some extent split into two camps on this issue, a Karttunen influenced group who believe that the the major problem of presupposition theory is to account for when presuppositions are projected, and a Gazdar influenced group who think the major problem is to account for when presuppositions are cancelled.

I do not believe that a defeasible treatment of presupposition is incompatible with the general approach developed in this paper. To support this claim, which some may find surprising, I present in definitions 13–15, below, an update logic in which presuppositions introduce default rules, along the lines of the treatment of defaults in Veltman (1990).

Definition 13 (Symbols for Defeasible Update Logic)

An information state $\mathcal{I}$ is a pair $\langle \mathcal{I}_0, \mathcal{I}_1 \rangle$, where $\mathcal{I}_0$ is a set of worlds $w \in W$ and $\mathcal{I}_1$ is a preordering relation “at least as plausible as” on worlds; the optimal worlds in $\mathcal{I}$ are defined as the set $m_{\mathcal{I}} = \{w \in \mathcal{I}_0 \mid \neg \exists w' \in \mathcal{I}_0 (w', w) \in \mathcal{I}_1 \wedge \langle w, w' \rangle \notin \mathcal{I}_1\}$; the minimal information state $1 = \langle W, W \times W \rangle$; the absurd information state $0 = \langle \emptyset, \{\langle w, w \rangle \mid w \in W\} \rangle$.

Definition 14 (Defeasible Update Logic)

$$\begin{aligned} \mathcal{I}[\phi] &= \langle \{w \in \mathcal{I}_0 \mid w(\phi)\}, \mathcal{I}_1 \rangle \quad (\text{for atomic } \phi) \\ \mathcal{I}[\phi \wedge \psi] &= \mathcal{I}[\phi][\psi] \\ \mathcal{I}[\neg\phi] &= \langle \mathcal{I}_0 \setminus (\mathcal{I}[\phi])_0, (\mathcal{I}[\phi])_1 \rangle \\ \mathcal{I}[\partial\phi] &= \langle (\mathcal{I}[\phi])_0, \{\langle w, w' \rangle \in \mathcal{I}_1 \mid \\ &\quad ((\{w\}, \mathcal{I}_1)[\phi])_0 = \{w\} \rightarrow ((\{w'\}, \mathcal{I}_1)[\phi])_0 = \{w'\}\} \rangle \\ \mathcal{I}[\text{presumably}\phi] &= \mathcal{I} \text{ iff } (\langle m_{\mathcal{I}}, \mathcal{I}_1 \rangle[\phi])_0 = m_{\mathcal{I}} \\ &\quad \text{else} = 0 \end{aligned}$$

Definition 15 (Entailment in Defeasible Update Logic)

$\phi_1, \dots, \phi_n \models \psi$ iff $1[\phi] \dots [\phi_n] = 1[\phi] \dots [\phi_n][\psi]$

Here an information state consists of a pair the first member of which is a set of worlds, intuitively representing definitely established facts, and the second member of which is an ordering over worlds, encoding default extensions to the definite facts. I shall refer to the two components of an information state as the *established worlds* and the *default ordering* respectively.

The meaning of an atomic presupposition is defined intersectively on the established worlds, as earlier in this paper, and conjunction is defined as functional composition, again as earlier. Negation is a complement operation on the established worlds, combined with inheritance of any new defaults triggered in the argument.

The crucial clauses are those for ∂ and *presumably*. The ∂ clause, whilst reminiscent of Veltman’s *normally*, is importantly different in that it both asserts the truth of its complement and changes the default ordering so that worlds where the complement is untrue are at least as implausible as worlds where the complement holds. The factual assertion in the ∂ definition reflects the fact that presuppositions occurring in simple positive contexts are not normally defeasible: it is unnatural say “John knows that Bill is happy but Bill is unhappy.” Also ∂ is not subject to the syntactic restrictions on its occurrence which hold of Veltman’s *normally*: ∂ can occur arbitrarily deeply embedded within formulae. This reflects the fact that it makes sense for presuppositional defaults to be simply inherited, whereas this intuition would not hold of explicit default rules. The sentence “Bill is not normally happy” should not trigger a default rule to the effect that Bill should be expected to be happy. However, it does make sense for the first sentence in (18) “John doesn’t know that Bill is happy” to trigger such a default rule.

The clause for *presumably* ϕ provides a check that ϕ holds in the subset of the established worlds which are maximally compatible with the known defaults. This yields the following inference patterns:

1. $\phi \wedge \partial\psi \models \text{presumably}\phi$
2. $\phi \wedge \partial\psi \models \text{presumably}\psi$
3. $\neg(\phi \wedge \partial\psi) \models \text{presumably}\neg\phi$
4. $\neg(\phi \wedge \partial\psi) \models \text{presumably}\psi$
5. $\neg(\phi \wedge \partial\psi) \wedge \neg\psi \not\models \text{presumably}\psi$

I take it that this gives at least some substance to the claim made earlier in this paper that the inferences in Table 1, which I claimed characterised the projection problem, could be treated as being defeasible. This is reflected most obviously in the difference between the fourth and fifth of the inference patterns immediately above, where we see that addition of a flat contradiction of an earlier presupposition can non-monotonically remove previous entailments. Intuitively, this corresponds directly to the case in (18) above, where the first sentence alone would suggest that Bill was happy, but the combination of the two sentences does not.

It would be nice to present a similar account of the defeasibility of conversational implicatures. Unfortunately, DUL as presented above is not up to the task. Expression of a conditional "if A then B" implicates that neither A nor its negation is known (and similarly for B). However, this implication is cancellable: we can say things like "if A then B, and we all know whether A!", and of course an argument with the form of *Modus Ponens* will crucially involve the antecedent of a conditional being known information. So what is needed is a preference for default extensions which contain some worlds where the antecedent holds and some where it doesn't, and a logic for conversational implicature would involve an ordering not over single worlds but over epistemic alternatives, an idea introduced in Section 4.3, above. I leave such a development for another time.

Should simple cancellation examples like (18) lead us to abandon a logic of presupposition based on contextual entailment, like the earlier PUL, KPL and LUL systems in this paper, in favour of a default logic like DUL? I think we should be wary of jumping to conclusions without much closer scrutiny of the data. Consider the following examples, the first two of which seem to involve presupposition cancellation:

- (19) John doesn't *know* that Bill is happy, since Bill is not happy.
- (20) John doesn't *know* that Bill is happy, he merely *believes* it.
- (21) John doesn't *think* Bill is happy, he is *totally convinced*.
- (22) John doesn't *believe* that Bill is happy, he *knows* it.

My feeling is that examples like (19) and (20) pattern naturally with (21) and (22). The first thing to note is that all these examples carry "marked" stress, whether contrastive or not, so in principle there might be something to divide them from the cases considered elsewhere in this paper. More importantly, examples (21) and (22) suggest that negation can have some interesting non-truth functional effects. It is clear that John knowing something *entails* his belief in it, and John being totally convinced about something *entails* that he thinks it true, so that the negative particles in (21) and (22) must mark something other than the fact that the relevant clauses are untrue. I would suggest that whatever the negative particle is doing in these two examples, it is doing the same in (19) and (20). So perhaps these classic so-called cases of *presupposition cancellation* are not merely peculiarities for the presupposition theorist to deal with, but part of a much wider problem for truth-conditional accounts of negation. This is not to say that examples (19)-(22) in any way count against a default theory of presupposition, merely that they are not in themselves a sufficient motivation.

The following examples *prima facie* constitute a problem for the non-default CCP model developed in this paper, and have often been taken as motivating a default account like DUL:

- (23) You say that someone in this room loves Mary. Well maybe so. But it certainly isn't Fred who Loves Mary. And it certainly isn't John.... (*after enumerating people in the room.*) And that's everybody in the room!
- (24) If I realize later that I have not told the truth, then I will confess it to everyone.
- (25) I was not aware that you are allowed to do that! (*Said eg. by a teacher to a child who is smoking behind the bicycle sheds.*)

In the first of these examples (adapted slightly from an example due to Ed Keenan) the cleft sentences all presuppose that someone loves Mary, and this is apparently cancelled by sheer weight of logical argument. In the second (due to Gazdar) and third examples conversational implicatures seem to cancel out respectively the presuppositions that I have not told the truth, and that you are allowed to do whatever it is.

Yet all three examples could be interpreted in terms of an interesting form of accommodation, which I will call *temporary global accommodation*. The idea is that in these cases a sort of perspective shift occurs, whereby the speaker temporarily assumes a common ground (normally an apparent belief state of one of the other conversational participants) that he believes to be false. This is clear in (23), which has the form of a *reductio ad absurdum* argument. The speaker announces with his "well maybe so" that one could take a perspective where somebody loved Mary. He then implicitly presumes the proposition to be true in order to demonstrate its inconsistency with other known facts. The same can occur explicitly, with a statement like "let us assume that somebody loves Mary".

Similarly, it might be claimed that in (24) the speaker is implicitly taking the perspective of somebody who thinks that the speaker is lying, and that in (25) the speaker is taking the perspective of someone who believes that whatever is being done is allowed. All of this is rather tentative, and I do not wish it to be taken as an attack on default accounts of presupposition. On the contrary: I am not attempting to deny that presuppositions are defeasible, but to begin to explain this defeasibility in terms of underlying principles of communication. Yet there is much work to do. Temporary global accommodation is a dangerously powerful mechanism, and a satisfactory account of presupposition would have to involve considerable constraints on its application. My intuition is that all cases of temporary global accommodation will be highly marked. After all, if a speaker could simply change perspective in mid-flow without giving some

hint of what was happening, there would be little hope of ever providing a formal model of meaning! At a minimum, one should expect a speaker to maintain a constant perspective unless there is a salient alternative perspective to which he can shift.

4.5 On the Accommodation of Representations

Temporary global accommodation can have much the same effects as a combination of local and intermediate accommodation in a representationalist account like that of van der Sandt. Local accommodation would lead to addition within the representation of each of the negations in (23) of the proposition that somebody loves Mary, and addition within the negation in (25) of the proposition that whatever was being done was allowed, whilst intermediate accommodation would lead to addition within the antecedent of the conditional in (24) of the proposition that the speaker lied.

Indeed, a theory involving a judicious mixture of local, intermediate and global accommodation is a force to be reckoned with, and I think it fair to say that van der Sandt's theory, which crucially involves all three types of accommodation, is the most empirically adequate of any theory of presupposition currently around. Yet even his account is not perfect! Consider the following:

- (26) If John's unhappy, then it's a pity that both John and Mary are unhappy.
(27) ?I don't know whether Bush is ill. However, every Republican regrets that Bush is ill.

I think van der Sandt would admit that his theory as it currently stands makes the wrong predictions about both of these. On hearing (26) most people would conclude that Mary is unhappy. Yet van der Sandt would I think predict, after accommodation into the antecedent of the conditional, a meaning something like "If John and Mary are unhappy, then it's a pity." The second discourse is intuitively very odd. Yet van der Sandt would predict accommodation of the fact that Bush is ill into the restrictor of the universal, yielding a meaning for the second sentence logically equivalent to "If Bush is ill then every Republican regrets it." This is a perfectly sensible meaning for a sentence to have, but it is not the meaning of the second sentence in (27).

On the other hand, these examples should not be problematic for the CCP model developed in this paper, since in that there are no notions of local or intermediate accommodation. Yet the CCP model developed here is not incompatible with there being a level of discourse representation on which local and intermediate accommodation could operate. But then the processes of local and intermediate accommodation would be quite distinct from the mechanism of global accommodation as I have described it.

Global accommodation, as has been suggested, should be seen as a natural part of the orderly communication process in which conversational participants gradually determine their common ground. On the other hand, there may be times when what is said does not appear compatible with any assumption of common ground. Then the only alternative for a hearer will be to assume that the speaker did not really mean what he said, so the hearer will begin a non-compositional tampering with his representation of the meaning of the speaker's utterance, adding in things that the hearer seems to have omitted to mention but obviously meant. Yet this last process is complicated, and that should explain the observation of both Heim and van der Sandt, which goes unexplained in their work, that global accommodation is always preferred to local accommodation.

Incidentally, within the theories of van der Sandt or Heim it would seem considerably less natural to assert that local and intermediate accommodation are fundamentally difficult processes from global accommodation, since in their theories some examples which are obviously very natural and not apparently difficult to process require local or intermediate accommodation. This is the case, for instance, with example (6) above ("If John is married then his wife is very tolerant"). In van der Sandt's account local accommodation of a discourse marker has to occur, together with a condition on the identity of that marker, whereas in KPL the dynamism of the presupposition operator does the work.

5 Conclusion

Concerning the interaction between quantification and presupposition, we have seen that a range of examples that had seemed awkward for CCP accounts, but which have been treated successfully in van der Sandt's theory, can in fact be treated in a pure CCP model without invoking local or intermediate accommodation. However, there is much more data to be looked at, and the simple first-order treatment of quantification assumed is obviously inadequate.

A common fault of CCP theories is that they predict presuppositions that are too weak. It was suggested that the key to some of these difficulties might lie in a formalisation of how conversational participants maintain sets of plausible epistemic alternatives. This began to take us in the direction of a formal theory of pragmatics unifying the treatment of presupposition and implicature.

I tried to show that Gazdarian default accounts of presupposition are not so distant from CCP accounts as they might seem. The moral is that the approach of projecting weak non-defeasible

presuppositions which are then strengthened using implicature-triggered defaults, is similar in effect to producing a strong but cancellable presupposition in the first place, and either method is compatible with a dynamic view on interpretation. We thus saw two different ways of introducing defeasibility into an account of presupposition: our default expectations about plausibility can determine the inferences we make in deciding what to accommodate, and presuppositions can themselves be treated as default rules.

I have contended that global accommodation should be viewed not as a repair strategy on discourse representations, but as addition of information concerning the common ground. This eventually led to the possibility that local and intermediate accommodation of presuppositions, if these types of accommodation occur at all, may be quite different in nature from global accommodation.

It should be clear that the full potential of CCP based accounts of presupposition has not yet been realised, although after a somewhat dormant decade some might have thought that the CCP paradigm had been exhausted. It seems to me that there are several promising areas requiring both further technical development, and a deeper exploration of relevant data.

References

- Cooper, R. (1983) *Quantification and Syntactic Theory*. Dordrecht: D. Reidel.
- Dekker, P. (1992) An Update Semantics for DPL. In P. Dekker and M. Stokhof, eds., *Proceedings of the 8th Amsterdam Colloquium*, University of Amsterdam.
- Gazdar, G. (1979) *Pragmatics, Implicature, Presupposition, and Logical Form*. New York: Academic Press.
- Goldberg, J., L. Kalman and Z. Szabo (1990) External Presuppositions Versus Discourse Presuppositions. Paper presented at a workshop on Presupposition, Lexical Meaning and Discourse Processes of ESPRIT Working Group 3315.
- Grice, P. (1989) Presupposition and Conversational Implicature. In *Studies in the Way of Words*, pp. 269-282. Cambridge, Mass.: Harvard University Press.
- Groenendijk, J. and M. Stokhof (1990) Dynamic Predicate Logic: Towards a compositional, non-representational semantics of discourse. In J. van Benthem, ed., *Partial and Dynamic Semantics I*, pp. 53-108. DYANA Report R2.1.A, Centre for Cognitive Science, University of Edinburgh.
- Groenendijk, J. and M. Stokhof (1991a) Dynamic Montague Grammar. In M. Stokhof, J. Groenendijk and D. Benveniste, eds., *Quantification and Anaphora I*. DYANA Report R2.2.A, Centre for Cognitive Science, University of Edinburgh.
- Groenendijk, J. and M. Stokhof (1991b) Two Theories of Dynamic Semantics. Technical Report. University of Amsterdam.
- Heim, I. (1983) On the Projection Problem for Presuppositions. In M. Barlow, D. P. Flickinger and M. T. Wescoat, eds., *West Coast Conference on Formal Linguistics*, pp. 114-123, Stanford, Stanford.
- Karttunen, L. (1973) Presuppositions of compound sentences. *Linguistic Inquiry* 4, 169-193.
- Karttunen, L. (1974) Presuppositions and linguistic context. *Theoretical Linguistics* 1, 181-194.
- Karttunen, L. and S. Peters (1979) Conventional implicature. In C. Oh and D. Dinneen, eds., *Syntax and Semantics*, Vol. 11: *Presupposition*, pp. 1-56. New York: Academic Press.
- Langendoen, D. T. and H. B. Savin (1971) The Projection Problem for Presuppositions. In C. J. Fillmore and D. T. Langendoen, eds., *Studies in Linguistic Semantics*. New York: Holt, Rinehart and Winston.
- Lewis, D. (1979) Score-keeping in a language game. In R. Bauerle, U. Egli and A. von Stechow, eds., *Semantics from Different Points of View*. Berlin: Springer-Verlag.
- Mercer, R. E. (1988) Solving some persistent presupposition problems. In *Proceedings of the 12th International Conference on Computational Linguistics*, pp. 420-425, Budapest.
- van der Sandt, R. (1990) Anaphora and Accommodation. Paper presented at a workshop on Presupposition, Lexical Meaning and Discourse Processes of ESPRIT Working Group 3315.
- Stalnaker, R. C. (1974) Pragmatic presuppositions. In M. K. Munitz and P. K. Unger, eds., *Semantics and Philosophy*, pp. 197-213. New York: New York University Press.
- Stalnaker, R. C. (1978) Assertion. In P. Cole, ed., *Syntax and Semantics*, Vol. 9: *Pragmatics*, pp. 315-332. New York: Academic Press.
- Veltman, F. (1990) Defaults in Update Semantics. In H. Kamp, ed., *Conditionals, Defaults and Belief Revision*, pp. 28-64. DYANA Report R2.5.A, Centre for Cognitive Science, University of Edinburgh.
- Zeevat, H. (1991) Aspects of Discourse Semantics and Unification Grammar. Ph.D. thesis, University of Amsterdam.

Comparative Quantifiers

Filippo Beghelli
Linguistics Dept., UCLA

Introduction

We study the semantics of quantity comparatives such as **more-than**, **fewer-than**, **exactly as many-as**, etc., as they occur in the following syntactic environments:

- (1) a. more students read than write (N', (VP, VP))
 b. more students than teachers read ((N', N'), VP)
 c. more students read than teachers write ((N', VP), (N', VP))

In (1a) **more-than** co-occurs with a common noun (*student*) and two tensed VPs (*read*, *write*) forming an S. The VPs are understood to predicate of the objects denoted by the common noun – e.g., the common noun must satisfy the selectional restrictions of both VPs (cf. Keenan 1984, 1987b)¹.

Semantically we ignore (here) tense and aspect and treat both common nouns and VPs as denoting properties of individuals. We write P for the set of properties (so a member q of P is, to within isomorphism, a set of individuals), $[A \rightarrow B]$ for the set of functions from A into B , and 2 for the set $\{\text{True}, \text{False}\}$ of truth values. We treat **more-than** in (1a) as denoting in $[P \rightarrow [PxP \rightarrow 2]]$. That is, it maps a property (*student*) to a function mapping a pair of properties (*read*, *write*) to a sentence denotation. So, **more-than** in (1a) is of type $\langle 1, \langle 1, 1 \rangle \rangle$, or $\langle 1, 1^2 \rangle$ for short (cf. Lindström's (1966)).

In (1b), **more-than** co-occurs with two N's and a tensed VP. So is of type $\langle 1^2, 1 \rangle$ ($= \langle \langle 1, 1 \rangle, 1 \rangle$), denoting in $[PxP \rightarrow [P \rightarrow 2]]^2$. In (1c) **more-than** co-occurs with two pairs, each consisting of a N' and VP which predicates of it. It denotes in $[[(PxP)x(PxP)] \rightarrow 2]$, and has type $\langle 1^2, 1^2 \rangle$.

Informally speaking, we say that **more-than** in (1c) has two "heads" and two "tails", in (1b) two heads and one tail, and in (1a) one head and two tails.

This paper is presented in three parts: Part 1 overviews the range of expressions that occur in each of the three syntactic environments given in (1). We observe a significant overlap, but also some differences. Part 2, the major part of this paper, presents a number of semantic generalizations which hold of the types of quantity comparatives presented in Part 1. And Part 3 presents a semantically based proposal to account for the syntactic and semantic similarities among the different types of comparatives studied. Our proposal treats **more-than** and other such comparatives which occur in all three contexts as denoting just one function, one whose arguments are determined by the syntactic context in which it occurs. This analysis contrasts –

at least in spirit – with syntactic analyses that derive all the comparative expressions in (1) from the same source, a source that most closely resembles our type $\langle 1^2, 1^2 \rangle$.

Part 1: Overview of the determiners in the new types

For reference we show in A. - G. below the range of quantity comparatives which occur in at least one of the syntactic environments in (1):

- A. fewer-than, (at least) as many-as, exactly as many-as, at most as many-as, many more-than, almost as many-as, ...
- B. a greater number of-than, a smaller (lesser) number of-than, an equal number of-as, the same number of-as, an at least equal number of-as, ...
- C. at least two more-than, exactly two more-than, between two and six more - than, three times as many-as, a twice greater number of-than, , (at least) twenty percent more-than, exactly fifty percent more-than, more than twenty more-than, ...
- D. proportionally more-than, a greater percentage of-than, ...
- E. more of John's-than, fewer... at John's party-than, more French-than, ...
- F. not more-than, no fewer-than, at least ten more but no more than thirteen more-than, at least as many or a greater number of-than, no more than two more-than, two more French and six fewer Dutch-than, ...
- G. the same-as, different-than, whatever-also

A exemplifies the simplest and most basic comparatives of quantity, what we may call classic comparatives. Essentially synonymous, though syntactically more complex expressions are given in B. C includes numerically modified comparatives, a type of construction which is very productive in English. In E we find other examples of comparative + modifier (e.g., adjectival phrases, locative PPs and possessive phrases). Motivation for including the modifier as part of the determiner comes in part from the existence of conjoined comparatives such as "more of John's but fewer of Mary's students came early than left late", which are members of the next group, F. F includes boolean combinations of comparatives – numerically and otherwise modified ones – which show that complex comparatives can be built out of others of the same type by means of conjunction, disjunction, and negation.

We consider first which of these expression occur in the environment (1a) – that is, which can replace more-than in "more students read than write". The answer is that essentially all the sorts of expressions above occur in this environment. A further group of expressions that occur naturally in the syntactic context of (1a) is G; for example:

- (2) The same students came early as left late
- (3) Different students came early than left late
- (4) Whatever students came early also left late

The comparatives in D we call proportionality comparatives. Note that a sentence like:

- (5) ?Proportionally more students came early than left late

is semantically odd, possibly because it merely means "more students came early than left late". Some other proportionality comparatives in this syntactic environment sound less bad, though the same anomaly is present:

- (6) (?)A greater percentage of students came early than left late

As for the environment ((N', N'), VP) as in (1b), all groups except G are in its range. Group G expressions are decidedly bad in this type:

- (7) *The same students as teachers came early

- (8) *?John interviewed different students than teachers

The last sentence shows G expressions are also bad in object position. These sentences are out, apparently because we find unnatural to specify the same set twice in different ways³. Regarding group E, one reason for considering the modifier as part of the determiner here is that the more natural reading for these comparatives is the one where the modifier distributes. So "more of John's plays than poems" preferably means "more of John's plays than of John's poems", though the more restricted reading (where only the head adjacent to the modifier is affected) is sometimes possible.

Third, we consider the environment ((N', VP), (N', VP)). Again, group G generally produces unacceptable results:

- (9) *The same students came early as teachers left late

is somehow a nonsense⁴. Other restrictions in this type are with group E: the readings where the modifier distributes are usually not available. So in "?more of John's students came early than teachers left late" or in "more French students came early than teachers left late" the natural readings are not those where "John's" or "French" are understood to apply also to "teachers". This is unlike the $\langle 1^2, 1 \rangle$ case, where the modifier most naturally distributes over the two N's. This fact is responsible for extremely marginal or plainly ungrammatical results in type $\langle 1^2, 1^2 \rangle$, as shown by the following pairs, the first a $\langle 1^2, 1 \rangle$, the second a $\langle 1^2, 1^2 \rangle$ construction:

- (10)a. Fewer of John's cats than dogs were inoculated

- b. *?Fewer of John's cats were inoculated than dogs taken to the pound

- (11)a. Ten more dogs than cats were inoculated

- b. ??Ten more dogs were inoculated than cats were captured

This observation shows that comparatives of type $\langle 1^2, 1^2 \rangle$ are more restricted than the other subtypes. Often, acceptable expressions in this type are somewhat less acceptable than their corresponding ones in the other environments.

Generalizations on the distribution of comparatives

- 1) The presence of numerals in types $(N', (VP, VP))$ and $((N', N'), VP)$ as in (12a, b) below guarantees that the number of expressions occurring in these syntactic environments is infinite.

- (12)a. Ten more students came early than left late
b. Ten more students than teachers came to the party

This argument can be extended to expressions of type $((N', VP), (N', VP))$, although we note (as we did in (11b)) that numerically modified comparatives in this type are often marginal:

- (13) ?Ten more students came early than teachers left late

The classes of adjectivally, prepositionally, or otherwise modified comparatives, at least in types $(N', (VP, VP))$ and $((N', N'), VP)$, are also recursive. We see then that the class of quantity comparatives is open, in distinction to e.g. the set of modals and auxiliary verbs in English.

- 2) Expressions of type $(N', (VP, VP))$ enjoy the greatest freedom, then come the $((N', N'), VP)$ ones, and last the $((N', VP), (N', VP))$.

These syntactic observations argue against an analysis of quantity comparatives on which expressions of types $\langle 1, 1^2 \rangle$ and $\langle 1^2, 1 \rangle$ would be derived from $\langle 1^2, 1^2 \rangle$ ones. We note as well that much of the appeal of deriving (14a-b) below from (14c-d) respectively is that it provides an account of the meaning of (14a-b) in terms of that of (14c-d):

- (14)a. More students came early than left late
b. More students than teachers came early
c. More students came early than students left late
d. More student came early than teachers came early

But this relationship is not constant: the pairs in (15-17) differ semantically with respect to modifier scope and binding:

- (15)a. More students than teachers read some book
b. More students read some book than teachers read some book
(16)a. More boys than girls got an 'A' in their class
b. More boys got an 'A' in their class than girls got an 'A' in their class
(17)a. More friends of theirs admire Mary and Mitzi than envy Jill and Jane
b. More friends of theirs admire Mary and Mitzi than friends of theirs envy Jill and Jane

Part 2: Semantic generalizations

The quantity comparatives studied here all relate in various ways common noun and VP denotations. In this respect they are like ordinary dets such as *all*, *most*, etc. In "most senators snore" *most* is of type $\langle 1, 1 \rangle$, mapping the pair $\langle \text{senator}, \text{snore} \rangle$ of properties to a truth

value. It turns out that many of the characterizations developed in the study of $\langle 1,1 \rangle$ quantifiers extend to quantity comparatives. We review these below:

Review of the basic properties of standard quantifiers (type $\langle 1,1 \rangle$)

First, as is well known, dets of type $\langle 1,1 \rangle$ do not denote freely in $[P \rightarrow [P \rightarrow 2]]$, but they only denote conservative functions. A function f is conservative if the truth of f applied to a pair of properties (p, q) depends only on p and $p \wedge q$, i.e., on which individuals are p 's and which of these are also q 's. Conservativity, thus, says that the intersection of q and the complement of p is irrelevant for the truth of $f(p)(q)$.

Def: Conservativity for $\langle 1,1 \rangle$ functions. A function $f : P \rightarrow [P \rightarrow 2]$ is conservative iff for all p, q, q' in P , if $(p \wedge q) = (p \wedge q')$ then $f(p)(q) = f(p)(q')$

We write $CONS_{\langle 1,1 \rangle}$ for the set of conservative functions of type $\langle 1,1 \rangle$. Complex dets in type $\langle 1,1 \rangle$ can be built from others by means of the boolean operations of conjunction, disjunction, and negation: for instance, from dets such as *most*, *all*, *some*, *Mary's*, *John's*, we build dets such as *most but not all*, *some of Mary's* and *most of John's*, etc. $CONS_{\langle 1,1 \rangle}$ provides denotations for these complex dets since it is (provably) closed under the boolean operations $\wedge$ (meet), $\vee$ (join), and complements.

Second, many type $\langle 1,1 \rangle$ dets in English satisfy a condition stronger than mere conservativity: they are intersective. The value of an intersective function depends only on the meet (intersection) of the pair of arguments (p, q) . To determine the truth value of an intersective function $f(p)(q)$ one only has to check which individuals are both p 's and q 's. The notion of intersectivity generalizes and extends Barwise-Cooper's (1981) notion of weak determiner. Examples of intersective determiners in English are *some*, *no...but John*, *at least five*, *between two and six*, etc. *all*, *most*, *both-and*, *John's*, *all but five*, on the other hand, are of non-intersective.

Def: Intersectivity for $\langle 1,1 \rangle$ functions: f in $[P \rightarrow [P \rightarrow 2]]$ is intersective iff for all p, q, p', q' in P , if $(p \wedge q) = (p' \wedge q')$ then $f(p)(q) = f(p')(q')$

The set $INT_{\langle 1,1 \rangle}$ of intersective functions⁵ of type $\langle 1,1 \rangle$ is a proper subset of $CONS_{\langle 1,1 \rangle}$. It is closed under the boolean operations in $CONS_{\langle 1,1 \rangle}$ and thus a proper subalgebra.

Recall that NP's built from intersective determiners occur naturally in Existential There contexts, while non-intersective ones are problematic there.

Third, most intersective dets are actually cardinal: they depend only on the cardinality of $p \wedge q$. It appears that all the lexically simple intersective determiners in English are cardinal. Examples of cardinal dets in type $\langle 1,1 \rangle$ are *some*, *no*, and numerals or numerical dets: *at least two*, *just finitely many*, *exactly ten*, *at most five*, *between two and six*, etc.

Def: Cardinal $\langle 1,1 \rangle$ functions. A function $f : P \rightarrow [P \rightarrow 2]$ is cardinal iff for all p, q, p', q' in P , if $|p \wedge q| = |p' \wedge q'|$ then $f(p)(q) = f(p')(q')$

Again, $CARD_{\langle 1,1 \rangle}$ is a (complete, atomic) subalgebra of $INT_{\langle 1,1 \rangle}$.

1. Conservativity

We have so far characterized the sets in which comparatives of each syntactic type denote as including functions from triples (or quadruples) of properties into 2. But empirical inspection of the functions denoted by the expressions in A - G shows that they do not denote freely in these sets. Rather, for each type T, comparative expressions of type T only denote functions which meet a condition analogous to conservativity for type $\langle 1,1 \rangle$ determiners.

For a function in the new types to be conservative means that all we need to know in order to determine its truth value is which individuals have the common noun property(s) and which of these have the corresponding VP property(s). To take the most complex case, $((N', VP), (N', VP))$, to know the truth of "more students read than teachers write" one needs to know at most which individuals are students, which teachers, which of the students read, and which of the teachers write. This is correct, and shows that the function denoted by *more-than* in type $\langle 1^2, 1^2 \rangle$ is conservative.

The same holds for the other two types: to know the truth of "more students read than write" one only needs to check at most which individuals are students and read, and which students write, overlooking the individuals who read or write but are not also students. And to determine the truth of "more students than teachers read" one at most has to check which individuals are students or teachers and which of those individuals also read.

That quantifiers in the new types are conservative is perhaps surprising considering that the comparative builds an actual NP only in the case of type $((N', N'), VP)$ - i.e., we have $[_{NP} \text{ more students than teachers }] [_{VP} \text{ read }]$. In the other two types there are no NP's formed by applying the comparative quantifier to its common noun(s), hence, no generalized quantifiers in the terminology introduced by Barwise-Cooper (1981).

The formal definitions of conservativity are given in Appendix A1. We illustrate the $\langle 1, 1^2 \rangle$ case: a function $f : P \rightarrow [PxP \rightarrow 2]$ is conservative iff for all p, q_1, q_2, q_1', q_2' in P , if $(p \wedge q_1) = (p \wedge q_1')$ and $(p \wedge q_2) = (p \wedge q_2')$ then $f(p)(q_1, q_2) = f(p)(q_1', q_2')$.

The observation that English determiners denote conservative functions can thus be extended from ordinary (type $\langle 1,1 \rangle$) determiners to the comparatives we are studying, which we can now call comparative determiners. We conclude that the denotations for English determiners, in all types, are conservative:

Generalization 1: Dets of type T, for T in $\{\langle 1,1 \rangle\} \cup \{\langle 1^2,1 \rangle, \langle 1,1^2 \rangle, \langle 1^2,1^2 \rangle\}$, denote conservative functions of type T.

Conservativity constrains the class of possible denotations in each type quite differently, determining a fundamental distinction among the types of comparative quantifiers, not suggested by views on which they are syntactic shorthands for expressions of type $\langle 1^2, 1^2 \rangle$. This can be shown by counting how many functions there are in each type relative to the number of

individuals in the model. A comparison of the figures for each type is given below, including for completeness figures for $\langle 1,1 \rangle$. Proofs of these results are given in Appendix A3. For n individuals in the domain of discourse, we have that:

(18)

TYPE T	$\langle 1,1 \rangle$	$\langle 1,1^2 \rangle$	$\langle 1^2,1 \rangle$	$\langle 1^2,1^2 \rangle$
$ F_T $	2^{4^n}	2^{8^n}	2^{8^n}	2^{16^n}
$ CONS_T $	2^{3^n}	2^{5^n}	2^{7^n}	2^{9^n}

The notation $|F_T|$ indicates the cardinality of the set of all functions in type T (including non-conservative ones). The figures on the bottom row are provided to show first that the sets of conservative functions in the three types are very proper subsets of the sets of functions in these types. Note now that although there are as many functions in type $\langle 1,1^2 \rangle$ as in type $\langle 1^2,1 \rangle$ – since these are all functions from triples of elements of P into 2 – conservativity determines two very different denotation sets, each quite different from the one for $\langle 1^2,1^2 \rangle$ functions.

2. Closure under the boolean operations

As observed in Part 1 (cf. group F), complex comparative determiners built from lexically simpler ones by means of the boolean operations of conjunction, disjunction, and negation occur in all three syntactic environments, though only negation is productive. We observe:

Generalization 2: For T in $\{\langle 1,1 \rangle\} \cup \{\langle 1^2,1 \rangle, \langle 1,1^2 \rangle, \langle 1^2,1^2 \rangle\}$, the set of conservative functions of type T is a complete and atomic Boolean Algebra (hence closed under the boolean operations).

Thus the sets $CONS_T$ for different types T are rich enough to provide denotations for whatever boolean compounds of comparatives of type T can be formed. Moreover there are many comparatives such as **exactly ten more-than** whose denotations are boolean functions of denotations of lexically simpler determiners even though this is not reflected in their syntactic make-up: **exactly 10 more than** in fact means the same as **at least 10 more than and no more than 10 more than**.

Definitions of $CONS_T$ for each type T as a (complete, atomic) boolean algebra are given in Appendix A1.

3. Intersectivity

Intersective determiners depend for their truth value only on the intersections of their head and tail properties. We can ask if the denotation of **more-than** in, say, type $\langle 1,1^2 \rangle$ is intersective. It is, since to know the truth of "more students read than write" we only need to compare the individuals who are students and read with those who are students and write. No more is required. In particular we don't need to know about students who don't read or don't

write. Similarly, to know the truth of type $\langle 1^2, 1^2 \rangle$ **more-than**, as in "more students read than teachers write" one has merely to know which students read and which teachers write: it is not necessary to know in general which individuals are students (but don't read, etc.). Similar observations can be made for the denotation of **more-than** in type $\langle 1^2, 1 \rangle$. **more-than**, hence, is intersective in all three types. Definitions of intersectivity are given in Appendix A1. To illustrate one case here, a function $f : P \rightarrow [P \times P \rightarrow 2]$ is intersective iff for all p, q_1, q_2, q_1', q_2' in P , if $(p \wedge q_1) = (p' \wedge q_1')$ and $(p \wedge q_2) = (p' \wedge q_2')$ then $f(p)(q_1, q_2) = f(p')(q_1', q_2')$

Consider now the class of comparative expressions that occur in the syntactic environment $(N', (VP, VP))$, and denote $\langle 1, 1^2 \rangle$ functions. As noted in Part 1, with qualifications for group D, all of the comparatives expressions we have been able to collect occur in this environment. Empirical verification shows that all the functions denoted by comparatives in this type are intersective. The classic comparatives of quantity (i.e., **more-than**, **fewer-than**, **as many-as**) are, as well as numerically modified ones. **the same-as** and **different-than** are also intersective. Insofar as they are definable in extensional terms, which is the characterization we limit ourselves to here, complex comparative determiners formed by modification with AP's and PP's are intersective⁶. And $\langle 1, 1^2 \rangle$ comparatives that are boolean combinations of other comparatives are also intersective, since intersectivity is preserved under the boolean operations (cf. Appendix A1).

Even group D, proportionality comparatives, are intersective in $(N', (VP, VP))$. The truth of "a greater percentage of students came early than left late" is determined by $|student \wedge came early|$ and $|student \wedge left late|$, and thus another three properties with the same intersections will give the same results. Thus we have:

Generalization 3: Comparative expressions of type $\langle 1, 1^2 \rangle$ all denote intersective functions in that type.

Generalization 3 does not hold for the other types of comparatives. Proportionality comparatives in type $\langle 1^2, 1 \rangle$ are not intersective – nor are those in type $\langle 1^2, 1^2 \rangle$. The truth of "a greater percentage of students than teachers came early" cannot be determined if we merely know which students came early and which teachers came early. We must still compare the number of elements in these two sets with, respectively, the total number of students and the total number of teachers.

The subsets of intersective functions in the new types form subalgebras of the respective CONS algebras (cf. Appendix A1)⁷. Unlike the conservative ones, the sets of intersective functions are not distinct by size in the three types, as shown by the table below (cf. Appendix A3 for proofs of these results):

(19)

TYPE T	$\langle 1, 1 \rangle$	$\langle 1, 1^2 \rangle$	$\langle 1^2, 1 \rangle$	$\langle 1^2, 1^2 \rangle$
$ INT_T $	2^{2^n}	2^{4^n}	2^{4^n}	2^{4^n}

These results follow because the asymmetry introduced by conservativity between head(s) and tail(s) of the determiners disappears when only their intersections become relevant for the truth of the function. Whatever their type, intersective determiners in $\langle 1^2, 1 \rangle$, $\langle 1, 1^2 \rangle$, or $\langle 1^2, 1^2 \rangle$ depend on exactly two sets, each the intersection of two of the arguments of the determiner function (cf. Appendix A1). Furthermore, the intersective subtypes not only have the same cardinality, but:

Generalization 4: The semantic types of intersective comparative functions are isomorphic to each other.

The observation that the determiners in the new types are intersective predicts their ability to occur in Existential-There contexts:

- (20)a. There are more students in the garden than in the cellar $\langle 1, 1^2 \rangle$
 b. There are different professors on John's committee than on Mary's
- (21)a. There are more students than teachers in the yard $\langle 1^2, 1 \rangle$
 b. There is the same number of students as teachers in the yard
- (22)a. There are more students in the garden than cats on the roof $\langle 1^2, 1^2 \rangle$
 b. There are no fewer cats on the mat than rats in the attic

We can thus generalize a known result for $\langle 1, 1 \rangle$ determiners in English:

Generalized Existential-There Context Claim: Intersective determiners in all types occur naturally in Existential-There contexts⁸

4. Cardinality

Most intersective functions denoted by comparative expressions are cardinal, i.e., they depend only on the cardinality of the intersection of their noun and (corresponding) VP properties. The definitions of cardinal functions in the new types are given in Appendix A1. They are quite parallel to the definitions of intersective functions.

The syntactically simplest comparatives in the types (the classic comparatives *more-than*, *fewer-than*, *as many-as*; group A), are cardinal, as well as the semantically equivalent group B expressions such as *a greater/lesser/equal number of*, and those obtained from the former by means of numerical modification (group C). Adjectivally or PP modified determiners in the new types are not cardinal since their truth conditions don't depend just on the cardinality of the intersection of their arguments: they are affected by permutations of the individuals in the domain, and cardinal determiners are all permutation-invariant.

In each type T, $CARD_T$ is a (complete and atomic) subalgebra of INT_T . The classification of comparatives in each type T as belonging to successively more restricted subtypes, i.e., $CONS_T > INT_T > CARD_T$, correlates with their distribution in the syntactic environments listed in (1). The generalization is that the more restricted the subtype they denote in, the more freely the comparative determiners are represented in the types. Cardinal determiners have the widest distribution: if they occur in one type, they tend to occur in the other two syntactic types,

and combine most productively with modifiers (numerical, adjectival, etc.) to form complex comparative determiners. They are also most productive in forming complex determiners by boolean combination with other determiners in the same type. Classic comparatives, e.g., enter as components of most of the comparative expressions listed in Part 1. I will thus refer to **more-than**, **fewer-than**, and **as many-as** and their correlates in group B as the (syntactically) most basic comparatives.

Determiners that are intersective, but not also cardinal, do not occur freely in all of the three types, since, as noted, the **same-as** and **different-than** are generally ungrammatical in type $\langle 1^2, 1 \rangle$ as well as in type $\langle 1^2, 1^2 \rangle$. Adjectivally or PP-modified comparative determiners also do not seem to occur as such in type $\langle 1^2, 1^2 \rangle$: in this syntactic environment the distributive reading is absent, which suggests that aside from the marginality of these expressions in $((N', VP), (N', VP))$, the modifier should not be considered as part of the comparative determiner in this environment, but rather as part of the N' denotation. Thus:

Generalization 5: The comparative expressions which most freely occur in all three syntactic environments denote in the semantically most restricted class, that of cardinal functions.

Given an interpretation for the most basic comparatives, the interpretation of complex comparatives in the three types can generally be given compositionally. Keenan-Moss (1984) offer an illustration for $\langle 1^2, 1 \rangle$ comparatives: to provide the meaning of a complex quantity comparative such as e.g., **more French-than** in "more French students than teachers write" we can extend the modifier function denoted by **French** into an n -place function, whose values are defined coordinatewise on the argument tuple (so, if the function f and properties s, t are the denotations of **French**, **student**, and **teacher**, $f(s, t) = \langle f(s), f(t) \rangle$). Functional composition of the denotations of **French** and of **more-than** gives the desired result, i.e., *more-than* $\cdot f(s, t)$ (r) = *more-than*($f(s), f(t)$)(r). Similar procedures can be used in the other cases of complex comparatives listed in Part 1.

Just like intersective functions, the sets of cardinal functions have the same number of elements in the three types. Namely, we have:

(23)

TYPE T	$\langle 1, 1 \rangle$	$\langle 1, 1^2 \rangle$	$\langle 1^2, 1 \rangle$	$\langle 1^2, 1^2 \rangle$
$CARD_T$	$2^{(n+1)}$	$2^{(n+1)^2}$	$2^{(n+1)^2}$	$2^{(n+1)^2}$

More importantly, as observed for the class of intersective functions, the subsets of cardinal functions in the three semantic types are isomorphic (cf. Appendix A4):

Generalization 6: The semantic types of cardinal functions are isomorphic: i.e., for T, T' in $\{\langle 1^2, 1 \rangle, \langle 1, 1^2 \rangle, \langle 1^2, 1^2 \rangle\}$, $CARD_T \cong CARD_{T'}$

h and j below illustrate these Type correspondences:

h: $CARD_{\langle 1,1^2 \rangle} \rightarrow CARD_{\langle 1^2,1 \rangle}$ defined as follows: $(\text{hf})(q_1, q_2)(p) = f(p)(q_1, q_2)$;

(e.g., more students are vegetarians than are alcoholics $\leftrightarrow$ more vegetarians than alcoholics are students)

j: $CARD_{\langle 1,1^2 \rangle} \rightarrow CARD_{\langle 1^2,1^2 \rangle}$ defined as follows: $(\text{jf})(p_1, p_2)(q_1, q_2) = f(1)(p_1 \wedge q_1, p_2 \wedge q_2)$;

(e.g., more individuals are students and vegetarians than teachers and alcoholics $\leftrightarrow$ more students are vegetarians than teachers are alcoholics)

The above figures regarding the number of distinct functions in each (sub)type reveal a marked difference in magnitude between the denotation sets for the most basic (i.e., cardinal) and the other comparatives. While the size of the subtypes of cardinal functions is exponential in the size of the model (i.e., $= 2^{(n+1)^2}$, for n the size of the model), the size of the denotation sets for the other determiners is hyperexponential in the size of the model (e.g., equals 2^{4^n} , if we consider the INT subtypes, or greater, if we consider the whole $CONST$).

This difference in denotational complexity has been observed by Keenan's (1987a) for other semantic types. Keenan observes that 'large' semantic categories, i.e., categories which are hyperexponential in the size of the universe, have lexical members that are semantically restricted to denote in a 'small' subcategory, i.e., one which is at most exponential in the size of the model. The syntactically basic members of large categories always denote in the small subcategory.

Generalization 7: Comparatives in all types are not lexically free (Keenan (1987a)), i.e., the syntactically simplest comparatives in each type denote in a small subset of the set in which comparatives of that type in general may denote.

Finally, we can speculate that cardinal comparatives occur in all the syntactic types because they provide the building blocks out of which (semantically) complex determiners can be manufactured. The types differ with regard to the productivity of these operations.

Part 3: Identity of meaning across types

The isomorphism of the subtypes of cardinal (and intersective) functions offers an explanation for an obvious question raised by quantity comparatives: their identity of meaning in the distinct syntactic environments shown in (1). The answer we give is a semantic one: since the denotations of cardinal and intersective comparatives in the three types are, to within isomorphism, the same, there is only one meaning to be learned for these comparatives. Which isomorphic variant appears in a particular linguistic context depends on the syntactic configuration the determiner occurs in. (On the other hand, comparatives that denote outside of the isomorphic subtypes, namely proportionality comparatives, won't necessarily have the same denotations in the three syntactic environments, a fact that is reflected in their being interpreted as intersective in type $\langle 1,1^2 \rangle$).

We illustrate with **more-than**. The examples of isomorphic mappings given above not only show that the subclasses of intersective and cardinal functions have the same structure, but also that these isomorphisms map the denotation of each determiner in one type into the denotation for the same determiner in another type. The isomorphisms shown above, in other words, have the property of "preserving lexical denotations": they map the denotation of **more-than** in type $\langle 1, 1^2 \rangle$ into the denotation of **more-than** in type $\langle 1^2, 1 \rangle$, and similarly for the other lexical elements in the types. By the truth conditions of **more-than** in type $\langle 1, 1^2 \rangle$, "more students are vegetarians than are alcoholics" is true iff the cardinality of (*student* $\wedge$ *vegetarian*) is greater than the cardinality of (*student* $\wedge$ *alcoholic*); but if we look at the image in type $\langle 1^2, 1 \rangle$ of this function under e.g. isomorphism *h* above, i.e., "more vegetarians than alcoholics are students", we find that its truth conditions are exactly the same, and hence that the isomorphism has preserved the denotation. In this sense we can say that the speaker really only has to learn **more-than** only once. If it turned out that the only isomorphisms between the subtypes of, say, $\langle 1^2, 1 \rangle$ intersective functions into $\langle 1^2, 1^2 \rangle$ functions mapped **more-than** into **three fewer-than** in the respective subtypes, the claim about learnability would be greatly weakened.

Pursuing this line of reasoning: can we define the "common meaning" of cardinal comparatives (and of intersective comparatives that occur in all the types), i.e., the one function that the speaker learns? The answer to this question is yes. Observe that each use of **more-than** merely involves comparing the cardinalities of two properties. It is just how the properties are chosen which varies with the syntactic context. The common meaning of **more-than** can thus be given by $\phi_{\text{more-than}}$, defined as follows⁹:

$$(24) \quad \phi_{\text{more-than}}(r)(s) = \text{True iff } |r| > |s|$$

The denotation of **more-than** in the three types is determined directly from $\phi_{\text{more-than}}$:

$$\begin{aligned} \text{more-than}_{\langle 1^2, 1 \rangle} (p_1, p_2)(q) &= \phi_{\text{more-than}}(p_1 \wedge q)(p_2 \wedge q) \\ \text{more-than}_{\langle 1, 1^2 \rangle} (p)(q_1, q_2) &= \phi_{\text{more-than}}(p \wedge q_1)(p \wedge q_2) \\ \text{more-than}_{\langle 1^2, 1^2 \rangle} (p_1, p_2)(q_1, q_2) &= \phi_{\text{more-than}}(p_1 \wedge q_1)(p_2 \wedge q_2) \end{aligned}$$

$\phi_{\text{more-than}}$ gives the correct truth definitions for **more-than** in each type, once the syntactic context in which **more-than** occurs in is taken into account in the form of an appropriate mapping. The ϕ_f 's, for *f* the denotation of a cardinal (or intersective) comparative determiner, are not generally conservative functions – therefore are not available as denotations of English determiners, but, as indicated above, determine conservative functions depending on the syntactic context the comparative determiner occurs in.

The relationship of $\phi_{\text{more-than}}$ to $\text{more-than}_{\langle 1^2, 1 \rangle}$, $\text{more-than}_{\langle 1, 1^2 \rangle}$, and $\text{more-than}_{\langle 1^2, 1^2 \rangle}$ characterizes **more-than** as a polymorphic expression, not unlike **and**, which corresponds to distinct functions in the different types it denotes in (e.g., as a determiner conjunction, or an intransitive verb conjunction, or a propositional conjunction, etc.), yet can be

characterized more abstractly, as the boolean operator *meet* (i.e. the greatest lower bound operator in a boolean algebra).

Unlike syntactic accounts of comparatives which justify the identity of meaning found in (most) comparative determiners across their syntactic types by deriving them transformationally from one another, we have suggested that these identities follow from certain semantic properties of their denotations. We have not chosen one type as the most basic, but traced their semantic similarities to an abstract meaning which is not the denotation of any English determiner.

Appendix

A1 Semantic types for comparatives det_s

- Df1.** Conservativity for type $\langle 1^2, 1 \rangle$ functions: A function $f : PxP \rightarrow [P \rightarrow 2]$ is conservative iff $f(p_1, p_2)(q)$ depends at most on which individuals have p_1 , which p_2 , and which have $p_1 \wedge q$ and which $p_2 \wedge q$, i.e., for all p_1, p_2, q, q' in P :
- if $(p_1 \wedge q) = (p_1 \wedge q')$, and $(p_2 \wedge q) = (p_2 \wedge q')$ then $f(p_1, p_2)(q) = f(p_1, p_2)(q')$.
- Df2.** Conservativity for type $\langle 1, 1^2 \rangle$ functions: A function $f : P \rightarrow [PxP \rightarrow 2]$ is conservative iff for all p, q_1, q_2, q_1', q_2' in P :
- if $(p \wedge q_1) = (p \wedge q_1')$ and $(p \wedge q_2) = (p \wedge q_2')$ then $f(p)(q_1, q_2) = f(p)(q_1', q_2')$.
- Df3.** Conservativity for type $\langle 1^2, 1^2 \rangle$ functions: A function $f : PxP \rightarrow [PxP \rightarrow 2]$ is conservative iff for all $p_1, p_2, q_1, q_2, q_1', q_2'$ in P :
- if $(p_1 \wedge q_1) = (p_1 \wedge q_1')$ and $(p_2 \wedge q_2) = (p_2 \wedge q_2')$ then $f(p_1, p_2)(q_1, q_2) = f(p_1, p_2)(q_1', q_2')$.
- Df4.** For $CONS_{\langle 1^2, 1 \rangle}$ the set of conservative functions in $[PxP \rightarrow [P \rightarrow 2]]$, let $\langle CONS_{\langle 1^2, 1 \rangle}, \leq \rangle$ be defined (pointwise) as a boolean algebra, as follows: for all f, g in $CONS_{\langle 1^2, 1 \rangle}$, all p_1, p_2, q in P , $f \leq g$ iff $f(p_1, p_2)(q) \leq g(p_1, p_2)(q)$.
- Df5.** For $CONS_{\langle 1, 1^2 \rangle}$ the set of conservative functions in $[P \rightarrow [PxP \rightarrow 2]]$, let $\langle CONS_{\langle 1, 1^2 \rangle}, \leq \rangle$ be defined (pointwise) as a boolean algebra, as follows: for all f, g in $CONS_{\langle 1, 1^2 \rangle}$, all p, q_1, q_2 in P , $f \leq g$ iff $f(p)(q_1, q_2) \leq g(p)(q_1, q_2)$.
- Df6.** For $CONS_{\langle 1^2, 1^2 \rangle}$ the set of conservative functions in $[PxP \rightarrow [PxP \rightarrow 2]]$, let $\langle CONS_{\langle 1^2, 1^2 \rangle}, \leq \rangle$ be defined (pointwise) as a boolean algebra, as follows: for all f, g in $CONS_{\langle 1^2, 1^2 \rangle}$, all p_1, p_2, q_1, q_2 in P , $f \leq g$ iff $f(p_1, p_2)(q_1, q_2) \leq g(p_1, p_2)(q_1, q_2)$.
- Th1.** For all T in $\{\langle 1, 1^2 \rangle, \langle 1^2, 1 \rangle, \langle 1^2, 1^2 \rangle\}$ $CONS_T$ is complete and atomic.
- Df7.** Intersectivity for type $\langle 1^2, 1 \rangle$ functions: A function $f : PxP \rightarrow [P \rightarrow 2]$ is intersective iff for all p_1, p_2, q, q' in P ,

if $(p_1 \wedge q) = (p_1' \wedge q')$ and $(p_2 \wedge q) = (p_2' \wedge q')$ then $f(p_1, p_2)(q) = f(p_1', p_2')(q')$

Df8. **Intersectivity for type $\langle 1, 1^2 \rangle$ functions:** A function $f : P \rightarrow [PxP \rightarrow 2]$ is intersective iff for all p, q_1, q_2, q_1', q_2' in P ,

if $(p \wedge q_1) = (p' \wedge q_1')$ and $(p \wedge q_2) = (p' \wedge q_2')$ then $f(p)(q_1, q_2) = f(p')(q_1', q_2')$

Df9. **Intersectivity for type $\langle 1^2, 1^2 \rangle$ functions:** A function $f : PxP \rightarrow [PxP \rightarrow 2]$ is intersective iff for all $p_1, p_2, q_1, q_2, q_1', q_2' \in P$,

if $(p_1 \wedge q_1) = (p_1' \wedge q_1')$ and $(p_2 \wedge q_2) = (p_2' \wedge q_2')$ then $f(p_1, p_2)(q_1, q_2) = f(p_1', p_2')(q_1', q_2')$

The following theorems characterize intersective determiners in the new types, where E is the unit element of P (i.e., E corresponds to the set of all individuals):

Th2. A function $f : PxP \rightarrow [P \rightarrow 2]$ is intersective iff for all p_1, p_2, q in P , $f(p_1, p_2)(q) = f(p_1 \wedge q, p_2 \wedge q)(E)$

Th3. A function $f : P \rightarrow [PxP \rightarrow 2]$ is intersective iff for all p, q_1, q_2 in P , $f(p)(q_1, q_2) = f(E)(p \wedge q_1, p \wedge q_2)$

Th4. A function $f : PxP \rightarrow [PxP \rightarrow 2]$ is intersective iff for all p_1, p_2, q_1, q_2 in P , $f(p_1, p_2)(q_1, q_2) = f(E, p_1 \wedge q_1)(E, p_2 \wedge q_2)$

The following theorems characterize intersective and cardinal functions in the semantic types as subalgebras of the conservative functions in each type:

Th5. For all T in $\{\langle 1^2, 1 \rangle, \langle 1, 1^2 \rangle, \langle 1^2, 1^2 \rangle\}$, INT_T is a (complete, atomic) subalgebra of $CONS_T$

Th6. For all T in $\{\langle 1^2, 1 \rangle, \langle 1, 1^2 \rangle, \langle 1^2, 1^2 \rangle\}$, $\langle CARD_T, \leq \rangle$ is a (complete, atomic) subalgebra of INT_T .

A2 Atoms

The following theorems tell us what are the atoms in the (sub)types. These results will be used to determine the size of the (sub)types.

Th7. $Atm(CONS_{\langle 1, 1^2 \rangle}) = \{g_{\langle p, q, r \rangle} \mid \text{for all } s, t, u \text{ in } P, g_{\langle p, q, r \rangle}(s)(t, u) = 1 \text{ iff } s = p \text{ and } s \wedge t = q, s \wedge u = r, \text{ for } q, r \leq p\}.$

Th8. $Atm(INT_{\langle 1, 1^2 \rangle}) = \{g_{\langle q, r \rangle} \mid \text{for all } s, t, u \text{ in } P, g_{\langle q, r \rangle}(s)(t, u) = 1 \text{ iff } s \wedge t = q \text{ and } s \wedge u = r\}$

Th9. $Atm(CARD_{\langle 1, 1^2 \rangle}) = \{g_{\langle n, n' \rangle} \mid \text{for all } s, t, u \text{ in } P, g_{\langle n, n' \rangle}(s)(t, u) = 1 \text{ iff } |s \wedge t| = n \text{ and } |s \wedge u| = n', \text{ for some } n, n' \text{ such that } 0 \leq n, n' \leq |1_P|\}.$

Th10. $Atm(CONS_{\langle 1^2, 1 \rangle}) = \{g_{\langle p, v, q \rangle} \mid \text{for all } s, z, t \text{ in } P, g_{\langle p, v, q \rangle}(s, z)(t) = 1 \text{ iff } s = p, z = v \text{ and } s \wedge t = q, \text{ for } q \leq p, r \leq v\}.$

- Th11.** $\text{Atm}(\text{INT}_{\langle 1^2, 1 \rangle}) = \{g_{\langle q, r \rangle} \mid \text{for all } s, z, t \text{ in } P, g_{\langle q, r \rangle}(s, z)(t) = 1 \text{ iff } s \wedge t = q \text{ and } z \wedge t = r\}$
- Th12.** $\text{Atm}(\text{CARD}_{\langle 1^2, 1 \rangle}) = \{g_{\langle n, m \rangle} \mid \text{for all } s, z, t \text{ in } P, g_{\langle n, m \rangle}(s, z)(t) = 1 \text{ iff } |s \wedge t| = n \text{ and } |z \wedge t| = m, \text{ for some } n, m \text{ such that } 0 \leq n, m \leq |1_P|\}$.
- Th13.** $\text{Atm}(\text{CONS}_{\langle 1^2, 1^2 \rangle}) = \{g_{\langle p, v, q, r \rangle} \mid \text{for all } s, z, t, u \text{ in } P, g_{\langle p, v, q, r \rangle}(s, z)(t, u) = 1 \text{ iff } s = p, z = v \text{ and } s \wedge t = q, z \wedge u = r, \text{ for } q \leq p, r \leq v\}$.
- Th14.** $\text{Atm}(\text{INT}_{\langle 1^2, 1^2 \rangle}) = \{g_{\langle q, r \rangle} \mid \text{for all } s, z, t, u \text{ in } P, g_{\langle q, r \rangle}(s, z)(t, u) = 1 \text{ iff } s \wedge t = q \text{ and } z \wedge u = r\}$
- Th15.** $\text{Atm}(\text{CARD}_{\langle 1^2, 1^2 \rangle}) = \{g_{\langle n, m \rangle} \mid \text{for all } s, z, t, u \text{ in } P, g_{\langle n, m \rangle}(s, z)(t, u) = 1 \text{ iff } |s \wedge t| = n \text{ and } |z \wedge u| = m, \text{ for some } n, m \text{ such that } 0 \leq n, m \leq |1_P|\}$.

A3 Counts

These theorems determine the size of the (sub)types. Since each of the (sub)types below is a complete and atomic boolean algebra it is isomorphic to the power set of the atoms. So to count the type we need only count the atoms.

Th16. $|F_{\langle 1, 1^2 \rangle}| = |F_{\langle 1^2, 1 \rangle}| = 2^{8^n}$

PROOF: $F_{\langle 1, 1^2 \rangle} = [P \rightarrow [P \times P \rightarrow 2]] \approx [P^3 \rightarrow 2]$. There are, then, $(2^{2^n} \cdot 2^n) \cdot 2^n = 2^{8^n}$ distinct such functions. The same argument holds for $F_{\langle 1^2, 1 \rangle}$.

Th17. $|\text{CONS}_{\langle 1, 1^2 \rangle}| = 2^{5^n}$

PROOF: We show $|\text{Atm}(\text{CONS}_{\langle 1, 1^2 \rangle})| = 5^n$, for $n = |\text{Atm}(P)|$. Consideration of the atoms of $\text{CONS}_{\langle 1, 1^2 \rangle}$ shows that there is a bijection between $P \times P \times P$ and the atoms of $\text{CONS}_{\langle 1, 1^2 \rangle}$: i.e., Let $g_{\langle p, q, r \rangle}$ be an atom of $\text{CONS}_{\langle 1, 1^2 \rangle}$ (see Th7). Then $|\text{Atm}(\text{CONS}_{\langle 1, 1^2 \rangle})| = |\{\langle p, q, r \rangle \mid q, r \leq p\}|$. Now define $h: \{\langle p, q, r \rangle \mid q, r \leq p\} \rightarrow [\text{Atm}(P) \rightarrow 5]$ as follows: for p, q, r in P , $q, r \leq p$, and a an atom of P ,

$$h(\langle p, q, r \rangle)(a) = \begin{array}{ll} 1 & \text{if } a \leq \neg p \\ 2 & \text{if } a \leq p \wedge q \wedge r \\ 3 & \text{if } a \leq p \wedge q \wedge \neg r \\ 4 & \text{if } a \leq p \wedge \neg q \wedge r \\ 5 & \text{if } a \leq p \wedge \neg q \wedge \neg r \end{array}$$

h is well defined since the conditions are exclusive and exhaustive. We show that h is a bijection, which will prove that $|\text{Atm}(\text{CONS}_{\langle 1, 1^2 \rangle})| = 5^{\text{Atm}(P)}$.

(A) h is 1-to-1. Assume $\langle p, q, r \rangle \neq \langle p', q', r' \rangle$; show $h(\langle p, q, r \rangle) \neq h(\langle p', q', r' \rangle)$. Case 1: $p' \neq p$, i.e., $p \not\leq p'$ or $p' \not\leq p$. Without loss of generality, assume the former. Then $(p \wedge \neg p') \neq 0$, so there is an atom b such that $b \leq (p \wedge \neg p')$, i.e., $b \leq p$ and $b \not\leq p'$. Since $b \leq p$, it is not the case that $b \leq \neg p$ (no atom is $\leq$ a property and its complement), hence $h(\langle p, q, r \rangle)(b) \neq 1$. And since b

$\leq -p'$, $h(\langle p', q', r' \rangle)(b) = 1$. Case 2: $p=p'$, but $q \neq q'$. As before, let q not $\leq q'$. Then there is an atom c , $c \leq (q \wedge -q')$. Again, since $c \leq q$, $h(\langle p, q, r \rangle)(c)$ is either 2 or 3; and since $c \leq -q'$, $h(\langle p', q', r' \rangle)(c)$ is either 4 or 5. Case 3: $r \neq r'$ is analogous.

(B) h is onto. Let f be a function with domain $\text{Atm}(P)$ and range 5. We need to show that there is a triple $\langle p, q, r \rangle$ such that $h(\langle p, q, r \rangle) = f$. We define a triple $\langle p_f, q_f, r_f \rangle$ as follows:

$$p_f = \bigvee \{a \text{ in } \text{Atm}(P) \mid f(a) \text{ in } \{2, 3, 4, 5\}\}$$

$$q_f = \bigvee \{a \text{ in } \text{Atm}(P) \mid f(a) \text{ in } \{2, 3\}\}$$

$$r_f = \bigvee \{a \text{ in } \text{Atm}(P) \mid f(a) \text{ in } \{2, 5\}\}$$

Obviously, $q_f, r_f \leq p_f$. We show $f = h(\langle p_f, q_f, r_f \rangle)$:

Case 1. $h(\langle p_f, q_f, r_f \rangle)(a) = 1$ iff $a \leq -p_f$, iff $f(a)$ not in $\{2, 3, 4, 5\}$, i.e., iff $f(a) = 1$;

Case 2. $h(\langle p_f, q_f, r_f \rangle)(a) = 2$ if $a \leq -p_f$ and $a \leq q_f$ and $a \leq r_f$, iff $f(a)$ is in $\{2, 3, 4, 5\}$ and $f(a)$ is in $\{2, 3\}$ and $f(a)$ is in $\{2, 4\}$; i.e., iff $f(a) = 2$;

Case 3. $h(\langle p_f, q_f, r_f \rangle)(a) = 3$ if $a \leq p_f$ and $a \leq q_f$ and $a \leq -r_f$, iff $f(a)$ is in $\{2, 3, 4, 5\}$ and $f(a)$ is in $\{2, 3\}$ and $f(a)$ is not in $\{2, 4\}$; i.e., iff $f(a) = 3$;

Case 4. $h(\langle p_f, q_f, r_f \rangle)(a) = 4$ if $a \leq p_f$ and $a \leq -q_f$ and $a \leq r_f$, iff $f(a)$ is in $\{2, 3, 4, 5\}$ and $f(a)$ is not in $\{2, 3\}$ and $f(a)$ is in $\{2, 4\}$; i.e., iff $f(a) = 4$;

Case 5. $h(\langle p_f, q_f, r_f \rangle)(a) = 5$ if $a \leq p_f$ and $a \leq -q_f$ and $a \leq -r_f$, iff $f(a)$ is in $\{2, 3, 4, 5\}$ and $f(a)$ is not in $\{2, 3\}$ and $f(a)$ is not in $\{2, 4\}$; i.e., iff $f(a) = 5$. QED

Th18. $|\text{INT}_{\langle 1, 1^2 \rangle}| = 2^{4^n}$

PROOF: We show $|\text{Atm}(\text{INT}_{\langle 1, 1^2 \rangle})| = 4^n$, $n = |\text{Atm}(P)|$. Consideration of $\text{Atm}(\text{INT}_{\langle 1, 1^2 \rangle})$ shows that each atom corresponds univocally to a pair of properties, i.e., that a map $h : \text{PxP} \rightarrow \text{Atm}(\text{INT}_{\langle 1, 1^2 \rangle})$ given by $h(q, r) = g_{q, r}$ is surjective. h is also injective, since if $\langle q', r' \rangle$ then $g_{q, r} \neq g_{q', r'}$. Hence h is a bijection, and $|\text{PxP}| = |\text{Atm}(\text{INT}_{\langle 1, 1^2 \rangle})| = 2^n \cdot 2^n = 4^n$. QED

Th19. $|\text{CARD}_{\langle 1, 1^2 \rangle}| = 2^{(n+1)^2}$

PROOF: $|\text{Atm}(\text{CARD}_{\langle 1, 1^2 \rangle})| = |\{ \langle n, n' \rangle \mid 0 \leq n, n' \leq |E| \}| = (n+1)^2$. Since for all such $\langle n, n' \rangle$ the functions $g_{n, n'}(s)(t, u) = 1$ iff $|s \wedge t| = n$ and $|s \wedge u| = n'$ are the atoms of $\text{CARD}_{\langle 1, 1^2 \rangle}$, distinct pairs give rise to distinct functions. QED

Proofs that $|\text{INT}_{\langle 1^2, 1 \rangle}|$ and $|\text{INT}_{\langle 1^2, 1^2 \rangle}| = 2^{4^n}$, and that $|\text{CARD}_{\langle 1^2, 1 \rangle}|$ and $|\text{CARD}_{\langle 1^2, 1^2 \rangle}| = 2^{(n+1)^2}$ are similar to the proofs for type $\langle 1, 1^2 \rangle$.

Th20. $|\text{CONS}_{\langle 1^2, 1 \rangle}| = 2^{7^n}$

PROOF: Show $|Atm(CONS_{\langle 1^2, 1 \rangle})| = 7^n$, for $n = |Atm(P)|$. The proof is similar to the one for type $\langle 1, 1^2 \rangle$ given above.

Th21. $|F_{\langle 1^2, 1^2 \rangle}| = 2^{16^n}$

PROOF: $F_{\langle 1^2, 1^2 \rangle} = [[(PxP)x(PxP)] \rightarrow 2] : [P^4 \rightarrow 2]$, and has cardinality $(2^{2^n} \cdot 2^n)^{2^n} = 2^{16^n}$.

Th22. $|CONS_{\langle 1^2, 1^2 \rangle}| = 2^{3^n} \cdot 3^n = 2^{9^n}$

PROOF: By the definition of $CONS_{\langle 1^2, 1^2 \rangle}$, each of these functions is defined by a pair of type $\langle 1, 1 \rangle$ conservative functions. We know there are 2^{3^n} distinct functions in $CONS_{\langle 1, 1 \rangle}$; therefore, there are $(2^{3^n})^{3^n} = 2^{9^n}$ distinct functions in $CONS_{\langle 1^2, 1^2 \rangle}$.

A4 Isomorphisms

The theorems in this section establish the existence of isomorphisms between the subtypes of INT and CARD functions in the new types. A complete proof is given only for the first one.

Th23. The subalgebra of INTERSECTIVE dets of type $\langle 1, 1^2 \rangle$ is isomorphic to the subalgebra of INTERSECTIVE dets of type $\langle 1^2, 1 \rangle$.

PROOF: Define $h : \langle INT_{\langle 1, 1^2 \rangle}, \leq \rangle \rightarrow \langle INT_{\langle 1^2, 1 \rangle}, \leq \rangle$ as follows: $(hf)(q_1, q_2)(p) = f(p)(q_1, q_2)$. Claim: h is an isomorphism.

(A) h is 1-to-1. Assume for f, f' in $INT_{\langle 1, 1^2 \rangle}$, $f \neq f'$; let p, q_1, q_2 be such that $f(p)(q_1, q_2) \neq f'(p)(q_1, q_2)$. $(hf)(q_1, q_2)(p) = f(p)(q_1, q_2) \neq f'(p)(q_1, q_2) = (hf')(q_1, q_2)(p)$. Hence $hf \neq hf'$.

(B) h is onto. For g arbitrary in $INT_{\langle 1^2, 1 \rangle}$, show there is g' in $INT_{\langle 1, 1^2 \rangle}$ such that $h(g') = g$. For g in $INT_{\langle 1^2, 1 \rangle}$, define $g' : P \rightarrow [PxP \rightarrow 2]$ by $g'(p)(q_1, q_2) = g(q_1, q_2)(p)$. g' is in $INT_{\langle 1, 1^2 \rangle}$, for if $|p \wedge q_1| = |p' \wedge q_1'|$ and $|p \wedge q_2| = |p' \wedge q_2'|$, $g(q_1, q_2)(p) = g'(p)(q_1, q_2) = g(q_1', q_2')(p) = g'(p')(q_1', q_2')$. Show $h(g') = g$. $(hg')(q_1, q_2)(p) = g'(p)(q_1, q_2) = g(q_1, q_2)(p)$.

(C) h preserves the boolean operations. Show $h(f \wedge f')(q_1 \wedge q_2)(p) = (hf \wedge hf')(q_1 \wedge q_2)(p)$. $h(f \wedge f')(q_1 \wedge q_2)(p) = (f \wedge f')(p)(q_1 \wedge q_2) = f(p)(q_1, q_2) \wedge f'(p)(q_1, q_2) = (hf)(q_1, q_2)(p) \wedge (hf')(q_1, q_2)(p) = (hf \wedge hf')(q_1 \wedge q_2)(p)$. The first and third equalities are by definition of h ; the second and fourth by pointwise definition in the subalgebras $\langle INT_{\langle 1^2, 1 \rangle}, \leq \rangle$, $\langle INT_{\langle 1, 1^2 \rangle}, \leq \rangle$ respectively. Similarly, we show $h(-f)(q_1 \wedge q_2)(p) = -(hf)(q_1 \wedge q_2)(p)$. $h(-f)(q_1 \wedge q_2)(p) = (-f)(p)(q_1 \wedge q_2) = -f(p)(q_1, q_2) = -(hf)(q_1, q_2)(p) = -(hf)(q_1 \wedge q_2)(p)$. QED

Th24. The subalgebra of INTERSECTIVE dets of type $\langle 1, 1^2 \rangle$ is isomorphic to the subalgebra of INTERSECTIVE dets of type $\langle 1^2, 1 \rangle$.

PROOF: Define $h : \langle \text{INT} \langle 1, 1^2 \rangle, \leq \rangle \rightarrow \langle \text{INT} \langle 1^2, 1^2 \rangle, \leq \rangle$ as follows: $(hf)(p, p)(q_1, q_2) = f(p)(q_1, q_2)$. Then h is an isomorphism.

Th25. The subalgebra of CARDINAL dets of type $\langle 1, 1^2 \rangle$ is isomorphic to the subalgebra of CARDINAL dets of type $\langle 1^2, 1 \rangle$.

PROOF: Define $h : \langle \text{CARD} \langle 1, 1^2 \rangle, \leq \rangle \rightarrow \langle \text{CARD} \langle 1^2, 1 \rangle, \leq \rangle$ as follows: $(hf)(q_1, q_2)(p) = f(p)(q_1, q_2)$. h is an isomorphism.

Th26. The subalgebra of CARDINAL dets of type $\langle 1, 1^2 \rangle$ is isomorphic to the subalgebra of CARDINAL dets of type $\langle 1^2, 1^2 \rangle$.

PROOF: Define $h : \langle \text{CARD} \langle 1, 1^2 \rangle, \leq \rangle \rightarrow \langle \text{CARD} \langle 1^2, 1^2 \rangle, \leq \rangle$ as follows: $(hf)(p, p)(q_1, q_2) = f(p)(q_1, q_2)$. h is an isomorphism.

Footnotes

¹ I am concerned here only with quantity comparatives, although the range of constructions exemplified in (a-c) above includes comparatives of degree as well. Consider:

- (2') a. in Linguistics 567 there are more advanced students than teachers
- b. people with more nimble brains than limbs prefer math classes to football
- (3') a. more talented contestants were excluded from competing than received awards
- b. less intelligent students ended up in John's class than in Mary's
- (4') a. more sensitive authors wrote the plays than critics assigned the prizes
- b. Jane picks more faithful boyfriends than John chooses girlfriends

The comparatives in (2'-4') correspond to the types exemplified in (1a-c). The sentences in (2'-4'), to be sure, are ambiguous between a quantity reading and a degree reading: for (3'a) the readings are:

- (5') the contestants who were excluded from competing are more talented than the ones that received awards [degree]
- (5'') the number of talented contestants who were excluded from competing exceeds the number of talented contestants who received awards [quantity]

² Note that we have enriched Lindström's "flat" notation. (1a) and (1b) are understood in different ways even when their argument properties are the same. Consider:

- (1) b'. more students teach than read

(1a) and (1b') have different truth values, a fact that would be obscured if we treated both more-than's as having the unstructured type $\langle 1, 1, 1 \rangle$, as per Lindström's notation.

³ It is rarely possible to insert G expressions in $((N', N'), VP)$ contexts, and only with abstract nouns or the use of temporal modifiers. A candidate:

- (6') The same retreating fugitives as, years before, proud conquerors, passed through the city

⁴ Some possible examples of group G expressions in this context occur in object position in sentences like:

Footnotes (continued)

(7') John has the same fears as Mary has desires

(8') I have the same doubts about John as Mary seems to have certainties

⁵ Note the following theorem:

Thm: A function $f : P \rightarrow [P \rightarrow 2]$ is intersective iff for all p, q in P , $f(p)(q) = f(1)(p \wedge q)$, where 1 is the property that all individuals have.

The above theorem provides a test which is sometimes useful in determining whether a given determiner is intersective or not: intersective determiners (such as *some*) are those for which the following equivalence holds:

Some students are vegetarians $\leftrightarrow$ some individuals are both students and vegetarians

⁶ Adjectives characterizable purely in extensional terms are for most part "absolute" ones, i.e., those denoting functions f such that for p the denotation of a common noun, $f(p) = f(1) \wedge p$. So, for instance, a French lawyer is a French individual who's a lawyer.

⁷ Although there seem to be members of the new types that are not intersective, as the proportional ones, it can be noted that there are no determiners in the new types that behave like the universal quantifier, in the sense of being co-intersective (cf. Keenan-Stavi (1986): a function f is co-intersective iff it depends only on which individuals have the head and lack the corresponding tail properties).

⁸ Anna Szabolcsi (pers. comm.) points out that the noted exceptions to the generalization that all determiners in the new types are intersective, namely proportional comparatives such as "proportionally more-than", do not actually behave like the intersective ones with respect to Existential-There Contexts. Although "proportionally more-than" is fine in an existential construction such as:

(9') There are proportionally more students than teachers in the yard

It is crucial that there be a "coda" property for the existential construction to be well formed. Without a coda, the construction is not semantically acceptable:

(10') *?There are proportionally more students than teachers

In general, intersective determiners don't require codas to occur in There-Existential Contexts:

(11') There are more students than teachers

⁹ This analysis is similar to the one developed in Nerbonne (1990).

References

- Barwise, J. & Cooper, R.: 1981, Generalized Quantifiers in Natural Language, *L&P* 4(1), 159-220
- Bresnan, J.: 1973, Syntax of the Comparative Clause Construction in English, *LI* 4 (3), 275-343
- Lindström, P.: 1966, First-order Predicate Logic with Generalized Quantifiers, *Theoria* 32, 186-195.
- Keenan, E. & Faltz, L.: 1985, *Boolean Semantics for Natural Language*, Reidel, Dordrecht
- Keenan, E. & Moss, L.: 1984, Generalized Quantifiers and the Expressive Power of Natural Language, in van Benthem, J. and ter Meulen, A. (eds): *Generalized Quantifiers in Natural Language*, Foris, Dordrecht.
- Keenan, E. & Stavi, J.: 1986, A Semantic Characterization of Natural Language Determiners, in *L&P* 9, 253-326.
- Keenan, E.: 1987a, Lexical Freedom and Large Categories, in Groenendijk et al. (eds): *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*, Foris, Dordrecht.
- Keenan, E.: 1987b, Multiply-Headed Noun Phrases, *LI* 18 (3),
- Napoli, D.J.: 1983, Comparative Ellipsis: A Phrase Structure Analysis, *LI* 14 (4), 675-694.
- Nerbonne, J.: 1991, Nominal Comparatives and Generalized Quantifiers, unpublished draft
- Reinhart, T.: 1991, Elliptic Conjunctions – Non-Quantificational LF, in Kasher, A. (ed): *The Chomskyan Turn*, Blackwell
- van Benthem, J.: 1986, *Essays in Logical Semantics*, Reidel, Dordrecht.

Tense, Temporal Reference and Tense Logic

Patrick Blackburn
Universiteit van Amsterdam

Few formal semanticists look with much favour upon Priorean tense logic. For a start, natural language tenses usually have referential import — but reference to times is not possible in Priorean languages. Secondly, the focus of research on tense has shifted in recent years. The anaphoric properties of tense, and the role temporal constructions have to play in discourse, are now centre stage. Such issues seem alien to tense logic.

These shortcomings are real, nonetheless there is a great deal of value in Prior's approach. Priorean tense logic is a modal logic, and thus it insists upon an *internal* view of time, rather than the Gods-eye-view of classical languages. Intuitively this seems a promising way to approach temporal language: we live in time, and the 'internal view' is imprinted upon natural languages in many ways. Moreover, as we now know from *correspondence theory* [1][2], Prior's insistence on the internal perspective has logical consequences. To work in a modal language is to work in a very restricted fragment of classical logic, thus the mere fact that a problem can be modeled in a modal language already tells us something interesting. Finally, the fundamental logical properties of Prior's languages are well understood. For example, completeness and decidability results for various classes of temporal structures have been obtained. Given the increasing interplay between formal semantics and neighbouring disciplines such as knowledge representation, and the consequent emphasis on the role of inference, such logical insight promises to be valuable.

This paper is an attempt to build on Prior's approach. Various extensions of tense logic are introduced and shown to be adequate for dealing with the problems mentioned above, as well as a number of other phenomena. What is unusual is the direction these extensions take. Rather than introducing a variety of new temporal operators, we *sort* the atomic symbols of the language. That is, we introduce different sorts of propositional symbols, and constrain their interpretation in various ways. This sorting strategy is the heart of the paper.

We proceed as follows. After a *résumé* of Priorean tense logic, we introduce our first sorted language: *nominal tense logic*. In this extension a second sort of propositional variable — the *nominal* — is introduced. Nominals are constrained to be true at exactly one time, thus they act as a name for the time they are true at. I first show how this simple referential mechanism enables us to combine the ideas of Prior and Reichenbach to good effect, and go on to argue that nominals enable us to deal with certain anaphoric phenomenon; the key idea here is to use nominals as if they were discourse markers of temporal DRT.

With the strategy of using different sorts of propositional symbols thus established, I apply the idea to indexicals. I extend the language of nominal tense logic by adding three new propositional symbols: *yesterday*, *today*, and *tomorrow*, and interpret them using the semantic machinery of the Californian theory of reference. As we shall see, the exploitation of this machinery by means of sorted atomic symbols yields a clean account of some basic interactions between tense and temporal adverbials.

We conclude the paper by looking further afield. A number of additional extensions are noted, and the possibility of sorting in the richer setting of interval based logic is briefly

discussed.

A few historical remarks are in order. The idea of sorting intensional languages can be traced back to work done in the late 1960s by Arthur Prior [21][22] and Robert Bull [4]. This work seems to have been largely overlooked. The Sofia school of modal logic independently developed similar ideas in the 1980s, both in the context of Propositional Dynamic Logic [18][19], and ordinary modal logic [7][8]. Slightly later, my thesis [3] considered sorting in the setting of temporal logics.

1 Priorean Tense Logic

The language of (propositional) Priorean tense logic is constructed out of the following primitive symbols. First we have a countably infinite collection VAR of propositional variables, which we officially write as p , q , and r , possibly subscripted; however when discussing examples drawn from natural language we shall frequently use more mnemonic symbols such as *John run*, *Fido bark* and *The third door on the left be brown*. In addition we have some truth functionally adequate collection of Boolean connectives (in this paper we shall choose $\neg$ and $\wedge$), and the tense operators P and F .

We form the well formed formulas (or sentences) of Priorean tense logic from these symbols as follows. First we define ATOM, the set of atomic symbols of the language, to be precisely VAR, the set of propositional variables. We then stipulate that WFF, the set of well formed formulas of the language, is to be the smallest set containing ATOM that is closed under negation, conjunction, and the application of tense operators. We make free use of all the usual defined symbols. In particular, we define the other Boolean connectives $\rightarrow$, $\vee$ and $\leftrightarrow$ in the usual manner, and in addition define $G\phi$ to be $\neg F\neg\phi$, and $H\phi$ to be $\neg P\neg\phi$, for all wffs ϕ .

All this will probably be familiar to the reader, with one (perhaps puzzling) exception: why did we bother introducing the set ATOM? This is clearly redundant: we could have defined WFF directly in terms of VAR. In fact the introduction of ATOM is an anticipation of the later sorted languages we shall consider. Syntactically, sorting merely amounts to introducing different sorts of atomic symbol, and indeed in what follows the only syntactical work done is to successively enrich the definition of ATOM. That is, all the languages discussed in this paper have exactly the same formation rules, namely those stated above in terms of ATOM.

To interpret the Priorean language we make use of *frames* and *Kripke models*. A frame T is a pair $\langle T, < \rangle$ where T is a non-empty set, and $<$ is a binary relation on T . Frames are a mathematical characterisation of a simple conception of time, namely that time is some kind of flow. Thus the elements of T are often called *times* or *points*, and $<$ is usually called the *precedence relation*. In order to make this simple model more realistic we shall place some constraints on allowable precedence relations. In what follows we shall assume that $<$ is always a *strict total order*, that is, a relation that is transitive, irreflexive, and trichotomous. In short, we are assuming a linear flow of time. This assumption will suffice for the first half of the paper; thereafter we will need to be a little more demanding.

A Kripke model M is a pair $\langle T, V \rangle$ where $T (= \langle T, < \rangle)$ is a frame and V is a *valuation* on T . A valuation on T is a function from ATOM to $Pow(T)$. Valuations thus impose 'information distributions' on frames, and Kripke models can be thought of as time flows decorated with information about what happens when.

Given a Kripke model $M (= \langle T, <, V \rangle)$, a wff ϕ , and a point $t \in T$ the fundamental thing we are concerned with is whether M *satisfies* ϕ at t , or in symbolic form, whether $M \models \phi[t]$. This relation is defined as follows:

$M \models a[t]$	iff	$t \in V(a)$, for all atoms a
$M \models \neg\phi[t]$	iff	not $M \models \phi[t]$
$M \models \phi \wedge \psi[t]$	iff	$M \models \phi[t]$ and $M \models \psi[t]$
$M \models F\phi[t]$	iff	$\exists t'(t < t' \text{ and } M \models \phi[t'])$
$M \models P\phi[t]$	iff	$\exists t'(t' < t \text{ and } M \models \phi[t'])$

If $M \models \phi[t]$ we often say that ϕ is *true in M at t*. We occasionally make use of the idea of *logical equivalence*. This is defined as follows. Two wffs ϕ and ψ are logically equivalent iff for all models M and all points t in M :

$$M \models \phi[t] \text{ iff } M \models \psi[t].$$

This semantic machinery is probably familiar to the reader, but (once again) it's worth noting that it is all defined in terms of ATOM. In fact, because of this parameterisation on ATOM, most of the above machinery (for example, the satisfaction definition) will remain unchanged throughout the paper. The only changes will occur on the atomic level. As already been stated, ATOM is going to acquire a syntactic 'fine structure'. This fine structure will have a semantic correlate: we are going to place constraints on how the different pieces may be interpreted. In short, time and again through the paper we will be placing constraints on what functions are to be admissible as valuations, but once these constraints have been stated, the languages are interpreted exactly as described above.

2 Nominal Tense Logic

As Priorean tense logic contains no mechanism for referring to times, its use as a model of tense in natural language has severe limitations. Tenses aren't solely, or even primarily, quantificational in nature: they usually have referential import. For example, an utterance of "John ran" doesn't mean that John ran at some completely unspecified past time, but that he ran at some *particular* past time. The natural representation of this sentence in Priorean tense logic, $P(\text{John run})$, fails to mirror the referential nature of the English original.

Our solution to this difficulty is very simple. We make Priorean tense logic referential by means of the following extension. We add a new type of propositional symbol, called *nominals*, to our language. Nominals will typically be written as i, j, k and so on. They will be wffs, and we will be able to freely combine them with other wffs in the usual fashion to form new wffs. The point of introducing nominals lies in their interpretation: *we'll insist that in any model nominals are to be true at exactly one time*. Nominals thus 'name' the unique time they are true at and our problem is solved. For example we will now have a simple representation for "John ran" that captures its referential force: $P(i \wedge \text{John run})$ suffices, for in order for this to be true John must run at the past time picked out by i .

That's all there is to it. Let's go through it in more detail and see where it leads.

Syntactically we obtain the language of nominal tense logic NTL from the language of Priorean tense logic by adding a denumerably infinite set NOM of fresh atomic symbols to the language. The elements of NOM are denoted by $i, j, k, \dots$, possibly subscripted, and they are called nominals. These symbols are 'fresh' in the sense that $\text{NOM} \cap \text{VAR} = \emptyset$ — that is, nominals and propositional variables are distinct. We define the set of atoms ATOM of the language of NTL to be $\text{NOM} \cup \text{VAR}$, and we make the wffs of this enriched language using exactly the same formation rules as for the Priorean language: WFF is the smallest set containing ATOM that is closed under negation, conjunction, and application of tense operators. Thus the only syntactic difference is that ATOM is now $\text{VAR} \cup \text{NOM}$, instead

of simply VAR. Typical examples of wffs are the *purely nominal* wff $i \rightarrow \neg Fi$, the *purely Priorean* wff $FFp \rightarrow Fp$, and the *mixed* wff $F(i \wedge p) \wedge F(i \wedge q) \rightarrow F(i \wedge p \wedge q)$.

As with the Priorean languages we interpret NTL in Kripke models $M = \langle T, V \rangle$. As before T is a frame $\langle T, < \rangle$ and V is a function from ATOM to $Pow(T)$. However we now make a crucial demand: for all $i \in NOM$, $V(i)$ is a singleton subset of T . Functions V that don't obey this sortal restriction aren't valuations. As with the syntax, this atomic level change is the only change we make. Wffs of NTL are interpreted in Kripke models in exactly the manner familiar from Priorean tense logic.

As the representation of "John ran" as $P(i \wedge John\ run)$ shows, the utility of nominals stems from the fact that they are a referential mechanism. Consider $P(i \wedge John\ run)$ once more. Because i is true at one and only one point in any model, this representation makes a far stronger claim than $P(John\ run)$. For $P(i \wedge John\ run)$ to be true at some time of utterance t , not only must there be a time t' preceding t at which *John run* is true, but in fact the unique time picked out by i must precede t and it must be true at this particular time that *John run*.

In short, the enriched language gives us extra expressive power — and because temporal reference is such a pervasive phenomenon in natural language, this extra power is useful. For a start NTL provides a framework in which the ideas of Reichenbach and Prior can be naturally amalgamated.

Reichenbach claimed that understanding tensed expressions involved understanding the temporal relations that held between *three* special points of time: the *point of speech*, the *point of event* and the *point of reference*. The first two points have the obvious meanings: the point of speech is the utterance time, and the point of event is the time at which the event being spoken of occurred. Note that Priorean tense logic, with its internal view of time, gives a clear account of these two points: point of speech is the point in the model at which the utterance is evaluated, while point of event is the point in the model where the event being spoken of is verified.

But what of Reichenbach's novelty, *point of reference*? This concept is best introduced by example. Consider the past perfect sentence "John had run". Note the way it works. Our attention is not directed immediately to the time at which John runs, rather we are first referred to some point t' preceding the point of speech and told that at some point prior to this intermediate point John ran. Such an intermediate point is called a point of reference.

Reichenbach accounted for the variety of tenses found in natural language in terms of the different patterns of temporal precedence and coincidence these three points can exhibit. For example, in the past perfect we have that the point of event E precedes the point of reference R , which in turn precedes the point of speech S , or, to use Reichenbach's diagrammatic notation, $E - R - S$. To give a second example, Reichenbach proposed that the function of simple past tense was to locate the point of event and point of reference at the same point in the past, thus his diagram for the simple past was $E, R - S$. Now this concept of a special intermediate point picked out by tensed sentences simply isn't present in the Priorean framework. Equally clearly, once we have nominals in our language to refer to times, we can mirror this idea. Let's consider the matter systematically. Reichenbach's idea admits of thirteen possible patterns of precedence and coincidence. The following table, the first three columns of which are taken from Bernard Comrie [5, page 25], lists with examples the possibilities admitted by Reichenbach's analysis. The fourth column gives an NTL representation.

Structure	Name	English example	Representation
E-R-S	Pluperfect	I had seen	$P(i \wedge P\phi)$
E,R-S	Past	I saw	$P(i \wedge \phi)$
R-E-S	Future-in-the-past	I would see	$P(i \wedge F\phi)$
R-S,E	Future-in-the-past	I would see	$P(i \wedge F\phi)$
R-S-E	Future-in-the-past	I would see	$P(i \wedge F\phi)$
E-S,R	Perfect	I have seen	$P\phi$
S,R,E	Present	I see	ϕ
S,R-E	Prospective	I am going to see	$F\phi$
S-E-R	Future perfect	I will have seen	$F(i \wedge P\phi)$
S,E-R	Future perfect	I will have seen	$F(i \wedge P\phi)$
E-S-R	Future perfect	I will have seen	$F(i \wedge P\phi)$
S-R,E	Future	I will see	$F(i \wedge \phi)$
S-R-E	Future-in-the-future	(Latin: abiturus ero)	$F(i \wedge F\phi)$

We could sum up these entries by saying that we have analysed the English tenses into a 'shift' (performed by the tense operators) and a 'refer' (performed by the nominals). The 'shift' component is Prior's contribution, the 'refer' Reichenbach's.

Some further comments on these representations are called for. First note that all three logically possible permutations of *R*, *E* and *S* for both the Future-in-the-past and the future perfect are represented by a single formula. Such a mode of representation seems to be preferred by linguists. Comrie, for example, argues at some length that such non-committal representations are essential [5, pages 26 – 27]; indeed he goes so far as to criticise Reichenbach's diagrams for their specificity. His argument is that while one of the three possibilities may be favoured, and one largely ruled out, these preferences are merely Gricean implicatures: all three usages are possible, and a uniform representation is required. Now, it's not clear to me whether there is much of real substance at stake here, nonetheless the fact remains that such non-committal representations are easily provided in NTL.

Second, note that the perfect and the past tenses are represented differently. The identical representations these two tenses receive in standard tense logic is one of the major sources of dissatisfaction with the use of tense logic in the analysis of natural language. However even though it is pleasant that there is a distinction between the representation of these two tenses in NTL, I think it would be foolish to attach too much weight to this. The distinction has been achieved by adding something useful — a Reichenbachian component — to Prior's account of the past tense. On the other hand we have added nothing to the (unsatisfactory) Priorian account of the perfect. The perfect poses a number of difficult problems, and until these are solved the representational distinction achieved above will have a slightly hollow ring.

The table does not exhaust all the possibilities, even for the case of English: as Prior observed [21, page 13], certain sentences seem to require more than one point of reference. Consider Prior's example, "I shall have been going to see John". This seems to require two points of reference, *R*₁ and *R*₂. The pragmatically most plausible pattern they will exhibit is *S* – *R*₂ – *E* – *R*₁. The semantic possibilities are captured in NTL as follows:

$$F(i \wedge P(j \wedge F(\text{See John}))).$$

Comrie [5][6, pages 122-130] develops this idea to its logical conclusion. He proposes a modification of Reichenbach's system which essentially amounts to allowing an arbitrary number of reference points.¹ The possible tenses are then defined to be linkages between the

¹This is not quite accurate: Comrie also replaces Reichenbach's simultaneity relation with a relation of intervalic overlap. This idea is easy to capture in sorted interval based languages.

point of speech and the point of event mediated by a (possibly null) sequence of reference points. This idea isn't made completely precise, but the following seems to capture what Comrie intends. Let ϕ and ψ be metavariables over wffs and n a metavariable over nominals. We take as primitive tensed forms all instances of $P\phi$, ϕ , and $F\phi$. We then say that if ψ is a tensed form then so are $P(n \wedge \psi)$, $(n \wedge \psi)$, and $F(n \wedge \psi)$. Everything in the previous table falls under this definition, and according to Comrie, so do many of the tense forms exhibited in natural languages.

3 Intersentential anaphora

In this section we use nominals to model temporal intersentential anaphora. We approach this subject via Partee's paper "Nominal and Temporal Anaphora" [20]. Partee's discussion, which draws on earlier work by Hinrichs [10], is couched in the language of temporal DRT [12][13]. In the remainder of this section will see how to deal with Partee's examples can be dealt with in NTL. The key idea is to use nominals as discourse markers — or, to put a more Reichenbachian slant on it, to think of points of reference (as Reichenbach himself did) from the standpoint of intersentential anaphora.

Consider the discourse "The shutters were closed. It was dark." As Partee points out, the tense in the second sentence (which is a *state* sentence) anaphorically picks out the time referred to by the tense of the first sentence (which is also a *state* sentence). We can represent this discourse in NTL as follows:

$$P(i \wedge \text{The shutters be closed}) \wedge P(i \wedge \text{It be dark}).$$

Note that we have reused the i nominal provided by the first sentence in the representation of the second. This really does give us what is desired, namely reference to a past time that is both a shutters-being-closed-time and an it-being-dark time. Although the two occurrences of the i nominal are in different conjuncts and embedded within the scopes of different past tense operators, we achieve the desired semantic effect, as it follows from the semantics of NTL that the above representation is logically equivalent to

$$P(i \wedge \text{The shutters be closed} \wedge \text{It dark}).$$

Because nominals are 'hardwired' to a unique time, no matter how deeply they are embedded within the structure of our representations they are always available for reuse.

This ability of nominals to 'float out of the scope' of tense operators means we can build up discourse representations online. Consider the above discourse once more. Imagine that we are processing this discourse, building up its representation as we go. First we process "The shutters were closed". The parser goes to work and builds $P(i \wedge \text{The shutters be closed})$. The second sentence is then analysed. Because it is a sentence in the simple past tense a representation of the form $P(n \wedge \text{It dark})$ must be built, where n is to be instantiated with some nominal. But which nominal should be used? Because the second sentence is a *state* sentence that follows hot on the heels of another, the parser reuses the i of the previous representation. As we have just seen, this natural strategy is semantically correct.

Matters are more interesting when we have a sequence of two or more *event* sentences. Consider the discourse: "Mary woke up sometime during the night. She turned on the light." Representing this in NTL by

$$P(i \wedge \text{Mary wake up sometime during the night}) \wedge P(i \wedge \text{She turn on the light})$$

is inadequate: it is clear that the illumination should follow the awakening. The event verb of the second sentence has advanced the referential focus. This is a typical feature of event

sentences. Whereas state sentences elaborate on the time in focus, one of the functions of event sentences is to move the story along.² It is simple to advance the reference time in the fashion required by event sentences: we need merely introduce a new reference marker (that is, a new nominal) and insist that it names a more recent time than did the old. Here's how this can be done:

$P(i \wedge \text{Mary wake up sometime during the night}) \wedge P(j \wedge Pi \wedge \text{She turn on the light}).$

Note how this works. The reference time of the first sentence is picked out by *i*. The second sentence then advances the reference time, and we capture the effect of this as follows. First we introduce a new nominal *j* to pick out the new reference time, and then by adding an extra conjunct *Pi* we guarantee that the old reference time lies in the past.

As a third example consider the following discourse which mixes stative and event verbs: "John got up, went to the window, and raised the blind. It was light out. He pulled the blind down and went back to bed. He wasn't ready to face the day. He was too depressed." We can represent this as follows:

$$\begin{aligned} & P(i \wedge \text{John get up}) \\ \wedge & P(j \wedge Pi \wedge \text{go to the window}) \\ \wedge & P(k \wedge Pj \wedge \text{raise the blinds}) \\ \wedge & P(k \wedge \text{It be light out}) \\ \wedge & P(i_1 \wedge Pk \wedge \text{He pull blind down}) \\ \wedge & P(j_1 \wedge Pi_1 \wedge \text{go back to bed}) \\ \wedge & P(j_1 \wedge \text{He be not ready to face the day}) \\ \wedge & P(j_1 \wedge \text{He be too depressed}) \end{aligned}$$

This representation merely amounts to 'shift and refer' coupled with 'advance the reference time when an event verb is encountered'.

I'll conclude my examination of intersentential anaphora here. The basic idea — using nominals as analogs of discourse markers — should now be clear. But does this analogy tell us anything useful? I believe it may yield a strategy for investigating inference in DRT.

Although DRT is probably the most influential contemporary semantic theory, relatively little is known about its logical properties. On the other hand, modal logics have been extensively investigated, and although NTL is a non-standard modal logic, it turns out that standard modal techniques can be straightforwardly adapted to it; see [3] for a number of completeness and decidability results.

Experience with NTL indicates that modal languages with nominals are in certain respects rather like systems of temporal DRT. It would be interesting to see whether substantial fragments of DRT could be translated into NTL, or (say) one of its sorted interval based cousins. In this fashion one might be able to induce proof theories onto DRT.

4 Indexicals

In this section we will see how sorting can be used to model the indexicals 'yesterday', 'today' and 'tomorrow'. We add three new symbols — *yesterday*, *today* and *tomorrow* — to NTL. These new symbols are atomic wffs and may be freely combined with other wffs to make more complex wffs. They will be used to represent English sentences containing

²This is a considerable simplification of a complex phenomena, however my aim here is to show how to mimic Partee's analyses, not to explore difficult issues in the temporal structure of narrative.

indexicals. For example, we will be able represent "John climbed yesterday" as

$$P(\text{yesterday} \wedge \text{John climb}),$$

and this representation will correctly capture the fact that "John ran yesterday" is true iff John did in fact run at some point during the day immediately preceding the day of utterance.

But how are we to specify the semantics of these new items so that they correctly mimic the behaviour of the English originals? An elegant answer lies to hand: we shall make use of semantic machinery developed by Montague [16], Kamp [11] and Kaplan [14][15], namely the machinery underlying the California theory of reference. In spite of its age this theory remains one of the most detailed responses to the problems posed by indexicals. As we shall see, the main ideas of the Californian approach mesh smoothly with the techniques of the present paper, and in fact the major difference between this paper and the earlier work is not conceptual but technical. Hitherto the semantic machinery of the California theory of reference was exploited by means of additional modal operators. We exploit the same machinery here, but not with operators: instead we sort. This leads to a clean account of the basic interactions between tense and indexical reference.

Let's begin with the simplest task, defining the syntax of our language. This is NTL augmented with three new atomic symbols: *yesterday*, *today* and *tomorrow*. The set of atomic symbols of this language is defined as follows:

$$\text{ATOM} = \text{VAR} \cup \text{NOM} \cup \{\text{yesterday}, \text{today}, \text{tomorrow}\}.$$

We make wffs from this stock of symbols exactly as before. That is, WFF is defined to be the smallest set containing ATOM that is closed under negation, conjunction and the application of tense operators.

Now for the task of defining the semantics. First of all, we need to pay more attention to our choice of temporal ontology. These indexicals all presuppose the concept of dayhood, thus it is sensible to work with temporal structures rich enough to support a reasonably realistic picture of this assumed day structure. As there's no obvious single best choice, let's agree for the purpose of the present paper to work with a particularly simple option: $\mathbf{Q} (= \langle \mathbf{Q}, < \rangle)$, the rational numbers in their usual order. From now on we'll only consider models based on this frame. We can impose a reasonable looking day structure on $\mathbf{Q}$ as follows. Define the set of *Days* on $\mathbf{Q}$ to be $\{[z, z+1) : z \in \mathbf{Z} (\subseteq \mathbf{Q})\}$. Here $\mathbf{Z} (= \langle \mathbf{Z}, < \rangle)$ is the set of integers, and we're using them as a simple (and arbitrary) way to mark off the division between successive days. We also make use of the following three functions. The function *day* maps elements q of $\mathbf{Q}$ to the unique element of *Days* that contains q . That is, given a point of time, *day* answers the question 'What day does this point belong to?'. The functions *next* and *prev* are both maps from *Days* to *Days*. In particular, for all integers z , $\text{next}([z, z+1)) = [z+1, z+2)$ and $\text{prev}([z, z+1)) = [z-1, z)$, so these functions take as input a day and return, respectively, the next day and the previous day. This is all we'll need to assume in the way of day structure for the purposes of this paper.

With these preliminaries established, we are now ready to introduce the apparatus of the California theory of reference. The key idea of the California theory of reference is that context must be taken into account when evaluating the truth of utterances. This simple idea is made precise as follows: contexts are added to Kripke models, and when wffs are evaluated they are evaluated not only at a certain time, but also at (or in) a certain context of utterance.

Let's make this precise. Fix some non-empty set C , the set of *contexts*, or *contexts of utterance*, and specify a function $g : C \rightarrow \mathbf{Q}$. The function g is to be thought of as

specifying the *utterance time* of each context of utterance. We call the quadruple $\langle Q, <, C, g \rangle$ a *contextualisation of Q*. Clearly this is an extremely meagre notion of context, nonetheless it suffices for the phenomena we're going to consider.

We are now ready to interpret our language. As always with sorted systems, the key point is to say what we mean by a valuation. Here is the definition. A valuation is a function $V : ATOM \times C \rightarrow Pow(Q)$ that respects the following constraints. First, for each nominal i and each context c , $V(i, c)$ must be a singleton. Second, for each context c :

$$\begin{aligned} V(today, c) &= day(g(c)) \\ V(tomorrow, c) &= next(day(g(c))) \\ V(yesterday, c) &= prev(day(g(c))) \end{aligned}$$

The intuition behind these constraints should be clear. In any context c the *today* atom is to be true at all points in the day containing the utterance time and at no others. That is, *today* picks out precisely those points belonging to the 'day of utterance'. Next, *yesterday* and *tomorrow* are to be true at, respectively, precisely those points in the day preceding, and the day succeeding, the day of utterance. Finally, note that within a fixed context of utterance c , nominals (as before) pick out a unique time. This definition allows the possibility that in different contexts of utterance the same nominal i may be used to name different times.

In giving these atomic level stipulations we have essentially completed our task. The following clauses explain how to interpret arbitrary wffs, but as the reader familiar with the Californian theory will observe, they are standard: everything of importance has already taken place. So, define model M to be a pair $\langle \langle Q, <, C, g \rangle, V \rangle$ where $\langle Q, <, C, g \rangle$ is a contextualisation of Q and V is a valuation. Then for all times $t \in Q$ and all contexts $c \in C$ we define:

$$\begin{aligned} M \models a[t, c] &\text{ iff } t \in V(a, c), \text{ for all atoms } a \\ M \models \neg\phi[t, c] &\text{ iff not } M \models \phi[t, c] \\ M \models \phi \wedge \psi[t, c] &\text{ iff } M \models \phi[t, c] \text{ and } M \models \psi[t, c] \\ M \models F\phi[t, c] &\text{ iff } \exists t'(t < t' \text{ and } M \models \phi[t', c]) \\ M \models P\phi[t, c] &\text{ iff } \exists t'(t' < t \text{ and } M \models \phi[t', c]) \end{aligned}$$

Let's put these enrichments to work. Consider the sentence "John ran yesterday". We could represent this as, $F(yesterday \wedge John \text{ run})$. This representation captures the temporal force that the 'yesterday' adds to "John ran". For suppose we're in some context of utterance c , and suppose we evaluate $F(yesterday \wedge John \text{ run})$ at the pair $[g(c), c]$. That is, we're asking ourselves what happens if we 'utter this wff' in the context c at the utterance time for c . Then it follows from the above truth definition that this wff is true iff there is a time t' in the day preceding the utterance day such that *John run* is true at $[t', c]$. In short, this wff is true iff John did run during the day classified as the yesterday of the context of occurrence.

Now this is precisely what is required. Moreover, as many temporal semanticists have pointed out, it's something that multiple operator approaches have difficulty achieving. To take a recent example, Oversteegen [17, page 3] complains that:

If temporal adverbs like *yesterday* are treated as operators, there is a scope problem: both sequences, *yesterday* in the scope of the tense operator and the tense operator in the scope of *yesterday*, are obviously wrong.

Such scoping problems simply don't arise with the sorting strategy. The reason is simple. Our indexicals don't in any sense quantify, rather they act as a range restrictors on the quantification performed by the familiar temporal operators. For example the role of *tomorrow* in the wff $F(tomorrow \wedge John \text{ run})$ is to ensure that only those times classified as being

'tomorrow' are relevant to the truth or falsity of the wff. The familiar problems resulting from operator interactions are bypassed.

This talk of restricting the range of temporal quantification bears on the work of the previous section. There we talked of factoring tenses into a 'shift' and a 'refer': the tense operators pointed us in the desired direction, and nominals filled the 'referential slot' giving us the desired specificity. But it is clear that the mechanism underlying both $F(i \wedge \text{John run})$ and $F(\text{tomorrow} \wedge \text{John run})$ is essentially the same. Not only do these wffs share a common syntactic form, i and *tomorrow* share the common semantic role of restricting the times of relevance to the F operator.

Thinking in this more general fashion in terms of quantifications and restrictions seems a useful way of thinking about temporal representations. This is particularly apparent when it comes to building up discourse representations. Consider the following discourse: "John went running yesterday. It was wet and cold and nasty." This can be represented as follows:

$$\begin{aligned} & P(i \wedge \text{yesterday} \wedge \text{John run}) \\ \wedge & P(i \wedge \text{Wet and cold and nasty}) \end{aligned}$$

Note that the representation of the first sentence contains *two* restrictors: the nominal i and *yesterday*. As described in the previous section, the nominal is contributed by the simple past tense of 'went'. It gives us a name for the reference point. The *yesterday* is contributed by the word 'yesterday'. Its function is to constrain where the running took place, namely to 'within yesterday'. Together the two restrictors enable the temporal information encapsulated in "John went running yesterday", to be captured, and crucially part of this representation (the i) is reused to correctly anchor the second sentence.

Our discussion has got rather general. Let's return to specifics. Another pleasant feature of these restricting representations is that they handle certain semantically anomalous sentences rather nicely, namely those sentences where the shift demanded by the tense clashes with the reference demanded by the indexical. For example, both

*John will run yesterday

and

*John ran tomorrow

are semantically anomalous in this sense. Now many formalisms don't correctly handle such interactions. In fact Harper and Charniak [9] list the ability to handle such examples as one of their five desiderata for any system of temporal representation. But our sorted language handles the matter straightforwardly. Consider the obvious representations,

$$F(\text{yesterday} \wedge \text{John runs})$$

and

$$P(\text{tomorrow} \wedge \text{John runs}).$$

Neither wff is satisfiable in any any model. That is, no matter what model we choose, no matter what pair $[t, c]$ we evaluate at, both wffs evaluate to false. To some extent our language is mimicking the way temporal reference and tense interact in English.

5 Further extensions

This section sketches some further extensions of the sorting strategy. First we see how calendar terms such as 'January', 'Friday', 'March' and '1984' can be modeled, and then

we conclude the paper by noting the possibilities raised by sorting in the richer setting of interval based logic.

We extend our language to deal with calendar terms by adding five further sorts:

DAY	=	{ <i>Sunday</i> , ..., <i>Saturday</i> }
DATE	=	{1, ..., 31}
MONTH	=	{ <i>January</i> , ..., <i>December</i> }
YEAR	=	{1, 2, 3, ..., 1989, 1990, ...}
ERA	=	{ <i>BC</i> , <i>AD</i> }

We define CAL to be $\text{DAY} \cup \text{DATE} \cup \text{MONTH} \cup \text{YEAR} \cup \text{ERA}$ — we assume that all these sets are mutually disjoint from each other and from all the sorts we already have — and then define:

$$\text{ATOM} = \text{VAR} \cup \text{NOM} \cup \{\text{yesterday}, \text{today}, \text{tomorrow}\} \cup \text{CAL}.$$

The wffs are made from ATOM in the usual fashion. As an example of how we are going to use this language, the sentence “John ran on Monday 4th December 1989” will be represented by:

$$P(\text{Monday} \wedge 4 \wedge \text{December} \wedge 1989 \wedge \text{AD} \wedge \text{John run}).$$

I'll now sketch the semantic definitions that ensure that this wff does its work properly. Recall from the last section that we are only considering models based on the frame Q , and that we have defined what we mean by a day structure over Q . Thus it is relatively straightforward — if a little tedious — to define the needed *calendar structure* to interpret our new atoms. The required calendar structure *Cal* is a 4-tuple:

$$\text{Cal} = \langle \text{Weeks}, \text{Months}, \text{Years}, \text{Eras} \rangle.$$

Each of its four components, *Weeks*, *Months*, *Years* and *Eras*, is a partition of Q .³ We assume without further ado the natural induced ordering of *Weeks*, *Months*, *Years* and *Era*. For example, if $m, m' \in \text{Months}$ we say that $m < m'$ iff $\forall t \in m \forall t' \in m' (t < t')$.

We want to place the obvious conditions on each of these partitions so that *Cal* really does look like a calendar. Consider the case of *Years*, for example. First, define a *year* to be the set of points contained in 365 or 366 consecutive elements of *Days*.⁴ We then define the set *Years* in the calendar structure to be any ‘leap year structured’ partition of Q such that every element of the partition is a *year*. By ‘leap year structured’ is meant that every fourth element in *Years* is a *year* containing 366 days, while the remaining three out of four *years* contain only 365 days.

We construct *Weeks*, *Months* and *Eras* in similar fashion. Every *week* in *Weeks* is made up of the points in seven consecutive days; every *month* in *Months* is made up of the points in either 28, 29, 30, or 31 consecutive days; and *Eras* partitions Q into two. However as well as ensuring that each of these four sets *Weeks*, *Months*, *Years* and *Eras* is correctly structured, we must also ensure that they ‘hang together’ correctly. For example, the end of the first era should coincide with the end of some year; the start of a year should coincide with the start of a month containing 31 days, while the second month in any year should contain 28 days, unless the year is a leap year when it must contain 29; and so on. Stating these details

³ By a partition of Q is meant a subset P of $\text{Pow}(Q)$ such that $\bigcup P = Q$, each $p \in P$ is non-empty, and for all $p, p' \in P$, $p \neq p'$ implies $p \cap p' = \emptyset$.

⁴ Recall from the previous section that *Days* is the set of days in the day structure, and further recall that both successor and predecessor functions on days are defined as part of the day structure. Thus ‘consecutive days’ are defined in the obvious way using this apparatus.

exhaustively would be exhausting; I'll assume that the reader understands what is intended and proceed.

We must now describe what we mean by valuations. These are maps from

$$V : \text{ATOM} \times C \longrightarrow \text{Pow}(T),$$

which satisfy the constraints of the previous section, and which in addition satisfy the needed constraints concerning the sorts in CAL. These constraints, though obvious, are wearisome to spell out in detail, so I'll only give them for the simplest case, the two atoms BC and AD in ERA. Recall that Era is a binary partition of Q ; let us write this partition as $\{bc, ad\}$, where $bc < ad$. If V is to qualify as a valuation it must satisfy the demands that:

$$\begin{aligned} V(BC, c) &= bc \\ V(AD, c) &= ad, \end{aligned}$$

for all $c \in C$. I leave the (rather thankless) task of precisely stipulating the constraints needed for the other new sorts to the reader; the principle should be clear.

We now have language with a fairly wide referential repertoire. Further extensions in a similar vein are possible, and relatively routine to define. For example, we could add 'clock terms' to the language, and define 'clock structures' in a manner that would allow "John reached the summit at three o'clock" to be represented by

$$P(\text{three o'clock} \wedge \text{John reach the summit of Ruapehu}).$$

But there is a potentially much more interesting direction to explore: the abandonment of the point based setting in favour of sorting an interval based language. I'll conclude the paper with some remarks on this topic.

Like the Priorean language, interval modal languages are relatively simple systems which describe temporal structure from an internal perspective. They tend to be slightly more syntactically complex than the (essentially Priorean) languages we have explored in this paper, because they typically have a wider repertoire of operators. This is because these languages quantify over a more complex temporal ontology, and additional operators are needed to make the desired distinctions.

Let's consider an example. A (fairly simple) choice of intervalic ontology is the following: by an interval structure I is meant a 4-tuple $\langle T, <, \sqsubseteq, o \rangle$, where T is a non-empty set, the set of intervals, and $<$, $\sqsubseteq$, and o are all binary relations on T . These relations are thought of as the *precedence*, *subinterval* and *overlap* relations respectively. Of course it is necessary to specify further properties of these relations and their mutual interactions if these interval structures are to look realistic. We won't do so here but refer the reader to [2] for a thorough discussion of the various options.

To describe this interval structure we might use the following language. As primitive symbols we have a set VAR of atomic symbols, written p , q , r and so on, and the logical symbols $\neg$, $\wedge$, F , P , $\downarrow$ and O . We make wffs from this collection of symbols in the obvious way. To interpret this language we again use Kripke models. In the present setting a Kripke model M is a pair $\langle I, V \rangle$ where $I (= \langle T, <, \sqsubseteq, o \rangle)$ is an interval structure and V is a valuation that assigns subsets of T to atomic symbols.

We interpret wffs in models in the expected fashion. The clauses for the atomic symbols, and for $\neg$, $\wedge$ F and P are precisely as for Priorean languages. The two new clauses are as follows:

$$\begin{aligned} M \models \downarrow \phi[t] &\quad \text{iff} \quad \exists t'(t' \sqsubseteq t \text{ and } M \models \phi[t']) \\ M \models O\phi[t] &\quad \text{iff} \quad \exists t'(t' o t \text{ and } M \models \phi[t']) \end{aligned}$$

That is, $\downarrow\phi$ is true at an interval t iff ϕ is true at a subinterval t' of t ; and $O\phi$ is true at an interval t iff ϕ is true at an interval t' that overlaps t .

Now this new language is certainly more expressive than the Priorean language, but it also suffers from a familiar weakness: there is no mechanism for temporal-reference. However (as should now be apparent) this weakness is easily removed by sorting. For example, let's add a second sort INOM, the sort of interval nominals, to the language. We'll write these new symbols as e, d, c and so on and insist that in any valuation these items are to be true at precisely one interval. Thus we could represent "John ran" by $P(e \wedge \text{John run})$. This asserts that *John run* is true at some past interval, namely the one picked out by e . Note that it's a somewhat better representation than that offered by NTL: the use of e — which can be true over an *extended* interval — captures the fact that the speaker was referring to an extended period of time. It should also be clear that we can go on to adapt the California theory to the new setting and thus deal with indexicals.

But although doing these things will lead to some improvements, they are by and large just refinements of what we have already achieved in the point based setting. The real challenge of interval logic is that because of its richer ontology it offers a broader canvas on which to draw distinctions: distinctions which can subsequently be exploited by sorting. As an introduction to the possibilities I'll mention the following. First, we could draw a useful distinction between referring to extended interval and referring to point intervals. That is, instead of having simply interval nominals, we could have a syntactic distinction between those interval nominals which hold at extended intervals, and those which hold at minimal intervals. Second, we could attempt to give formal substance to the idea of an 'extended now'. It becomes natural to think of the time denoted by 'now' as a small, but extended, interval. By giving this a special name (say *now*) we can further investigate the interactions between tenses and temporal adverbials. Third, by adding operators to pick out intervals 'just before' the present one, we can deal with such sentences as "When John makes a phone call, he always lights up a cigarette beforehand," a sentence which Partee's system of temporal DRT cannot deal with.

Indeed there are many reasons for investigating sorting in interval based logics. However precisely because of the richness of this setting, a lot more work is needed to choose a good set of tools: the choices simply aren't as obvious as in the point based setting. Which sorts and which operators work well together to give good representations of natural language sentences? Only further work can determine this.

In the meantime I'll content myself with remarking that even in the rather simplistic setting of point based tense logic, combining Prior's 'internal perspective' with the idea of sorting allows a surprising amount to be achieved. Prior's account of temporal language may be criticised for being insufficiently detailed, but his ideas seem fundamentally sound. In this paper we have seen that there is a simple way of extending his account that is faithful to the spirit of his work.

Acknowledgements I would like to thank Inge Bethke, Claire Gardent, Hans Kamp, Alex Lascarides and Jerry Seligman for their advice, criticism and (not least) their patience in discussing the ideas of this paper.

References

- [1] Benthem, J. v.: 1983, *Modal Logic and Classical Logic*, Bibliopolis, Napoli.
- [2] Benthem, J. v.: 1991, *The Logic of Time*, 2nd edition, Kluwer Academic Publishers.

- [3] Blackburn, P.: 1990, *Nominal Tense Logic and other Sorted Intensional Frameworks*. PhD Thesis, Centre for Cognitive Science, University of Edinburgh, Scotland.
- [4] Bull, R.: 1970, An Approach to Tense Logic, *Theoria*, 12, pages 171–182.
- [5] Comrie, B.: 1981, On Reichenbach's Approach to Tense, in *Papers from the Seventeenth Regional Meeting, Chicago Linguistics Society*, edited by Hendrick et al., pages 24–30.
- [6] Comrie, B.: 1985, *Tense*, Cambridge University Press.
- [7] Gargov, G., Passy, S., Tinchev, T.: 1987 Modal Environment for Boolean Speculations (preliminary report). In *Mathematical Logic and its Applications*, edited by D. Skordev, Proceedings of the 1986 Summer School and Conference, honourably dedicated to the 80-th anniversary of Kurt Godel, Druzhba, Bulgaria. Plenum Press: New York and London.
- [8] Gargov, G., Goranko, V.: 1989 Modal Logic with Names. Submitted manuscript, Linguistic Modelling Laboratory, CICT, Bulgarian Academy of Sciences, Sofia; and Sector of Logic, Faculty of Mathematics and Computer Science, Sofia University.
- [9] Harper, M., Charniak, E.: 1986, Time and Tense in English, *Proceedings of the ACL conference*.
- [10] Hinrichs, E.: 1982, Temporal Anaphora in Discourses of English, paper presented at the Ohio State University Conference on the Semantics of Tense and Aspect in Discourse.
- [11] Kamp, H.: 1971, Formal properties of 'now', *Theoria*, 37, pages 227–273.
- [12] Kamp, H.: 1981, A Theory of Truth and Semantic Representation, in J. Groenendijk, T. Janssen, and M. Stokhof (eds), *Formal Methods in the Study of Language, Part 1*, Mathematisch Centrum, Amsterdam, pages 277–322.
- [13] Kamp, H., Rohrer, C.: 1983, Tense in Texts, in Bäuerle, Schwarze, and von Stechow (eds.), *Meaning, Use, and Interpretation of Language*, Walter de Gruyter, Berlin, pages 250 – 269
- [14] Kaplan, D.: 1977, Demonstratives: An Essay on the Semantics, Logic, Metaphysics and Epistemology of Demonstratives and Other Indexicals, Draft No 2, UCLA xerox.
- [15] Kaplan, D.: 1979, On the Logic of Demonstratives, in *Contemporary Perspectives in the Philosophy of Language*, edited by French, Uehling and Wettstein, pages 401–412, Minneapolis: University of Minnesota Press.
- [16] Montague, R.: 1970, Universal Grammar, in *Formal Philosophy. Selected Papers of Richard Montague*, edited by Richmond Thomason.
- [17] Oversteegen, E.: 1989, *Tracking Time. A Proposal for the representation of Temporal Expressions in Dutch*. Proefschrift, Rijksuniversiteit te Utrecht.
- [18] Passy, S., Tinchev, T.: 1985, PDL with Data Constants. *Information Processing Letters* 20, pages 35–41.
- [19] Passy, S., Tinchev, T.: 1989, An Essay in Combinatory Dynamic Logic. Submitted Manuscript, Sector of Mathematical Logic and Laboratory for Applied Logic, Sofia University, Bulgaria.

- [20] Partee, B.: 1984, Nominal and Temporal Anaphora, *Linguistics and Philosophy*, 7, pages 243–286.
- [21] Prior, A.: 1967, *Past, Present and Future*, Oxford University Press.
- [22] Prior, A.: 1968, “Now”, *Nous*, pages 101–119.
- [23] Reichenbach, H.: 1947, *Elements of Symbolic Logic*. Random House: New York.

On the Complexity of Attribute Value Logics

Patrick Blackburn and Edith Spaan
Universiteit van Amsterdam

Attribute Value Structures (AVSs) are probably the most widely used means of representing linguistic structure in current computational linguistics, and the process of unifying descriptions of AVSs lies at the heart of many parsers. As a number of people have recently observed (see for example [17], [4], [19] and [25]) the most common formalisms for describing AVSs are notational variants of propositional modal languages, AVSs themselves are Kripke models, and unification amounts to looking for a satisfying model for $\phi \wedge \psi$ given two (modal) wffs ϕ and ψ . This paper examines the complexity of various unification tasks of interest to computational linguistics by making use of this connection with modal logic. This approach allows standard ideas from modal logic to be used and often results in relatively simple arguments.

The paper is structured as follows. The first section begins with a brief introduction to such topics as 'attributes', 'values', and 'unification' and why they are of interest in computational linguistics. It then goes on to explain the link with modal logic, and gives the syntax and semantics of the three modal languages — L , L^N and L^{KR} — which correspond to the three most common unification formalisms. In the second section we examine the satisfiability problems for these languages, and show that all three are NP complete. We briefly discuss the import of these results. In the third section we introduce two stronger languages, $L^{KR\Box}$ and $L^\Box$. These are L^{KR} and L , respectively, augmented by the universal modality $\Box$. Adding this modality allows general constraints on linguistic structure to be expressed in a propositional modal language. As we will show, however, there is a price to pay: $L^{KR\Box}$ is undecidable. This means that even in a propositional modal language it is asking a lot to demand both the ability to enforce re-entrancy and to express generalisations. We then go on to show that dropping the ability to enforce re-entrancy results in a decidable system. In fact we show that the satisfiability problem for $L^\Box$ is EXPTIME complete. We close the paper by briefly discussing some of the issues arising from this work.

The paper is relatively self contained, however we do assume that the reader understands what is meant by such complexity classes as NP, PSPACE and so on; such definitions may be found in [1], for example. For further information on modal logic the reader is referred to [12]; and for more on unification based grammar, [30] is useful.

1 Attribute Value logic

Even the most cursory examination of recent proceedings of computational linguistics conferences reveals that there is a substantial level of interest in such topics as 'attributes', 'values', and 'unification'. This section presents a brief introduction to these topics, and explains what they have to do with modal logic. The basic point it makes is that the most common machinery underlying Attribute Value grammar formalisms is simply that of propositional modal logic, and that the unification process used as part of the parsing process for such grammars is nothing but building satisfying models for wffs of these languages. This

correspondence provides the *raison d'être* of the paper: by examining the complexity of the satisfiability problem for the modal languages involved, we learn — quite straightforwardly — about the complexity of tasks of interest to computational linguistics.

Perhaps the best way of approaching these topics is via Attribute Value Matrices (AVMs), or Feature Value Matrices as they are sometimes called. A (rather simple) AVM might look something like this:

$$\begin{bmatrix} \text{CASE} & \textit{nominative} \\ \text{AGREEMENT} & [\text{PERSON } 1st] \end{bmatrix}$$

Such an AVM is taken to be a partial description of some piece of linguistic structure. In this case we are partially describing a piece of linguistic structure that has two attributes, namely CASE and AGREEMENT. The CASE attribute takes as value the atomic value *nominative*, while the AGREEMENT attribute takes as value the complex entity [PERSON 1st]. This complex entity consists of an attribute PERSON that takes as value the atomic value *1st*. The particular atomic values (or constants) and attributes (or features) that may occur in AVMs varies widely from theory to theory, but typical choices of atomic entities a syntactician might make are *singular, plural, 3rd, 2nd, 1st, genitive* and *accusative*; and when it comes to a choice of attributes the same person might choose TENSE, NUMBER, PERSON, AGREEMENT, and CASE. But although the different theories differ on the particular choices made, and indeed in the uses they put this machinery to, they are united in agreeing that at least a part of our descriptions of linguistic structure should embody the idea of attributes taking (possibly complex) values.

The information expressed by AVMs can be considerably more complex than in the above example. Many linguists feel it is necessary to be able to express both disjunctive and negative information in their Attribute Value grammars. To give two well known examples due to Karttunen [14], one might write

$$\begin{bmatrix} \text{NUMBER} & \textit{plural} \\ \text{CASE} & \{\textit{nominative, genitive, accusative}\} \end{bmatrix}$$

an AVM which states that the attribute CASE takes one of the values *nominative, genitive, or accusative*, but doesn't say which; or one might write

$$\begin{bmatrix} \text{NUMBER} & \textit{plural} \\ \text{CASE} & [\neg \textit{dative}] \end{bmatrix}$$

an AVM which specifies that CASE doesn't take the value *dative*.

What do computational linguists do with AVMs? The answer is, they try to unify them. Intuitively, unifying two AVMs means forming another AVM that combines all the information about Attribute Value dependencies contained in the two constituent AVMs. For example, writing $\sqcup$ to indicate unification, we have:

$$\begin{bmatrix} \text{AGR} & [\text{PER } 1st] \\ \text{CASE} & \textit{nominative} \end{bmatrix} \sqcup [\text{AGR } [\text{NUM } \textit{plural}]] = \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUM} & \textit{plural} \\ \text{PER} & 1st \end{bmatrix} \\ \text{CASE} & \textit{nominative} \end{bmatrix}$$

There is a clear sense in which the AVM on the right hand side embodies all the information in the two constituent structures; it is the result of unifying these structures.

But this is not very precise. What *precisely* is unification? Answering this question will lead us first to AVSs, the semantics of AVMs, and then, quite naturally, to the link with modal languages.

AVSs are certain kinds of *decorated labeled graphs*. Such graphs play the central role in unification based linguistics: they are the mathematical model of linguistic structure underlying these frameworks. A number of definitions of AVSs exist in the literature. We shall work with a particularly simple one:

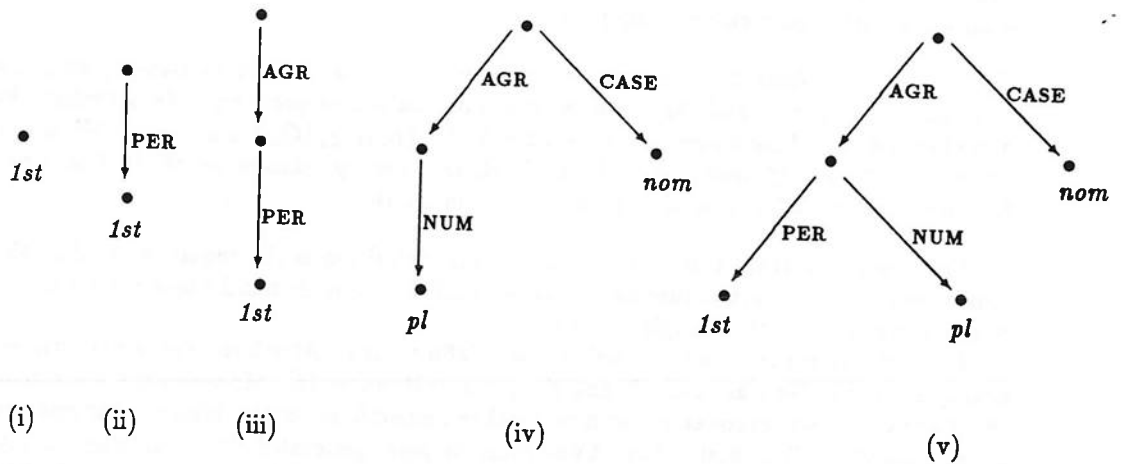
Definition 1.1 (Attribute Value Structures) *Let $\mathcal{L}$ and $\mathcal{A}$ be non-empty enumerable sets, the set of labels and the set of atomic information respectively. An Attribute Value Structure (AVS) of signature $\langle \mathcal{L}, \mathcal{A} \rangle$ is a triple $\langle W, \{R_l\}_{l \in \mathcal{L}}, \{Q_\alpha\}_{\alpha \in \mathcal{A}} \rangle$, where W is a non-empty set, the set of nodes; for all $l \in \mathcal{L}$, R_l is a binary relation on W that is a partial function; and for all $\alpha \in \mathcal{A}$, Q_α is unary relation on W . $\square$*

The most important thing to note about this definition is the requirement that all the binary relations be partial functions. As we shall see, this demand plays a crucial role in establishing some of our complexity results.

The definition covers all the well known definitions of Attribute Value Structures, for example those of Gazdar *et al.* [8] and Kaspar and Rounds [15]. Moreover it's not too loose: there are only two reasonably common further restrictions on the binary relations that it doesn't insist on. The first is that AVSs must be *point generated*. In point generated AVSs there is always a starting node $w_0 \in W$ such that all other nodes $w \in W$ are reachable via transition sequences from w_0 . The second is that AVSs must be *acyclic*, which means that it is never possible to return to a node w by following some sequence of R_l transitions from w . As neither of these restrictions plays a prominent role in the linguistics literature anymore, we ignore them here.

This definition also ignores three constraints computational linguists used to routinely place on node decoration. The constraints in question are these. First, for all $w \in W$ and all $\alpha, \beta \in \mathcal{A}$, if $w \in Q_\alpha$ and $\alpha \neq \beta$ then $w \notin Q_\beta$. That is, the constraint forbids what linguists call 'constant-constant clashes'. Second, for all $w \in W$, and all $\alpha \in \mathcal{A}$, $w \in Q_\alpha$ iff w is a terminal node. This constraint rules out 'constant-compound clashes'. Third, for all $w, w' \in W$, if $w \in Q_\alpha$ and $w' \in Q_\alpha$ then $w = w'$. Once again, the main reason for ignoring these demands is that they no longer play the prominent role they once did. Indeed in more recent work in computational linguistics, particularly work in the Head Driven Phrase Structure Grammar (HPSG) framework, much use is made of *sorts* [22]; and sorts are essentially pieces of atomic information that don't obey the above three restrictions.

Let's consider some concrete examples of AVSs. Suppose we are working with some linguistic theory which contains among its theoretical apparatus the attributes PERSON, NUMBER, CASE and AGREEMENT, and the atomic information, *3rd*, *2nd*, *1st*, *plural* and *nominative*. That is, our linguistic theorising has specified a signature $\langle \mathcal{L}, \mathcal{A} \rangle$ such that $\{\text{PERSON, NUMBER, CASE, AGREEMENT}\} \subseteq \mathcal{L}$, and $\{\text{3rd, 2nd, 1st, plural, nominative}\} \subseteq \mathcal{A}$. Then the following graphs are all examples of AVSs of this signature:



What do AVSs have to do with AVMs? As has already been remarked, AVMs are partial descriptions of linguistic structure, and in fact the structure they describe is the structure embodied in the definition of AVSs. That is, AVMs are a formal language for describing linguistic structure, AVSs provide the interpretation for AVMs, and thus the relationship is that which always exists between semantic and syntactic entities: we talk of AVSs satisfying (or failing to satisfy) the AVMs. To return to our examples, the first graph, consisting of a single node decorated with the atomic information *1st*, satisfies the atomic AVM *1st*. Why? Because this atomic AVM demands a node decorated with the atomic information *1st*, and the first graph is such a node. The second graph satisfies the AVM [PERSON *1st*] at its root node. Why? Because this AVM demands a node in some piece of linguistic structure that has the following property: a transition along an *RPERSON* relation takes one to a node decorated with the information *1st*. The root node of the second graph has this property. As a last example, consider the fourth graph. This satisfies the AVM

$$\left[\begin{array}{ll} \text{AGREEMENT} & [\text{NUMBER } \textit{plural}] \\ \text{CASE} & \textit{nominative} \end{array} \right]$$

at its root node.

Now, we could give a precise definition of what it means for an AVS to satisfy an AVM, but in fact this would be a waste of energy, for, as we'll now see, the satisfaction relation between AVSs and AVMs is just a disguised version of something very familiar: the satisfaction relation between Kripke models and modal wffs. There are two facets to this correspondence, the semantical and the syntactical. We'll treat each in turn, beginning with the semantical.

Consider once more the definition of AVSs as triples $\langle W, \{R_l\}_{l \in \mathcal{L}}, \{Q_\alpha\}_{\alpha \in \mathcal{A}} \rangle$. Such triples are just (multimodal) Kripke models: each R_l interprets a modal operator $\langle l \rangle$, and each unary relation Q_α interprets the propositional symbol p_α . To be sure, multimodal Kripke models are usually presented as triples $\langle W, \{R_l\}_{l \in \mathcal{L}}, V \rangle$, where V is a valuation function from a collection of propositional symbols VAR to $\text{Pow}(W)$. (In such presentations the pair $\langle W, \{R_l\}_{l \in \mathcal{L}} \rangle$ is usually given a special name, namely *multiframe*.) But obviously there is no mathematical substance to this difference. Given a traditionally presented Kripke model $\langle W, \{R_l\}_{l \in \mathcal{L}}, V \rangle$, we have that $\langle W, \{R_l\}_{l \in \mathcal{L}}, \{V(p) : p \in \text{VAR}\} \rangle$ is an AVS of signature $\langle \mathcal{L}, \text{VAR} \rangle$; and conversely, given any AVS $\langle W, \{R_l\}_{l \in \mathcal{L}}, \{Q_\alpha\}_{\alpha \in \mathcal{A}} \rangle$, we have that

$\langle W, \{R_i\}_{i \in \mathcal{L}}, V \rangle$ is a Kripke model, where V is the function from the set of (α -indexed) propositional variables VAR to $\text{Pow}(W)$ defined by $V(p_\alpha) = Q_\alpha$. In short, every AVS is a Kripke model, and vice versa.

Now for the syntactical correspondence. Consider the following AVM.

$$\begin{bmatrix} \text{AGREEMENT} & [\text{PERSON } 1st] \\ \text{CASE} & \text{nominative} \end{bmatrix}$$

This corresponds to

$$\begin{aligned} & \langle \text{AGREEMENT} \rangle \langle \text{PERSON} \rangle 1st \\ \wedge & \langle \text{CASE} \rangle \text{nominative} \end{aligned}$$

The key point to grasp is that the function of the attributes AGREEMENT, PERSON and CASE in the AVM is precisely analogous to the function of the existential modalities $\langle \text{AGREEMENT} \rangle$, $\langle \text{PERSON} \rangle$ and $\langle \text{CASE} \rangle$ in the modal wff. The function of the attributes is to demand the existence of certain transitions in AVSs, the function of the modalities is to demand the existence of certain transitions in Kripke models. But AVSs are just Kripke models, and thus the equivalence of the description languages is clear. The rest of the correspondence is straightforward: atomic values correspond to propositional symbols, and the modal wff is in effect just a linearisation of the AVM. To put it more generally, AVMs are just modal wffs written in a particularly perspicuous manner.

This correspondence extends in the obvious manner to AVMs with full Boolean expressivity. For example corresponding to the following AVM:

$$\begin{bmatrix} \text{NUMBER} & \neg \text{plural} \\ \text{CASE} & \{\text{nominative}, \text{genitive}, \text{accusative}\} \end{bmatrix}$$

we have the wff

$$\langle \text{NUMBER} \rangle \neg \text{plural} \wedge \langle \text{CASE} \rangle (\text{nominative} \vee \text{genitive} \vee \text{accusative}).$$

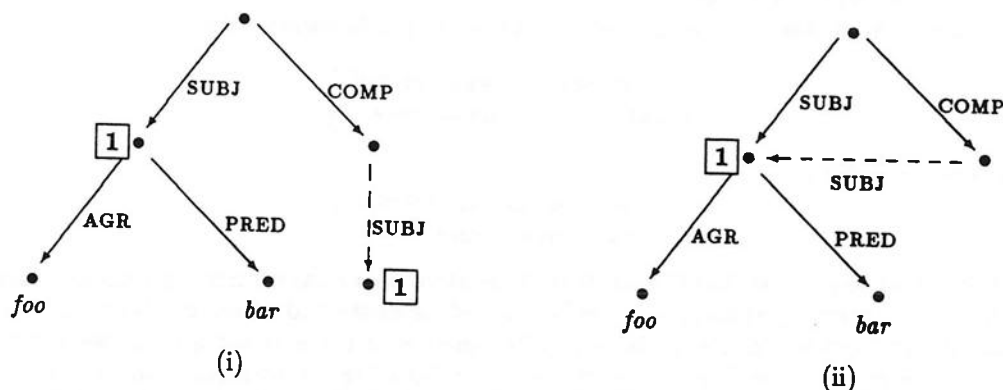
The most important aspect of the link between modal languages and AV formalisms is what it tells us about unification. Recall that unification is the attempt to coherently merge two AVMs. But what does 'coherent' mean? It means that the demands that the two AVMs make can be simultaneously satisfied at some node in some AVS. Now, both AVMs correspond to a modal wff. Call these two wffs ϕ and ψ respectively. Then we have that unification succeeds iff $\phi \wedge \psi$ is satisfiable at some node in some Kripke model. That is, *unification amounts to testing for modal satisfiability*. This observation lies at the heart of the paper. By investigating the complexity of the satisfiability problems for various modal languages, we learn about the complexity of the corresponding unification problems.

The correspondence just noted extends to richer unification formalisms than the rather simple AVMs we have so far considered. In particular, it extends to formalisms that have the ability to encode *re-entrancy*. Re-entrancy is a very influential idea in unification based approaches to grammar, and we need to discuss it, and how it can be dealt with in modal languages.

One of the simplest formalisms for dealing with re-entrancy is to extend AVM notation with 'boxlabels'. Consider the following AVM:

$$\begin{bmatrix} \text{SUBJ} & \boxed{1} \begin{bmatrix} \text{AGR} & \text{foo} \\ \text{PRED} & \text{bar} \end{bmatrix} \\ \text{COMP} & [\text{SUBJ } \boxed{1}] \end{bmatrix}$$

The boxlabels are the $\boxed{1}$ s. What is intended by this notation is explained by the following graphs:



The first graph does *not* satisfy the AVM at its root node. This is because $\boxed{1}$ is a name: it labels a *unique* node. The second graph *does* satisfy the AVM. The crucial difference is that in this graph the SUBJ *re-enters* the graph at the named node. Thus all the conditions demanded by the AVM are satisfied, including the demand that $\boxed{1}$ picks out a unique node.

How can we make modal languages referential in this way? The key idea needed can be traced back to early work by Arthur Prior [24], and Robert Bull [5]: it is to introduce a second sort of atomic symbol constrained to be true at exactly one node. These new symbols 'name' the unique node they are true at. In this paper these symbols are called *nominals*, and they are written as i, j and k and so on.

AVM boxlabels correspond straightforwardly to nominals. Consider once more the following AVM:

$$\left[\begin{array}{cc} \text{SUBJ} & \boxed{1} \left[\begin{array}{cc} \text{AGR} & \text{foo} \\ \text{PRED} & \text{bar} \end{array} \right] \\ \text{COMP} & [\text{SUBJ} \boxed{1}] \end{array} \right]$$

This corresponds to the following wff:

$$\langle \text{SUBJ} \rangle (i \wedge \langle \text{AGR} \rangle \text{foo} \wedge \langle \text{PRED} \rangle \text{bar}) \\ \wedge \langle \text{COMP} \rangle \langle \text{SUBJ} \rangle i$$

Note that the nominal i is doing the same work in the modal wff that $\boxed{1}$ does in the AVM. More generally, the use of nominals permits a transparent linearisation of those AVMs that utilise boxlabels.

Although AVM notation is widely used, it is certainly not the only notation computational linguists use to describe AVSs. Another influential notation arose from the command language of the PATR-II system.¹ In particular, the use of *path equations* for specifying re-entrancy arose from this source.

The following example shows how a user of PATR-II specifies re-entrancy. The user might write:

$$\langle \text{VP HEAD} \rangle = \langle \text{VP VERB HEAD} \rangle.$$

¹PATR-II is an 'implemented grammar formalism', a program which provides a high level interface language geared towards the needs of the linguist, together with a parser. The linguist writes grammars in the interface language and tests them using the parser.

This path equation means that the sequence of transitions encoded by the list of attributes on the left takes one to the same node as the sequence of transitions encoded by the list of attributes on the right. That is, both transition sequences lead to the same node. Note that although this mechanism permits re-entrancy to be specified, it does so in a very different way from the 'boxlabels' approach: no node labelling is involved.

To capture the effect of this in a modal language, we're going to extend the basic language in such a way as to permit 'modal path equations' to be formed. In particular, we'll add a new primitive symbol $\approx$ to allow us to equate strings of modalities. This will permit wffs such as

$$\langle \text{VP} \rangle \langle \text{HEAD} \rangle \approx \langle \text{VP} \rangle \langle \text{VERB} \rangle \langle \text{HEAD} \rangle,$$

to be formed, and we will define the semantics of these new wffs so that they capture the meaning of the PATR-II path equations. Actually, we'll also add a second new primitive symbol, 0. This will act as a name for the null transition, and will enable us to write path equations such as $\langle g \rangle \langle f \rangle \approx 0$. This last wff, for example, will mean that making an R_g transition followed by an R_f transition is the same as making the null transition. That is, the path $R_g R_f$ terminates at its starting point.

It should now be clear that various AV formalisms correspond straightforwardly to propositional modal languages. To conclude this section let's make our discussion of these modal languages more precise. Syntactically, the language L (of signature $\langle \mathcal{L}, \mathcal{A} \rangle$) is a language of propositional modal logic with an $\mathcal{L}$ indexed collection of distinct (existential) modalities and an $\mathcal{A}$ indexed collection of propositional symbols. As primitive Boolean symbols we choose $\neg$ and $\vee$. The wffs of the language are defined by saying that: (a) All propositional symbols p_α are wffs, for all $\alpha \in \mathcal{A}$; (b) If ϕ and ψ are wffs then so are $\neg\phi$, $\phi \vee \psi$, and $\langle l \rangle \phi$, for all $l \in \mathcal{L}$; (c) Nothing else is a wff. We define all the other familiar connectives in the usual way. In particular, we define $[l]\phi$ to be $\neg \langle l \rangle \neg\phi$, for all $l \in \mathcal{L}$ and all wffs ϕ . The *length* of a wff ϕ ($|\phi|$) is the number of symbols it contains. The *degree* of a formula is the number of (primitive) connectives it contains.

To interpret L we use Kripke models M of signature $\langle \mathcal{L}, \mathcal{A} \rangle$. Such a Kripke model is a triple $\langle W, \{R_l\}_{l \in \mathcal{L}}, V \rangle$, where W is a non-empty set (the set of nodes); each R_l is a binary relation on W that is also a partial function, that is, for every node w there exists at most one w' such that $w R_l w'$; and V (the valuation) is a function which assigns each propositional symbol p_α a subset of W . We interpret wffs of L on models M in the familiar fashion:

$$\begin{array}{lll} M \models p_\alpha[w] & \text{iff} & w \in V(p_\alpha) \\ M \models \neg\phi[w] & \text{iff} & M \not\models \phi[w] \\ M \models \phi \vee \psi[w] & \text{iff} & M \models \phi[w] \text{ or } M \models \psi[w] \\ M \models \langle l \rangle \phi[w] & \text{iff} & \exists w' (w R_l w' \ \& \ M \models \phi[w']) \end{array}$$

To sum up, the language L corresponds to the 'core' AVM notation used by computational linguists. Its models are just AVSs, and the way L formulas are evaluated in a model is just the way AVMs are checked against AVSs.

L lacks any mechanism for enforcing re-entrancy. This lack is made good its extensions, L^N and L^{KR} . The language L^N is L augmented by a collection of distinct new propositional symbols called *nominals*. These symbols are typically written as i, j, k etc. and can be freely combined with the other symbols in the usual fashion to make wffs. To interpret nominals we insist that any valuation must assign a singleton subset to each nominal. That is, an L^N model is just an L model whose valuation has been extended to assign singletons to nominals. Because each nominal is thus true at exactly one node in any model, it acts as a 'label' identifying that node. L^N corresponds to AVMs augmented with 'boxlabels' for indicating re-entrancy.

The language L^{KR} is L augmented by two new symbols, 0 and $\approx$. Using these symbols we make *path equations*: given any nonempty sequence of L modalities $\langle F \rangle$ and $\langle G \rangle$ then $0 \approx 0$, $0 \approx \langle G \rangle$, $\langle F \rangle \approx 0$ and $\langle F \rangle \approx \langle G \rangle$ are path equations. Path equations are wffs and can be combined with other wffs in the usual way to make more complex wffs. L^{KR} models are just L models, and we interpret the path equations as follows:

$$\begin{aligned} M &\models 0 \approx 0[w] \text{ for all } w \in W \\ M &\models 0 \approx \langle f \rangle \dots \langle g \rangle[w] \text{ iff } M \models \langle f \rangle \dots \langle g \rangle \approx 0[w] \text{ iff } wR_f \dots R_g w \\ M &\models \langle f \rangle \dots \langle g \rangle \approx \langle h \rangle \dots \langle l \rangle[w] \text{ iff } \exists w' (wR_f \dots R_g w' \ \& \ wR_h \dots R_l w') \end{aligned}$$

L^{KR} models the path equation mechanism of PATR-II. The negation free fragment of this language was first defined and studied by Kaspar and Rounds [15][27].

For further discussion of L , L^N and L^{KR} , including axiomatisations of a number of Attribute Value logics and expressivity results, the reader is referred to [4]. Here we're going to treat the issue of most relevance to computational linguistics: the complexity of their satisfiability problems.

2 Complexity results for L , L^N and L^{KR}

In this section we show that the satisfiability problems for L , L^N , and L^{KR} are all NP complete. The fundamental result is that for L , for it turns out that the methods used for this language generalise straightforwardly to its two extensions.

The key to the NP completeness result for L is to show that given a formula ϕ which is satisfiable in some model M , we can always find a suitably small model S^σ which also satisfies ϕ . The work that follows is devoted to defining S^σ , showing that it satisfies ϕ , and determining its size. Once this has been done the NP completeness result is immediate.

So, let M be any model, let v be a node in M , and let ϕ be any wff. The first step towards defining S^σ is to define a partial function σ which picks out precisely those nodes of M which are actually used when evaluating ϕ at v . More precisely, we define a partial function σ from occurrences of subformulas ψ of ϕ into the nodes of M as follows:

1. Define $\sigma(\phi) = v$.
2. For each occurrence of a subformula of the form $\neg\psi$ such that $\sigma(\neg\psi)$ is defined, $\sigma(\psi) = \sigma(\neg\psi)$; otherwise $\sigma(\psi)$ is undefined.
3. For each occurrence of a subformula of the form $\psi \vee \theta$ such that $\sigma(\psi \vee \theta)$ is defined, $\sigma(\psi) = \sigma(\theta) = \sigma(\psi \vee \theta)$; otherwise $\sigma(\psi)$ and $\sigma(\theta)$ are both undefined.
4. For each occurrence of a subformula of the form $\langle l \rangle \psi$ such that $\sigma(\langle l \rangle \psi)$ is defined, $\sigma(\psi) = w'$ where $wR_l w'$, if such a $w' \in W$ exists, and is undefined if there is no such w' ; otherwise $\sigma(\psi)$ is undefined.

Note that once M , v and ϕ have been specified, σ is uniquely determined by the above definition. The key thing to observe is that there is no 'free choice' involved in the selection of w' in the clause for $\sigma(\langle l \rangle \psi)$. *Because we are working on partial functional frames, if there is such an w' , it is unique.* This fact plays a key role in ensuring that the new model S^σ has the properties we require of it.

Let's now define S^σ . The nodes of S^σ will be the image of σ , and the relations and valuation of S^σ will be the restrictions of the relations and valuation of M to this image set. More precisely, we define S to be the image of σ ; for all $l \in \mathcal{L}$ we define R_l^* to be $R_l \cap (S \times S)$;

for all atoms p we define $V^s(p)$ to be $V(p) \cap S$; and finally we define the model S^σ to be $\langle S, \{R_i^s\}_{i \in \mathcal{L}}, V^s \rangle$. Now, the first useful property of S^σ is the following:

Lemma 2.1 (Equivalence Lemma) *For all occurrences of subformulas ψ of ϕ such that $\sigma(\psi)$ is defined:*

$$M \models \psi[\sigma(\psi)] \text{ iff } S^\sigma \models \psi[\sigma(\psi)].$$

Proof:

By induction on the degree of ϕ . For the base case, suppose that p is any occurrence of a propositional variable in ϕ such that $\sigma(p)$ is defined. (Note that this means that $\sigma(p) \in S$.) Then:

$$\begin{aligned} M \models p[\sigma(p)] & \text{ iff } \sigma(p) \in V(p) \\ & \text{ iff } \sigma(p) \in V(p) \cap S \\ & \text{ iff } \sigma(p) \in V^s(p) \\ & \text{ iff } S^\sigma \models p[\sigma(p)]. \end{aligned}$$

This establishes the base step. So suppose that for all wffs ψ such that $\sigma(\psi)$ is defined, and the degree of ψ less than k , the result holds.

The inductive clauses for the Boolean connectives are straightforward; we treat the case for negation. Suppose $\neg\psi$ has degree k and that $\sigma(\neg\psi)$ is defined. Then:

$$\begin{aligned} M \models \neg\psi[\sigma(\neg\psi)] & \text{ iff } M \not\models \psi[\sigma(\neg\psi)] \\ & \text{ iff } M \not\models \psi[\sigma(\psi)] & (\text{As } \sigma(\psi) = \sigma(\neg\psi).) \\ & \text{ iff } S^\sigma \not\models \psi[\sigma(\psi)] & (\text{Inductive hypothesis.}) \\ & \text{ iff } S^\sigma \models \neg\psi[\sigma(\psi)] \\ & \text{ iff } S^\sigma \models \neg\psi[\sigma(\neg\psi)] & (\text{As } \sigma(\psi) = \sigma(\neg\psi).) \end{aligned}$$

Now for the modalities. Suppose $\langle l \rangle\psi$ has degree k and that $\sigma(\langle l \rangle\psi)$ is defined. Further suppose that $M \models \langle l \rangle\psi[\sigma(\langle l \rangle\psi)]$. This can happen iff there is a node w in M such that $\sigma(\langle l \rangle\psi)R_l w$ and $M \models \psi[w]$. But since $\sigma(\langle l \rangle\psi)$ is defined, and since such a w exists, $\sigma(\psi)$ is also defined and equals w . Thus we know two facts: first that $\sigma(\langle l \rangle\psi)R_l \sigma(\psi)$, and second that $M \models \psi[\sigma(\psi)]$. But obviously both $\sigma(\langle l \rangle\psi)$ and $\sigma(\psi)$ are in S , hence as R_l^s is the restriction of R_l to S , it follows from the first fact that $\sigma(\langle l \rangle\psi)R_l^s \sigma(\psi)$. By the inductive hypothesis it follows from the second fact that $S^\sigma \models \psi[\sigma(\psi)]$. Taken together, we have that $S^\sigma \models \psi[\sigma(\langle l \rangle\psi)]$, as required.

Conversely suppose that $S^\sigma \models \langle l \rangle\psi[\sigma(\langle l \rangle\psi)]$. This can happen iff there is a node s in S^σ such that $\sigma(\langle l \rangle\psi)R_l^s s$ and $S^\sigma \models \psi[s]$. Now as R_l^s is the restriction of R_l to S , we have that $\sigma(\langle l \rangle\psi)R_l s$. Thus (by the fourth clause in the definition of σ), $\sigma(\psi)$ exists and is equal to s . This tells us that $\sigma(\langle l \rangle\psi)R_l \sigma(\psi)$. It also tells us that $S^\sigma \models \psi[\sigma(\psi)]$, which by the inductive hypothesis yields $M \models \psi[\sigma(\psi)]$. Taken together, this means that $M \models \psi[\sigma(\langle l \rangle\psi)]$, as required. Hence the required result follows by induction. $\square$

It is an immediate corollary of this result that if $M \models \phi[v]$, then $S^\sigma \models \phi[v]$. For suppose $M \models \phi[v]$. Then by our definition of σ , $\sigma(\phi)$ exists and equals v . Thus we have that $M \models \phi[\sigma(\phi)]$, and thus by the previous lemma $S^\sigma \models \phi[\sigma(\phi)]$, that is, $S^\sigma \models \phi[v]$.

Thus our new model S^σ satisfies ϕ . The next property we wish to establish about S^σ is that it is 'suitably small', which in the present context means that the number of nodes it contains is polynomial in the length of ϕ . This is the case: indeed a moments reflection reveals that S can contain at most one more node than there are modalities in ϕ . For if ϕ contains no modalities then S^σ contains only the node v . On the other hand, if there are

modalities in ϕ , each modality contributes at most one additional node to S^σ . In short, the fact that we are working on partial functional frames gives us a very sharp upper bound on the size of S^σ . The following (rather tedious) lemma merely makes this observation precise.

Some preliminary definitions and remarks will be helpful. Suppose $\langle l \rangle \theta$ is an occurrence of a formula and that ψ is some occurrence of a subformula of $\langle l \rangle \theta$. Then we say that ψ lies in the scope of $\langle l \rangle$. Moreover, if for all occurrences of modalities $\langle l' \rangle$ in θ we have that ψ does not lie in the scope of $\langle l' \rangle$, then we say that $\langle l \rangle$ is the *least modality containing* ψ . Finally note the following useful fact about σ . Suppose that ψ and ψ' are occurrences of subformulas of ϕ such that ψ occurs in ψ' . Further suppose that $\sigma(\psi)$ is defined. Then if for all occurrences of modalities $\langle l' \rangle$ in ψ' we have that ψ does not lie in the scope of $\langle l' \rangle$, then $\sigma(\psi')$ is defined and equals $\sigma(\psi)$. This follows immediately from the definition of σ .

Lemma 2.2 (Size Lemma) *Let M be a model, let v be a node in M , let ϕ be a wff, and let S^σ be constructed from these components as described above. Then $\text{card}(S) \leq n + 1$, where n is the number of modalities occurring in ϕ .*

Proof:

Recall that S is just the image of σ . It follows from the definition of σ that if ϕ contains no modalities then the image of σ is just $\{v\}$, and we have the required result.

So let ϕ be any wff containing at least k occurrences of modalities, where $k > 0$. Now either $S = \{v\}$, in which case there is nothing more to show, or $S \setminus \{v\}$ is non-empty. So suppose that the latter is the case. We will show that there is an injection τ from $S \setminus \{v\}$ to the occurrences of subformulas of ϕ of the form $\langle l \rangle \theta$. The existence of such an injection shows that S^σ contains at most k nodes distinct from v , which in turn shows that S contains at most $k + 1$ nodes.

Let $s \in S \setminus \{v\}$. As S is the image of σ we have that $s = \sigma(\psi)$ for some occurrence of a subformula ψ of ϕ . We define $\tau(s)$ to be the occurrence of a subformula of ϕ that has the form $\langle l \rangle \theta$, and such that $\langle l \rangle$ is the least modality containing ψ .

Clearly τ is well defined. For suppose there is no occurrence of a subformula of ϕ of the form $\langle l \rangle \theta$ such that ψ lies in the scope of $\langle l \rangle$. Then making use of the useful fact about σ mentioned above, we see that $\sigma(\psi) = \sigma(\phi)$. But as $\sigma(\phi) = v$ this contradicts our assumption that $s = \sigma(\psi) \in S \setminus \{v\}$. So ψ must always be in the scope of at least one modality, so there is a least modality containing ψ , and the definition of τ makes sense.

To show that τ is injective we argue by *reductio ad absurdum*. Suppose $s, s' \in S \setminus \{v\}$, $s \neq s'$, and $\tau(s) = \tau(s')$. Let $\langle l \rangle \theta$ be the mutual image point of s and s' under τ . We claim that $\sigma(\langle l \rangle \theta) R_i^s s$. We see this as follows. As $s \in S \setminus \{v\}$ there exists a subformula ψ such that $s = \sigma(\psi)$. Hence by the definition of τ , ψ is a subformula of θ that does not lie within the scope of any modality in θ . Thus by the useful fact mentioned above, $\sigma(\theta)$ is defined and equals $\sigma(\psi)$. That is, $\sigma(\theta) = s$. But as $\sigma(\theta)$ is defined then so is $\sigma(\langle l \rangle \theta)$, and by the definition of R_i^s , $\sigma(\langle l \rangle \theta) R_i^s \sigma(\theta)$. This means that $\sigma(\langle l \rangle \theta) R_i^s s$ as claimed. In a similar fashion we see that $\sigma(\langle l \rangle \theta) R_i^{s'} s'$. But now we have a contradiction: as $s \neq s'$ and as $\sigma(\langle l \rangle \theta)$ is R_i^s related to both s and s' then R_i^s is not a partial function. But this is impossible as R_i^s is the restriction of the partial function R_i to S . So τ is injective and we are through. $\square$

The equivalence lemma and the size lemma lead us directly to the main result:

Theorem 2.1 *The satisfiability problem for L is NP complete.*

Proof:

That this satisfiability problem is NP hard is clear, as it contains the satisfiability problem for propositional calculus as a special case. That the problem is in NP follows

directly from the fact that any satisfiable L wff ϕ can be satisfied in a model containing at most $n + 1$ nodes, where n is the number of occurrences of modalities in ϕ . Thus, given ϕ we can non-deterministically choose a suitable model of at most this size, and evaluate ϕ in this model in polynomial time. $\square$

Let's turn to the complexity of the satisfiability problem for the language L^N . Recall that this language is L augmented by distinct new set of atomic symbols called nominals which are constrained to be true at exactly one node in any model. It is easy to use the machinery developed above to prove that the satisfiability problem for L^N is also NP complete, in fact there is almost nothing new to be done. Given a L^N model M , a node v in M , and an L^N wff ϕ we define S^σ exactly as described above. Both the equivalence lemma and the size lemma hold, thus we are almost through. There is only one snag: S^σ is not guaranteed to be an L^N model as some nominals may be not denote any node at all. But this problem is more apparent than real. By adjoining a brand new node (say $*$) to S^σ , and insisting that all 'unassigned nominals' denote $*$ we convert S^σ into an L^N model $S^{\sigma*}$. Of course to maintain the truth of the equivalence lemma we have to be careful where we place $*$, but there are two obvious 'safe' choices. The simplest choice is to insist that $*$ is unrelated (by any of the relations) to any of the points in S . The second, slightly more elegant, is to insist that $*$ is related to v by some relation, but that none of the points in S is related to $*$. Choosing this second option means that $*$ point generates $S^{\sigma*}$. Either way it is clear that the addition of $*$ is harmless: we still have that $S^{\sigma*} \models \phi[v]$. And $S^{\sigma*}$ is 'suitably small', having at most $n + 2$ nodes where n is the number of modality occurrences in ϕ . Thus by precisely the same argument as for L we have:

Theorem 2.2 *The satisfiability problem for L^N is NP complete.* $\square$

Finally we turn to L^{KR} . The satisfiability problem for this language is also NP complete, but how are we to show this? Our definition of σ says nothing about occurrences of path equations, and as the following example shows, we definitely need to deal with them in some fashion.

Suppose we're working with some model M , suppose that in this model $\langle f \rangle \approx \langle g \rangle$ is true at node v , and suppose $\sigma(\langle f \rangle \approx \langle g \rangle) = v$. Note that because $\langle f \rangle \approx \langle g \rangle$ is true in M at v , there is a node v' in M such that $vR_f v'$ and $vR_g v'$. Let us suppose that $v \neq v'$. Now we want to generate all the points needed to make an equivalent small model M^σ by iteratively applying σ to subformulas of $\langle f \rangle \approx \langle g \rangle$. In particular, we need to ensure that v' is one of the points we generate. But things immediately go wrong. We know that $\sigma(\langle f \rangle \approx \langle g \rangle)$, but we cannot apply σ to any substructure of this path equation: the path equation has no proper subformulas, and thus σ has no means of making use of the two modalities it contains. This means that v' is not in the image of σ , and thus the small model M^σ we create will *falsify* $\langle f \rangle \approx \langle g \rangle$ at v . This is definitely not what we want.

But there's a simple way round the problem: we rewrite the input formula $\langle f \rangle \approx \langle g \rangle$ into a logically equivalent form that makes explicit the existential demands of the path equations. Suppose we rewrite $\langle f \rangle \approx \langle g \rangle$ as $\langle f \rangle \approx \langle g \rangle \wedge \langle f \rangle \top \wedge \langle g \rangle \top$. Although this formula is logically equivalent to the original, there is a crucial syntactic difference: the two new conjuncts make the modalities $\langle f \rangle$ and $\langle g \rangle$ available to iterative applications of σ . Consider what happens when we iteratively apply σ to this new formula. As σ commutes over $\wedge$, we must calculate $\sigma(\langle f \rangle \approx \langle g \rangle)$, $\sigma(\langle f \rangle \top)$ and $\sigma(\langle g \rangle \top)$. Now, as before, we can't do anything further with $\sigma(\langle f \rangle \approx \langle g \rangle)$, but we *can* do something further with both $\sigma(\langle f \rangle \top)$ and $\sigma(\langle g \rangle \top)$, as σ is defined for such expressions. In fact evaluating these formulas will produce the point v' that we need to build an equivalent small model.

Let's make this precise. Any path equation $\langle F \rangle \approx \langle G \rangle$ is logically equivalent to $\langle F \rangle \approx \langle G \rangle \wedge \langle F \rangle \top \wedge \langle G \rangle \top$. For any path equation $\langle F \rangle \approx \langle G \rangle$ we'll call $\langle F \rangle \approx \langle G \rangle \wedge \langle F \rangle \top \wedge \langle G \rangle \top$ its *explicit form*. Given an L^{KR} wff ϕ which we seek to satisfy, we'll first form a new L^{KR} wff ϕ^* by simultaneously substituting, for each occurrence of a path equation in ϕ , its explicit form. Note that ϕ^* is logically equivalent to ϕ , and that the length of ϕ^* is linear in the length of ϕ . The effect of this rewriting of ϕ means that our existing definition of σ suffices to produce all the points needed for the small model: precisely as illustrated in the above example, when we iteratively apply σ , the occurrences of the new subformulas of the form $\langle F \rangle \top$ and $\langle G \rangle \top$ ensure that all the needed evaluation points are included. Thus we can make S^σ as before and both the equivalence lemma and the size lemma are exactly as for L . So, by exactly the same argument we have that:

Theorem 2.3 *The satisfiability problem for L^{KR} is NP complete.* □

Let's summarise our results so far. The satisfiability problem for the core AV language L is NP complete. Adding either of two re-entrancy — nominals or the Kaspar Rounds path equality — does not increase the complexity: satisfiability remains NP complete. There is a result from the literature worth noting here: Kaspar and Rounds show that the negation free fragment of L^{KR} is NP complete. Our result for L^{KR} thus shows that the situation doesn't get worse if full Boolean expressivity is allowed.

What can be said at a more general level about these results? For a start, they're somewhat unexpected: with the exception of S5 most familiar modal logics are PSPACE complete. To put matters another way, usually adding modalities to a language of propositional logic makes matters worse, but here it hasn't. The reason, of course, is due to the fundamental constraint on our models, namely that all the relations be partial functional. It's this requirement which enabled us to build small models and thus kept the complexity to that of propositional logic. It's worth adding that this constraint seems to be peculiar to the representational formalisms used in computational linguistics. Various representation formalisms used in AI, such as KL-ONE, can be viewed from a modal perspective, and as has been recently observed, terminological logics are also modal logics [29]. But there is one important difference: the modal logics inspired by AI don't usually obey the partial functionality constraint. Typically they're nothing but multimodal versions of K , the modal logic which puts no constraints on accessibility relations. As is well known, the satisfiability problem for this logic is PSPACE complete [18].

Are there any plausible restrictions we can make on these languages to lower the complexity of the satisfiability problems? The one obvious thing is to (in effect) return to the early days of computational linguistics and only allow conjunction as a Boolean connective. As is well known this reduces the problem to one in P — but such a restriction would probably be most unwelcome to computational linguists. Beyond this matters are unclear. There is one intriguing possibility which (alas) leads nowhere. Many Attribute Value formalism use a finite number of atomic symbols. Given that the number of propositional symbols is some fixed finite number, might this not permit us to evade the NP hardness result? For propositional logic and for S5, such a restriction lowers the complexity of the satisfiability problem to P . However, the satisfiability problem for L (and thus for L^N and L^{KR}) remains NP-complete, even if we use only one propositional variable, and one modal operator. For consider the following set of L formulas: $\{p, \langle f \rangle p, \langle f \rangle \langle f \rangle p, \dots, \langle f \rangle^{n-1} p\}$. The values of these formulas are all independent, that is, for any sequence of truth values $b_0, \dots, b_{n-1}$, there exists a model such that $M \models \langle f \rangle^i p$ iff b_i is true. Now define function f from propositional formulas to L -formulas as follows:

$$f(\phi(p_1, \dots, p_n)) := \phi(p, \langle f \rangle p, \langle f \rangle \langle f \rangle p, \dots, \langle f \rangle^{n-1} p).$$

Obviously, f is polynomial time computable, and ϕ is satisfiable iff $f(\phi)$ is L satisfiable. It follows that the satisfiability problem for L with one propositional variable and one modality is NP hard, and hence, by theorem 2.1, NP complete. Thus there are no obvious routes to tractability save the most unpalatable: sacrificing expressive power.

3 Adding the universal modality

In this section we are going to examine the complexity of the satisfiability problems for two stronger modal languages, $L^{KR\Box}$ and $L^\Box$. These languages are, respectively, L^{KR} and L augmented by the *universal modality* [9]. The universal modality is a modal operator written as $\Box$ which has the following semantics: for all models M , all nodes w , and all wffs ϕ

$$M \models \phi[w] \text{ iff } M \models \phi[w'], \text{ for all nodes } w' \text{ in } M.$$

That is, $\Box\phi$ holds iff ϕ is true at all nodes.

But why should we be interested in $L^{KR\Box}$ and $L^\Box$? The answer is as follows. Underlying much work in Attribute Value grammar is an idea that can loosely be described as 'grammar equals feature logic'. Somewhat more precisely, the use of the apparatus of unification formalisms is attractive to many linguists because it enables them to view grammars of natural languages as theories in some sort of calculus of attributes and values. According to such a view, linguistic structure can be adequately modelled by Attribute Value Structures (possibly augmented by the notion of phrase structure), and the linguists' business is to state general constraints about which AVSs are admissible. Such views are discernible in some of the earliest work in attribute value grammar, namely Lexical Function Grammar (LFG) [13]. Generalised Phrase Structure Grammar (GPSG) [7], explicitly espouses such views, and its work on feature co-occurrence restrictions remains one of the best examples of the approach in action. More recently, Head Driven Phrase Structure Grammar (HPSG) [21], has taken this approach even further. Whereas in both GPSG and LFG the idea of unification was only one component (albeit an important one) of the systems, in HPSG the unificational apparatus completely dominates.

It is such developments that motivate the work of the present section. As we have, seen the most common unificational formalisms are nothing but modal languages. However as they stand these languages aren't strong enough to express generalisations, and indeed as the 'grammar equals feature logic' equation has become more widely accepted, work in Attribute Value grammar has tended to abandon the simple languages we have considered so far in favour of increasingly powerful formalisms. One particularly natural choice of language has been advocated by Smolka [31] [32]. He has explored the consequences of working in the first order language of attribute value structures, that is, the first order language (with equality) that contains a binary relation symbol F_i for each R_i , and a unary relation symbol P_α for each Q_α . This language, of course, is just the first order correspondence language (in the sense of modal correspondence theory [2]) of L , L^N and L^{KR} . Now this is clearly a crucial language to explore, but it is by no means the only available option. Another option is to strengthen the modal languages to allow them to express generalisations about AV structures. Adding the universal modality is precisely such a strengthening, and this section explores the complexity of such a move.

Let's consider matters more concretely. Suppose we strengthen L^{KR} by adding the universal modality. That is, we add a new formation rule to those of L^{KR} that states that if ϕ is a wff then so is $\Box\phi$, and we interpret such universal wffs using the definition given above. What can we express in this richer language?

Let's first consider GPSG feature co-occurrence restrictions, for example

$$[\text{VFORM FIN}] \Rightarrow [-N, +V].$$

This states that if a node has the value FIN for the attribute VFORM, then that node has the properties of being +N and -V. In other words, only a verb can have tense.

The important thing about this constraint is its generality. It's not something which happens to hold of this or that piece of linguistic structure, it's a pervasive fact of English. Any AVS which doesn't satisfy this generalisation can't represent an English sentence. We can express this generalisation in $L^\square$ as follows:

$$\square(\langle \text{VFORM} \rangle \text{fin} \rightarrow -n \wedge +v).$$

Let's now consider an example which makes use of $\approx$. The example is the representation of the Head Feature Convention of GPSG. This principle (which is also present in a more elaborate form in HPSG) states that for any phrasal constituent, the value of its head attribute is shared with the value of the head attribute of its head daughter. For a discussion of what this terminology all means, and why it's linguistically useful the reader is referred to [7]; here we'll be content to indicate how the constraint can be expressed in $L^{KR\square}$:

$$\square(\text{phrasal} \rightarrow \langle \text{HEAD} \rangle \approx \langle \text{HEAD-DTR} \rangle \langle \text{HEAD} \rangle).$$

Further experimentation will convince the reader that $L^{KR\square}$ is a rich language capable of expressing many interesting linguistic constraints. There are omissions: notably $L^{KR\square}$ doesn't possess any capacity for dealing with the so-called set values of HPSG.² But we won't make further extensions here, for, from the point of view of a philosophy that insists that only decidable theories should be capable of formulation in a linguistic framework, $L^{KR\square}$ has already gone too far. As we shall now show, the satisfiability problem for $L^{KR\square}$ is undecidable.

To prove the undecidability result it suffices to give a reduction from a Π_1^0 hard problem to $L^{KR\square}$ satisfiability. As is shown in [10], tiling problems provide a particularly elegant method of proving lower bounds for modal logics. A tile T is a 1×1 square fixed in orientation with coloured edges: $\text{right}(T)$, $\text{left}(T)$, $\text{up}(T)$, and $\text{down}(T)$ taken from some denumerable set C . A tiling problem takes the following form: given a finite set $\mathcal{T}$ of tiles, can we cover a certain part of $\mathbb{Z} \times \mathbb{Z}$ with tiles from $\mathcal{T}$ in such a way that adjacent tiles have the same colour on the common edge? There exist complete tiling problems for many complexity classes, and the proof that follows makes use of a certain Π_1^0 complete tiling problem.

Theorem 3.1 *The satisfiability problem for $L^{KR\square}$ is Π_1^0 hard.*

Proof:

As shown in [3] [26], the following problem is Π_1^0 complete:

$N \times N$ tiling: Given a finite set $\mathcal{T}$ of tiles, can $\mathcal{T}$ tile $N \times N$?

That is, does there exist a function t from $N \times N$ to $\mathcal{T}$ such that $\text{right}(t(n, m)) = \text{left}(t(n + 1, m))$ and $\text{up}(t(n, m)) = \text{down}(t(n, m + 1))$? Let $\mathcal{T} = \{T_1, \dots, T_k\}$ be a set of tiles. We construct a formula ϕ such that:

$$\mathcal{T} \text{ tiles } N \times N \text{ iff } \phi \text{ is satisfiable.}$$

² Actually, set values have been dealt with in modal languages, and in two different ways. One approach, due to Reape [25], is to add n place modalities to the language. The other, due to Moss [19], is to add a special unary modal operator that deals with the membership relation.

First of all we will ensure that, if ϕ is satisfiable in $\mathbf{M}$, then $\mathbf{M}$ contains a gridlike structure, where the nodes of $\mathbf{M}$ play the role of points in a grid, R_r is the right successor relation, and R_u is the upward successor relation. Define:

$$\phi_{grid} = \Box(\langle \tau \rangle \langle u \rangle \approx \langle u \rangle \langle \tau \rangle).$$

Suppose $\mathbf{M} \models \phi_{grid}[w_0]$. Then there exists a function f from $\mathbb{N} \times \mathbb{N}$ to W such that: $f(0,0) = w_0$, $f(n,m)R_rf(n+1,m)$, and $f(n,m)R_uf(n,m+1)$.

Next we must tile the model. To do this we use propositional variables $t_1, \dots, t_k$, such that t_i is true at some node w , iff tile T_i is placed at w . To force a proper tiling, we need to satisfy the following three requirements:

1. There is exactly one tile placed at each node.

$$\phi_1 = \Box\left(\bigvee_{i=1}^k t_i \wedge \bigwedge_{1 \leq i < j \leq k} \neg(t_i \wedge t_j)\right)$$

2. If T is the tile at w , and T' the tile at the right successor of w , then $right(T) = left(T')$.

$$\phi_2 = \Box\left(\bigvee_{right(T_i) \neq left(T_j)} \neg(t_i \wedge \langle \tau \rangle t_j)\right)$$

3. If T is the tile at w , and T' the tile at the up successor of w , then $up(T) = down(T')$.

$$\phi_3 = \Box\left(\bigvee_{up(T_i) \neq down(T_j)} \neg(t_i \wedge \langle u \rangle t_j)\right)$$

Putting this all together, we define ϕ to be $\phi_{grid} \wedge \phi_1 \wedge \phi_2 \wedge \phi_3$. We will prove that $\mathcal{T}$ tiles $\mathbb{N} \times \mathbb{N}$ iff ϕ is satisfiable.

Suppose $t : \mathbb{N} \times \mathbb{N} \rightarrow \mathcal{T}$ is a tiling of $\mathbb{N} \times \mathbb{N}$. We construct the satisfying model for ϕ as follows: $\mathbf{M} = \langle W, R_r, R_u, V \rangle$ such that:

$$\begin{aligned} W &= \{w_{n,m} : n, m \in \mathbb{N}\} \\ R_r &= \{\langle w_{n,m}, w_{n,m+1} \rangle : n, m \in \mathbb{N}\} \\ R_u &= \{\langle w_{n,m}, w_{n+1,m} \rangle : n, m \in \mathbb{N}\} \\ V(t_i) &= \{w_{n,m} : n, m \in \mathbb{N} \text{ and } t(n,m) = T_i\} \end{aligned}$$

Clearly, ϕ holds at any node w of $\mathbf{M}$. To see that the converse also holds, suppose that $\mathbf{M} \models \phi[w_0]$. Let f from $\mathbb{N} \times \mathbb{N}$ to W be such that $f(0,0) = w_0$, $f(n,m)R_rf(n+1,m)$ and $f(n,m)R_uf(n,m+1)$. Define the tiling $t : \mathbb{N} \times \mathbb{N} \rightarrow \mathcal{T}$ by $t(n,m) = T_i$ iff $\mathbf{M} \models t_i[f(n,m)]$. Note that t is well-defined and total by ϕ_1 . Furthermore, if $t(n,m) = T_i$ and $t(n+1,m) = T_j$, then $f(n,m)R_rf(n+1,m)$, $\mathbf{M} \models t_i[f(n,m)]$, and $\mathbf{M} \models t_j[f(n+1,m)]$. Since $\mathbf{M}$ satisfies ϕ_2 , we can conclude that $right(T_i) = left(T_j)$. Similarly, if $t(n,m) = T_i$ and $t(n,m+1) = T_j$, then ϕ_3 ensures that $up(T_i) = down(T_j)$. Thus, $\mathcal{T}$ tiles $\mathbb{N} \times \mathbb{N}$. $\square$

What are we to make of this undecidability result? The key technical point is that it is genuinely due to the interaction between $\Box$ and $\approx$. As we shall now show, the satisfiability problem for $L^\square$ is decidable.

Theorem 3.2 *If ϕ is a satisfiable $L^\square$ formula, then ϕ is satisfiable in a model with at most $2^{2|\phi|}$ nodes.*

Proof:

Suppose $M = \langle W, \{R_l\}_{l \in \mathcal{L}}, V \rangle$, and $M \models \phi[w_0]$. Let $Cl(\phi)$ be the smallest set that contains ϕ , and is closed under subformulas and single negations. Define an equivalence relation $\sim$ on W as follows:

$$w \sim w' \text{ iff } \forall \psi \in Cl(\phi) (M \models \psi[w] \Leftrightarrow M \models \psi[w'])$$

Let $W^F \subseteq W$ be such that W^F contains exactly one element from each equivalence class. Let V^F be the restriction of V to W^F , and define R_l^F as follows:

$$w R_l^F w' \text{ iff } \exists w'' (w R_l w'' \wedge w' \sim w'')$$

Let $M^F = \langle W^F, \{R_l^F\}_{l \in \mathcal{L}}, V^F \rangle$. Then M^F is a filtration of M through $Cl(\phi)$ in the sense of Hughes and Cresswell [12]. Therefore, $\forall w \in W^F \forall \psi \in Cl(\phi) (M \models \psi[w] \Leftrightarrow M^F \models \psi[w])$. It follows immediately that M^F satisfies ϕ . Since the size of $Cl(\phi)$ is at most $2|\phi|$, the size of W^F is bounded by $2^{2|\phi|}$. Furthermore, M^F is an $L^\square$ model, since the definition of R_l^F ensures that R_l^F is a partial function for any modality l . $\square$

From theorem 3.2, it follows immediately that the satisfiability problem for $L^\square$ is decidable in nondeterministic exponential time. In fact we can improve this result. Using methods similar to [23] and [11] we sketch a construction of a deterministic exponential time algorithm for $L^\square$ satisfiability.

Theorem 3.3 *The satisfiability problem for $L^\square$ is decidable in EXPTIME.*

Proof:

Let S_1 consist of all subsets Γ of $Cl(\phi)$ that are maximally propositionally consistent, and are closed under reflexivity of $\square$, i.e. if $\square\psi \in \Gamma$ then ψ is also in Γ . We will construct a sequence of sets $S_1 \supset S_2 \supset S_3 \supset \dots$ such that, if ϕ is satisfiable, then ϕ is satisfiable in a model M where for every node w : $\{\psi \in Cl(\psi) : M \models \psi[w]\} \in S_i$.

Suppose we have defined S_i . First of all, note that two sets of S_i can only occur in the same model if they are equivalent with respect to $\square$ -formulas. Define $\Gamma \sim_\square \Gamma'$ iff $\forall \square\psi (\square\psi \in \Gamma \Leftrightarrow \square\psi \in \Gamma')$.

Call a set Γ inconsistent iff one of the following situations occurs:

1. $\neg\square\psi \in \Gamma$, but for all $\Gamma' \sim_\square \Gamma$: $\psi \in \Gamma'$
2. For some modality l , there is no $\Gamma' \in S_i$ such that $\Gamma \sim_\square \Gamma'$ and $\forall (l)\psi \in Cl(\phi) ((l)\psi \in \Gamma \Leftrightarrow \psi \in \Gamma')$

If there are inconsistent sets in S_i , then we let S_{i+1} consist of all sets of S_i that are not inconsistent. Otherwise, ϕ is satisfiable iff $\phi \in \Gamma$ for some set $\Gamma \in S_i$.

Since S_1 is of exponential size, and S_{i+1} is strictly included in S_i , the algorithm terminates after at most exponentially many cycles. Determining which sets in S_i are inconsistent takes polynomial time in the length of the representation of S_i . Thus, the entire algorithm takes at most deterministic exponential time. $\square$

However as the next result shows, there is a clear sense in which this result cannot be improved.

Theorem 3.4 *The satisfiability problem for $L^\square$ is EXPTIME complete.*

Proof:

The upper bound follows from theorem 3.3. To prove the corresponding lower bound, it suffices to give a polynomial time computable reduction from an EXPTIME hard set to $L^\square$ -satisfiability. Let $PDL(\vdash, *)$ be the bimodal propositional language with modalities $\langle \vdash \rangle$ and $\langle \vdash^* \rangle$. We interpret wffs of $PDL(\vdash, *)$ on Kripke models $M = \langle W, R_\vdash, V \rangle$, where $R_\vdash$ is an arbitrary relation on $W \times W$, in the usual way, the key clause being:

$$M \models \langle \vdash^* \rangle \phi[w] \text{ iff } \exists w' (w R_\vdash^* w' \ \& \ M \models \phi[w']).$$

where $R_\vdash^*$ denotes the reflexive, transitive closure of $R_\vdash$. In [6], it is proven that the satisfiability problem for $PDL(\vdash, *)$ is EXPTIME-hard. In fact, from careful inspection of this proof, we can conclude that even the following set is EXPTIME-hard: Let C consist of all $PDL(\vdash, *)$ formulas ϕ such that: $\phi = \phi_1 \wedge [\vdash^*] \phi_2$, and

1. ϕ_1, ϕ_2 are $*$ -less and have modal depth ≤ 1 ,
2. ϕ is satisfiable in a model where every node has at most two successors.

Define the reduction f from C to $L^\square$ -satisfiability as follows:

1. If ϕ is *not* of the form $\phi_1 \wedge [\vdash^*] \phi_2$, where ϕ_1 and ϕ_2 are $*$ -less and of modal depth ≤ 1 , then $f(\phi) = \perp$
2. For ϕ_1, ϕ_2 $*$ -less and of modal depth ≤ 1 , $f(\phi_1 \wedge [\vdash^*] \phi_2) = s(\phi_1) \wedge \Box s(\phi_2)$, where s is defined on $*$ -less formulas as follows:

$$\begin{aligned} s(p) &= p \\ s(\neg\psi) &= \neg s(\psi) \\ s(\psi \vee \xi) &= s(\psi) \vee s(\xi) \\ s(\langle \vdash \rangle \psi) &= \langle g \rangle s(\psi) \vee \langle h \rangle s(\xi) \end{aligned}$$

Since s is polynomial time computable on $*$ -less formulas of modal depth ≤ 1 , f is polynomial time computable. Now, it is straightforward to prove the following fact by induction. If $M = \langle W, R_\vdash, \pi \rangle$ is a PDL-model, and $M' = \langle W, R_g, R_h, \pi \rangle$ is an $L^\square$ -model, such that $R = R_g \cup R_h$, then for all $*$ -less PDL($\vdash$)-formulas ϕ , and for all nodes $w \in W$, $M \models \phi[w] \Leftrightarrow M' \models s(\phi)[w]$. By making use of this it is easy to prove that f is indeed a reduction from C to $L^\square$ -satisfiability. $\square$

4 Conclusions

In this paper we have seen how to investigate the complexity of Attribute Value logics by making use of a modal perspective. This approach has a number of advantages. In particular, the realisation that AVSs are just Kripke models permits perspicuous model theoretic proofs of NP completeness, and the application of the tiling method for proving undecidability. It should be remarked that the tiling method can easily be applied to a number of other AV logics that have been considered in the literature. For example, they can be used to obtain undecidability results for extensions of L^{KR} employing either the 'Master Modality' of Gazdar *et al.* [8] and [17], or the apparatus of PDL [16]. But a proper discussion of these topics must await the full version of this paper.

Acknowledgements The authors would like to thank Johan van Benthem for his comments on an earlier draft of this paper, and Kees Vermeulen for his helpful question concerning finite numbers of propositional variables.

References

- [1] Balcázar J, Díaz, J. and Gabarró, J.: 1988, *Structural Complexity I*, EATCS Monographs on Theoretical Computer Science 11, Springer Verlag.
- [2] Benthem, J. v.: 1984, Correspondence Theory, in *Handbook of Philosophical Logic II*, edited by D. Gabbay and F. Guenther.
- [3] Berger, R.: 1966, The Undecidability of the Domino Problem, *Mem. Amer. Math. Soc.* **66**.
- [4] Blackburn, P.: 1991, Modal Logic and Attribute Value Structures, in *Colloquium on Modal Logic*, edited by Maarten de Rijke, Dutch Network for Language, Logic and Information.
- [5] Bull, R.: 1970, An Approach to Tense Logic, *Theoria*, **12**, pages 171–182.
- [6] Fischer, M. and Ladner, R.: 1979, Propositional dynamic logic of regular programs, *Journal of Computer and System Sciences* **18**, 194–211.
- [7] Gazdar, Klein, Pullum and Sag.: 1985, *Generalized Phrase Structure Grammar*, Basil Blackwell.
- [8] Gazdar, G., Pullum G., Carpenter, R., Klein, E., Hukari, T. and Levine, R.: 1988, Category Structures. *Computational Linguistics*, **14**, pages 1–19.
- [9] Goranko, V. and Passy, S.: 1989, Using the Universal Modality: Gains and Questions. ITLI prepublication X-90-06, University of Amsterdam.
- [10] David Harel, D.: 1983, Recurring dominoes: making the highly undecidable highly understandable, in *Proc. of the Conference on Foundations of Computing Theory*, Springer Lecture Notes in Computer Science **158**, 177–194.
- [11] Halpern, J. and Moses, M.: 1985, A guide to the modal logics of knowledge and belief, in *Proceedings 9th International Joint Conference on Artificial Intelligence*, 480–490, revised version to appear in *Artificial Intelligence*.
- [12] Hughes, G. and Cresswell, M.: 1968, *A Companion to Modal Logic*, Methuen & Co., Ltd.
- [13] Kaplan, K. and Bresnan, J.: 1982, Lexical functional grammar: A formal system for grammatical representation, in *The Mental Representation of Grammatical Relations*, edited by Joan Bresnan, MIT Press, 173–281.
- [14] Karttunen, L.: 1984, Features and Values, in *Proceedings of the 10th International Conference on Computational Linguistics and the 22nd Annual Meeting of the Association for Computational Linguistics*, Stanford, California, 28–33.
- [15] Kaspar, R. and Rounds, W.: 1986, A logical semantics for feature structures, in *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, Columbia University, New York, 257–266.
- [16] Keller, B.: 1991, *Feature Logics, Infinitary Descriptions and The Logical Treatment of Grammar*, PhD thesis, School of Cognitive and Computing Sciences, University of Sussex, United Kingdom.

- [17] Kracht, M.: 1989, On the Logic of Category Definition, *Computational Linguistics* **15**, 111–113.
- [18] Ladner, R.: 1977, The computational complexity of provability in systems of modal logic, *SIAM Journal on Computing* **6**(3), 467–480.
- [19] Moss, L.: 1991, Completeness Theorems for Logics of Feature Structures, Indiana University Logic Group Preprint No. IULG-91-2. To appear in *Proceedings of the MSRI Workshop on Logic From Computer Science*, edited by Yiannis N. Moschovakis, Springer Verlag.
- [20] Pereira, F and Shieber, S.: 1984, The semantics of grammar formalisms seen as computer languages, in *Proceedings of the 10th International Conference on Computational Linguistics and the 22nd Annual Meeting of the Association for Computational Linguistics*, Stanford, California, 123–129.
- [21] Pollard, C. and Sag, I.: *Information-Based Syntax and Semantics: Volume 1 - Fundamentals*. CSLI Lecture Notes, **13**, Stanford.
- [22] Pollard, J. (Forthcoming). Sorts in unification-based grammar and what they mean, To appear in *Unification in Linguistic Analysis*, edited by M. Pinkal and B. Gregor.
- [23] Pratt, V.: 1979, Models of program logics, in *Proceedings of the 20th IEEE Symposium on Foundations of Computer Science*, 115–122.
- [24] Prior, A.: 1967, *Past, Present and Future*, Oxford University Press.
- [25] Reape, M.: 1991, *An Introduction to the Semantics of Unification-Based Grammar Formalisms* DYANA deliverable R3.2.A, Centre for Cognitive Science, University of Edinburgh, Scotland.
- [26] Robinson, R.: 1971, Undecidability and Nonperiodicity for Tilings of the Plane, *Inventiones Math.* **12**, 177–209.
- [27] Rounds, W. and Kaspar, R.: 1986, A Complete Logical Calculus for Record Structures Representing Linguistic Information, in *Proceedings of the 15th Annual Symposium on Logic in Computer Science*, Cambridge, Massachusetts.
- [28] Rounds, W. and Kaspar, R.: 1990, The Logic of Unification in Grammar, *Linguistics and Philosophy* **13**, 33–58.
- [29] Schild, K.: 1990, A Correspondence Theory for Terminological Logics, Preliminary Report. Projekt KIT-BACK, Technische Universität Berlin, Germany.
- [30] Shieber, S.: 1986, *An Introduction to Unification-Based Approaches to Grammar*, Chicago University Press.
- [31] Smolka, G.: 1988, A Feature Logic with Subsorts, LILOG-REPORT 33, IBM Deutschland GmbH, Stuttgart, Germany.
- [32] Smolka, G.: 1989, Feature Constraint Logics for Unification Grammars, IWBS Report 93, IWBS, IBM Deutschland, Stuttgart, Germany.

Comprehension Grammars Generated from Machine Learning of Natural Languages

Patrick Suppes, Michael Böttner and Lin Liang

1 Introduction and Overview

We are developing a theory of machine language learning in the context of robotic instruction of elementary assembly actions. More specifically, our situations of language learning concern actions like moving to objects, picking up objects and moving objects in environments. Typical objects are screws, nuts, washers, and sleeves of various colours, sizes, and shapes, related to each other by common spatial relations. Typical commands to be comprehended are *Get a screw!* *Go to the black washer behind the screw!* *Put the screw in the left hole!* The system developed so far deals successfully with: English, Chinese, and German, as well as several other languages, which for a lack of space we will report on separately.

The overlap between the utterances that a person is able to understand and those that a person is able to produce may be small. A comprehension grammar would, e.g., be able to derive *Go to the left of the nut screw!* but not *Put a nut the near screw!* although neither of these utterances should be derivable in an English production grammar. A purely comprehension grammar should parse a superset of the set of strings we would expect a production grammar to derive.

Our axioms, given in Section 2, for machine learning of comprehension are long and complicated, even though we analyze here the learning of relatively small fragments of the natural languages considered but they do faithfully reflect the computer implementation, as well as extend the axioms given in our earlier paper (Suppes, Liang, Böttner, 1992), where the relevant literature is discussed. We do believe the learning principles expressed in the axioms can serve as the basis for learning much more extended fragments and represent principles that must be present in any system that starts with no knowledge of the natural language to be learned. The axioms naturally fall into three parts. First are those concerning computations using short-term memory based on learned associations; second those dealing with changes in the state of long-term memory, which include computations updating the comprehension grammar; and third the axiom describing the process of comprehension when more learning is not required to understand a particular command.

In Section 3 we describe the internal language, which is not learned, and then in Section 4 we provide an extensive commentary on the axioms, using, as would be expected, the resources of the internal language. In Section 5 we present detailed results on the comprehension grammars generated, and in Section 6, the final section, we review a number of significant problems we have not yet solved.

The results for the comprehension grammars of English, Chinese and German are linguistically surprising. Once congruence computations (see Axiom 2.5) are made, 9 grammatical rules are common to all three languages, 2 to Chinese and English, 7 to English and German, and 0 for Chinese and German. The surprise is that no special rules for English are required. Eight special rules are required for Chinese and 4 for German. As these numbers suggest and the details confirm, our comprehension grammars are vastly simpler than the corresponding production grammars.

2 Learning System

What we want to describe now is the process of learning in terms of the various events that happen on a given trial. Initially the robot has no knowledge of the lexicon or grammatical structure of the language to be learned. The robot begins a trial in a given state of memory. There are three parts of this memory that are changed due to learning: the probabilistic association relation, the set of grammatical forms and short-term memory of the current command and association computations.

1. The first is the association relation between words of a given language and internal symbols that represent denotations of members of categories in the internal representation language. The semantic categories of action, object, property and spatial relation, as well as the internal language (LISP), are given in advance and not learned. For example, the action of putting will be represented internally, let us say, by the symbol \$p and will have three different linguistic representations in English, German and Chinese. The problem will be to learn in each language what word is properly associated with the internal representation \$p. The same process of association must take place for the internal symbols for objects, properties and spatial relations (Axiom 1.2). But knowing such associations is not enough, contrary to some naive theories of associationism.
2. It is also important to have grammatical forms in long-term memory (Axiom 3). We will not try to lay out everything about grammatical forms that we have in our fully stated theory, but it will be useful to give some examples. Consider the verbal command *Get the screw!* This would be an instance of the grammatical form *A the O!*, where *A* is the category of actions and *O* is the category of objects. As might be expected we do not actually have just a single category of actions, but several subcategories, depending upon the number of arguments required, etc. The central points here, however, are that: (i) the grammatical forms are derived only by generalization from actual instances of verbal commands given to the robot, and (ii) the grammatical forms are used in parsing any new command.
3. The third part of memory that varies is the short-term memory that holds a given verbal command for the period of the trial on which it is effective. This memory content decays and is not available for access after the trial on which a particular command is given (Axiom 1.6). This is also true of any association computations (Axioms 1.1-1.5).

Once a verbal command is given to the robot, the learning program looks up the associations of the words in the verbal command that has been given (Axiom 3a). If associations exist for any of the words, the categories of the associations, which are the categories of the internal interpretation, are also retrieved (Axiom 3b). The categories are substituted for the associated words and an effort is then made to generate recursively the resulting grammatical form (Axiom 3c). For example, if mistakenly the word *get* had been associated to \$s, the internal symbol for *screw*, and *screw* had been associated with \$g, the internal symbol for *get*, then the grammatical form that would have been generated and now found upon a second occurrence of the verbal command would be *O the A!* Now if there were no such grammatical form generated, once the associations were formed such a grammatical form would be created by the process of generalization which is used to generate grammatical forms (Axiom 1.3).

When a grammatical form is stored in memory, also associated with this grammatical form in memory is its internal representation (Axiom 2.6). This is an important part of the memory that changes with learning as well. If the grammatical form is generated, the internal representation is then used to execute the verbal command that has been given (Axioms 1.4 and 3b). If the command is executed correctly then the robot is ready for a new learning trial.

The important case of learning is when no grammatical form can be generated recursively from the grammatical forms in memory to match the form of the given verbal command. In this case the robot is unable to make a response. The correct response must be coerced (Axiom 1.1). On the basis of this coercion, a new internal representation is formed. At this point the critical step comes of a probabilistic association between words of the verbal utterance and the internal denoting symbols of the new internal representation (Axiom 1.2). After this association is made, a new grammatical form is generated (Axiom 1.3), and possibly because of the new associations at least one of the old grammatical forms is deleted, for, according to our axioms, a word or internal denoting symbol can have exactly one association, so when a new association is formed for a word any old associations must be deleted (Axiom 2.7). With the new associations formed and old ones possibly deleted, the robot is ready for the next trial in a new state of memory.

The learning process as described so far must be extended to the dynamic computations, recursive in nature, for updating the grammar as new commands that cannot be parsed by the existing grammar are encountered. Axiom 1.5 formulates the recursive procedure for generating new associations of parts of a grammatical form with parts of the associated internal form. Axiom 2.6 then characterizes the computations for dynamically changing the current grammar by adding a new rule or deleting a now redundant old rule.

2.1 Learning Principles

To facilitate the statement of principles governing the learning process just described, certain notational conventions are useful. First, generally we use Latin letters to refer to verbal commands or their parts, whatever the natural language, and we use Greek letters to refer to internal representations or their parts. The letters a, a_i, a'_i etc. refer to words in a verbal command, and the Greek letters $\alpha, \alpha_j, \alpha'_j$, etc. refer to internal denotations. The Roman letter t , as well as t_i, t'_i refer to terms of a verbal command, and correspondingly, τ, τ_i, τ'_i to terms of an internal representation, i.e., in the present setup, LISP expressions. The symbols s, s' , and $s(t)$, showing that t is a term of s , refer to entire verbal commands, and correspondingly $\sigma, \sigma', \sigma(\tau)$ to entire internal representations. Grammatical forms – either sentential or term forms – are denoted by g or also $g(X)$ to show a category argument of a form; correspondingly the internal representations of a grammatical form are denoted by γ or $\gamma(X)$. We violate our Greek-Latin letter convention in the case of semantic categories or category variables X, X', Y , etc. We use the same category symbols in both grammatical forms and their internal representations. In order to avoid enumeration of cases, when we use $\gamma(X)$ or $\tau(X, Y)$, for example, it is not required that X , or X and Y , occur as free variables in the grammatical form.

Axioms of Learning

1. Association Computations using Working Memory

- 1.1 *Association by contiguity.* If on any trial verbal command s is contiguous with a coerced action that has internal representation σ , then s is associated with σ , i.e., in symbols $s \sim \sigma$.

1.2 *Probabilistic association.* On any trial n , let s be associated to σ in accordance with Axiom 1, let $\{a_i\}$ be the set of words of s not associated to any internal denoting symbol of σ , let $d_n(a_i)$ be the current denotational value of each such a_i and let $\{\alpha_j\}$ be the set of internal denoting symbols not currently associated with any word of s . Then

- i. an element α_j is uniformly sampled without replacement from $\{\alpha_j\}$,
- ii. at the same time an element a_i is sampled without replacement from $\{a_i\}$ with the sampling probability

$$p(a_i) = \frac{d_n(a_i)}{\sum_{\{a_i\}} d_n(a_i)}.$$

- iii. The sampled pairs are associated, i.e. $a_i \sim \alpha_j$.
- iv. Sampling continues until either the set $\{a_i\}$ or the set $\{\alpha_j\}$ is empty.

1.3 *Form generalization.* If at any step of an association computation on any trial, $g(t(Y)) \sim \gamma(\tau(Y))$, $t(Y) \sim \tau(Y)$ and $X \rightarrow \tau(Y)$, then $g(X) \sim \gamma(X)$, where $X \rightarrow \tau(Y)$ is a grammatical rule of the internal language.

1.4 *Form specification.* If at any step of an association computation on any trial, $g(X) \sim \gamma(X)$, $t(Y) \sim \tau(Y)$ and $X \rightarrow \tau(Y)$, then $g(t(Y)) \sim \gamma(\tau(Y))$.

1.5 *Form association.* Let $g \sim \gamma$ at any step of an association computation on any trial,

- (a) If X occurs in g and $(fo X *)$ occurs in γ , then $X \sim (fo X *)$.
- (b) If

- i. wX is the n -th occurrence of a substring of g with $g \sim \gamma$ and w is a **minimal** substring, possibly empty, of occurrences of nondenoting words, and
 - ii. $\tau(X)$ is the **maximal** LISP form of γ , containing the occurrence of X and no other occurrence of denoting categories,
- then $wX \sim \tau(X)$.

(c) If

- i. $X_1 w_1 X_2 \dots w_n X_n$ is a substring of g , where the $X_i, i = 1, \dots, n$ are not necessarily distinct category names and $w_i, i = 1, \dots, n+1$ are substrings, possibly empty, or words that have no association to internal symbols on the given trial,
 - ii. $\tau(X_{\pi(1)}, \dots, X_{\pi(n)})$ is the minimal LISP form of γ containing $X_{\pi(1)}, \dots, X_{\pi(n)}$.
- then $X_1 w_1 \dots w_n X_n \sim \tau(X_{\pi(1)}, \dots, X_{\pi(n)})$,
where π is a permutation of the numbers $1, \dots, n$.

1.6 *Delete contents.* The content of working memory is deleted at the end of each trial. (Axiom 1.2).

2. Changes in State of Long-term Memory

2.1 At the beginning of trial 1, the association relation $\sim$, the congruence relation and the set of grammatical rules G are empty. Moreover, the denotational value $d(a_i)$ is the same for all words a_i .

2.2 The semantic categories and the semantically interpreted internal language I.L., as characterized in Definition 1, are stored in long-term memory and remain unchanged from trial to trial.

2.3 *Denotational Learning Computations.* If at the end of trial n a word a_i in the presented verbal stimulus is associated with some denoting internal symbol α_j of the internal representation σ of s at the end of the trial, then

$$d_{n+1}(a_i) = (1 - \theta)d_n(a_i) + \theta,$$

and if a_i is not so associated

$$d_{n+1}(a_i) = (1 - \theta)d_n(a_i).$$

Moreover, if a word a_i does not occur on trial n , then

$$d_{n+1}(a_i) = d_n(a_i),$$

unless the association of a_i to an internal symbol α_j is broken on trial n , in which case

$$d_{n+1}(a_i) = (1 - \theta)d_n(a_i).$$

2.4 *Grammar computations.* First, if on trial n $g \sim \gamma$ on the basis of Axiom 1.5, then

- (a) if no substring g' of g is already in long-term memory with $g' \sim \gamma'$, $g \sim \gamma$ is stored in long-term memory,
- (b) if a substring g' of g is already in long-term memory with $g' \sim \gamma'$, then g and γ are reduced to $g(X) \sim \gamma(X)$ where X is the category generating γ' .
- (c) This recursive reduction continues until no further reduction is possible. Then the reduced $g'' \sim \gamma''$ is added to long-term memory if not already present.

Second, if $g \sim \gamma$ is in long-term memory at the beginning of trial n but a substring $g' \sim \gamma'$ of g is generated on the basis of Axiom 1.5, then the reduced $g(X) \sim \gamma(X)$ is stored in memory and g , γ and $g \sim \gamma$ are deleted from long-term memory.

2.5 *Congruence computations.* If two G.F.s g and g' are such that

- (a) $g \sim \gamma$ and $g' \sim \gamma$,
- (b) g and g' have only category symbols as denoting terms,
- (c) g and g' have exactly the same sequence of category symbols,

then g is γ -congruent to g' .

2.6 *Probabilistic associations stored in long-term memory.* This association is stored in long-term memory.

2.7 *Form memory trace.* The first time a form generalization (Axiom 1.3) is formed, the word associations on which the generalization is based are stored with it in long-term memory. This memory trace is inherited under the grammar computations (Axiom 2.4).

2.8 *Delete prior associations.* When a word in a verbal command or an internal symbol is given a new association (Axiom 2.3), any prior associations of that word or symbol are deleted from long-term memory.

- 2.9 *Delete form association.* If $a \sim \alpha$ is deleted (Axiom 3), then any generalization for which $a \sim \alpha$ is a memory trace is also deleted from long-term memory, as well as any G.F.s that inherited the memory trace.

3. Comprehension-and-Response axiom

If command s occurs at the beginning of trial n , then:

- (a) Long-term memory is accessed for a possible associated internal symbol for each word of s ;
- (b) The category of each associated internal symbol is retrieved from long-term memory;
- (c) A grammatical form g of s is computed by replacing each word a_i of s , which has an associated internal symbol α_j , by the symbol for the semantic category of α_j , in accordance with Axiom 1.3.
- (d) Using the parser Π on the set G_n of grammatical rules in long-term memory, a parse of g is attempted.
- (e) If exactly one parse g is found, the associated semantic form γ is computed in accordance with Axiom 1.4, and γ is then executed if possible;
- (f) If more than one parse is found, say, the set $\Pi(g)$ of parses, then each associated constructed semantic form γ is tested for compatibility with the given physical environment E_n , and parses incompatible with E_n are deleted from the set $\Pi(g)$; one parse is selected at random from the reduced set $\Pi'(g)$ and the associated γ is executed if possible; if execution is not possible, no other parses are sampled.
- (g) If no parse is found, no response is made.

We comment in detail on the learning process described by the axioms after the next section on the internal language.

3 Internal Language

The internal language is a LISP regimentation of English lexical semantics, but not, of course, English syntax. For example, the internal symbols for spatial relations correspond more closely to the semantics of English prepositions than to that of any of the other languages we have considered. It is an objective of our work to eliminate in the future this English lexical bias. We believe that much can be accomplished in this direction by forming for each set of concepts a partition fine enough to cover most if not all the languages we would be likely to consider. This is not to suggest that such partitions solve a host of other problems. Internal representations will therefore be LISP-expressions. A LISP-expression is any string $(FE_1, \dots, E_n)$ where F is a function symbol and $E_1, \dots, E_n$ are either atoms or LISP-expressions. The atoms refer to properties, actions, spatial relations, and objects of the environment and fall into corresponding categories. In particular, we have category P of properties like \$small, \$black, \$square; category OBJ of special properties serving to single out objects like \$screw, \$nut, \$washer, \$plate, \$hole, \$sleeve; category R of binary relations like \$front, \$back, \$left, \$right; category S of set of objects: *; action categories: A_1 like \$get, \$open, \$close, A_2 like \$go, A_3 like \$put.

In addition to these we have categories of expressions that are not atoms: category A of actions like (fa1 \$get (io (fo1 \$screw))); category O of objects like (fo1 \$screw *); category G of regions like (fo1 \$near (io (fo1 \$screw *)).

Expressions denoting subsets of these categories are derived by the semantic operations listed below, but commitment to a set-theoretical framework is not essential.

Grammar of Internal Language

1.	fo1 (focus-on-object-1)	S	→	(fo1 OBJ *)
2.		S	→	(fo1 P S)
3.		S	→	(fo1 G S)
4.	fo2 (focus-on-object-2)	S	→	(fo2 R S)
5.	io (identify object)	O	→	(io S)
6.	so (select-object)	O	→	(so S)
7.	fa1 (form-action-1)	A	→	(fa1 A ₁ O)
8.	fa2 (form-action-2)	A	→	(fa2 A ₂ G)
9.	fa3 (form-action-3)	A	→	(fa3 A ₃ O G)
10.	fa4 (form-action-4)	A	→	(fa4 O)
11.		A	→	(fa4 D O)
12.	fdir (form-direction)	D	→	(fdir R)
13.	fr1 (form-region-1)	G	→	(fr1 R O)
14.	and (form-and-property)	P	→	(and P P)
15.	or (form-or-property)	P	→	(or P P)
16.	not (form-not-property)	P	→	(not P)

The robot's internal language is the set of expressions of category *A*. Notice that the operations also return expressions of our initial categories, in particular *P* and *S* which means that these categories are recursive. So we will have in our internal language symbols for each content word of a command. The English command *Get a screw!* has the following internal representation (fa1 \$g (so (fo1 \$s *))), where \$g and \$s occur in the internal language as counterparts of the English content words *get* and *screw*. The internal representation given above has three more symbols: *fo1*, *so*, and *fa1*. These symbols are abbreviations of the semantic operations *focus on object*, *select object*, and *form action*. The purpose of these operations is to provide what we think is the procedural structure of the natural language command mentioned above. Before it can perform the action denoted by \$g, the robot has to determine the object of this action. The object that is intended to become the object of the action is returned by the operation *so*. This operation is nondeterministic: it just selects one out of a set of objects. The input for *so* is a set of objects. This set of objects is the output of the operation *fo1*: it takes a property and the set of objects in the robot's environment and returns a set of objects. The set of objects in the robot's environment is represented by the symbol * in our internal language. In our example, the property the environment is checked for is the presence of screws. The procedure *so* then takes this set of all screws of the environment as input and returns an arbitrary screw from this set. This particular screw together with the action *g* is the input of the operation *fa1*.

To illustrate our internal language we give some examples for basic types of action and object structures together with their associated English counterparts.

Action Structures

<i>Get the screw!</i>	~	(fa1 \$get (io (fo1 \$screw *)))
<i>Go near the screw!</i>	~	(fa2 \$go (fr1 \$near ₂ (io (fo1 \$screw *))))
<i>Put the screw behind the nut!</i>	~	(fa3 \$put (io (fo1 \$screw *)) (fr1 \$behind ₃ (io (fo1 \$nut *))))

Object Structures

<i>screw</i>	~	(fo1 \$screw *)
<i>large screw</i>	~	(fo1 \$large (fo1 \$screw *))
<i>screw which is large</i>	~	(fo1 \$large (fo1 \$screw *))
<i>screw which is not large</i>	~	(fo1 (fp- \$large) (fo1 \$screw *))
<i>large black screw</i>	~	(fo1 \$large (fo1 \$black (fo1 \$screw *)))
<i>large and black screw</i>	~	(fo1 (fp and \$large \$black) (fo1 \$screw *))
<i>left screw</i>	~	(fo2 \$left (fo1 \$screw *))
<i>left square screw</i>	~	(fo2 \$left (fo1 \$square (fo1 \$screw *)))
<i>screw behind a nut</i>	~	(fo1 (fr1 \$behind (so (fo1 \$nut *))) (fo1 \$screw *))
<i>large screw behind a nut</i>	~	(fo1 \$large (fo1 (fr1 \$behind (so (fo1 \$nut *))) (fo1 \$screw *)))

Putting object structures into action structures we may derive structures of arbitrary degrees of complexity, as e.g.,

(fa3 \$put (io fo1 (fr1 \$near (io (fo1 \$nut *))) (fo1 \$screw *)) (fr1 \$behind (io (fo2 \$left
(fo1 \$hole *))))))

which corresponds to the English command

Put the screw that is near the nut behind the left hole!

The operations *fo1* and *fo2* have in common that they focus the robot's attention on objects of a certain kind in the set of objects that can be perceived by the robot on the occasion of being given a particular command: *fo1* takes a property and a set of objects and returns the objects that are referred to by the element of *O*. So the expression

(fo1 \$screw *)

returns the set of all objects from the set in the robot's perceptual environment that are screws. In

(fo1 black (fo1 screw *))

fo1 operates recursively: it first returns the set of all screws in the robot's environment and then returns the subset of all black screws. The operation *fo2* focusses on a set of objects that bear another certain relation to a set of objects. A typical example is the English expression *left screw* which, in our internal language, is given the structure

(fo2 \$left (fo1 \$screw *))

A region is a set of locations that stand in a certain relation to at least one object. Regions are built from a binary spatial relation and an object by *fr1*. The operation *so* selects an arbitrary object from a set of perceptually given objects. The operation *io* identifies a unique object satisfying certain properties in a set of perceptually given objects. Not surprisingly a number of operations arise in the context of actions. This has to do with the various subclasses of actions depending on the different valencies exhibited by actions.

4 Detailed Comments on the Axioms

An important and distinguishing feature of the comprehension grammars we generate is that the denoting words of a language, as defined just above, are assigned to semantic rather than syntactic categories. For example, the English word *left* is in the category of spatial relations independent of its differing syntactic roles in the following phrases: *Move left!* *Go to the left!* *Pick up the left screw!* *Pick up the screw which is left of the box!* Similar examples appear in the other languages. Moreover, the nondenoting words are not assigned to any category at all except in the formal sense of the general category of being nondenoting, so that the usual syntactic distinctions that would apply to such words are not made. On the other hand, nondenoting words occur in different roles in grammatical forms.

4.1 Creativity

To get a better understanding of the axioms let us consider two examples that show that the grammars generated allow to comprehend new commands and an even infinite number of new commands. The first is to show that sentences can be understood by the grammar that are "new" to the learner. The second is to show that the grammar can analyse sentences that are longer than any of the sentences of the finite sample of the language to be learned.

Predication test. Once the generalized associations are established the internal representations of "new" commands can be derived. For instance, on the basis of having been learned correctly, *Get the screw!*, *Go to the nut!*, and *Get the nut!*, the following command can be understood by the comprehension grammars, and thus executed by the robot: *Go to the screw!*

Recursivity. Our system does not just derive new structures of the same length but is also able to cope with recursivity by understanding commands longer than any of the commands occurring in the learning sequence. On the basis of having learnt correctly

Get the nut!
Get the plate!
Get the red washer!
Get the nut which is left of the screw!
Get the screw behind the nut!

it would be able to to derive the following command:

Get the red screw behind the plate which is left of the washer!

Assume that, on the basis of previous experience, the robot's long-term memory holds the following associations:

$nut \sim \$n$; $get \sim \$g$; $screw \sim \$s$; $plate \sim \$p$; $washer \sim \$w$;
 $red \sim \$r$; $behind \sim \$b$; $left \sim \$l$

Coerce:

$Get\ the\ nut! \sim (fal\ \$g\ (io\ (fol\ \$n\ *)))$ (1)

$Get\ the\ red\ washer! \sim (fal\ \$g\ (io\ (fol\ \$r\ (fol\ \$w\ *))))$ (2)

$$\text{Get the nut which is left of the screw!} \sim (\text{fal } \$g (\text{io} (\text{fol} (\text{fr1 } \$l (\text{io} (\text{fol } \$s *))) (\text{fol } \$n *)))) \quad (3)$$

$$\text{Get the screw behind the nut!} \sim (\text{fal } \$g (\text{io} (\text{fol} (\text{fr1 } \$b (\text{io} (\text{fol } \$n *))) (\text{fol } \$s *)))) \quad (4)$$

Form generalization applied to (2)–(5) extends memory by

$$A_1 \text{ the OBJ!} \sim (\text{fal } A_1 (\text{io} (\text{fol } \text{OBJ} *))) \quad (5)$$

$$A_1 \text{ the P OBJ!} \sim (\text{fal } A_1 (\text{io} (\text{fol } P (\text{fol } \text{OBJ} *)))) \quad (6)$$

$$A_1 \text{ the OBJ which is R of the OBJ'!} \sim (\text{fal } A_1 (\text{io} (\text{fol} (\text{fr1 } R (\text{io} (\text{fol } \text{OBJ}' *))) (\text{fol } \text{OBJ} *)))) \quad (7)$$

$$A_1 \text{ the OBJ R the OBJ'!} \sim (\text{fal } A_1 (\text{io} (\text{fol} (\text{fr1 } R (\text{io} (\text{fol } \text{OBJ}' *))) (\text{fol } \text{OBJ} *)))) \quad (8)$$

Form association applied to (5)–(8) enables the robot to extend its memory by the following factorizations:

$$\text{OBJ} \sim (\text{fol } \text{OBJ} *) \quad (9)$$

$$\text{the OBJ} \sim (\text{io} (\text{fol } \text{OBJ} *)) \quad (10)$$

$$P \text{ OBJ} \sim (\text{fol } P (\text{fol } \text{OBJ} *)) \quad (11)$$

$$R \text{ of the OBJ} \sim (\text{fr1 } R (\text{io} (\text{fol } \text{OBJ} *))) \quad (12)$$

$$R \text{ the OBJ} \sim (\text{fr1 } R (\text{io} (\text{fol } \text{OBJ} *))) \quad (13)$$

Form generalization applied to (7), (12) and Internal Production Rule 13 which we abbreviate IL Rule 13) yields

$$A_1 \text{ the OBJ which is G!} \sim (\text{fal } A_1 (\text{io} (\text{fol } G (\text{fol } \text{OBJ} *)))) \quad (14)$$

Form generalization applied to (8), (13) and IL rule 13 yields:

$$A_1 \text{ the OBJ G!} \sim (\text{fal } A_1 (\text{io} (\text{fol } G (\text{fol } \text{OBJ} *)))) \quad (15)$$

Form Association applied to (14) and (15) yields:

$$\text{OBJ which is G} \sim (\text{fol } G (\text{fol } \text{OBJ} *)) \quad (16)$$

$$\text{OBJ G} \sim (\text{fol } G (\text{fol } \text{OBJ} *)) \quad (17)$$

Form generalization applied to (16), (9) and IL rule 1 yields:

$$S \text{ which is G} \sim (\text{fol } G S) \quad (18)$$

Form generalization applied to (17), (19) and IL rule 1 yields:

$$S G \sim (\text{fol } G S) \quad (19)$$

Form generalization applied to (9), (10) and IL rule 1 yields:

$$\text{the S} \sim (\text{io } S) \quad (20)$$

Form generalization applied to (9), (11) and IL rule 1 yields:

$$P S \sim (fo1 P S) \quad (21)$$

Form generalization applied to (12), (20) and IL rule 5 yields:

$$R of O \sim (fr1 R O) \quad (22)$$

Form generalization applied to (13), (20) and IL rule 5 yields:

$$R O \sim (fr1 R O) \quad (23)$$

Form generalization applied to (5), (10) and IL rule 7 yields:

$$A_1 O \sim (fa1 A_1 O) \quad (24)$$

Grammatical Computation reduces the memory to the subset of associations (9), (18) - (24). Assume now that the following command is given to the robot:

$$\text{Get the red screw behind the plate which is left of the washer!} \quad (25)$$

This command is more complex than any of the stimulus commands. Notice that its G.F. does not occur in any of the original G.F.s:

$$A_1 \text{ the } P \text{ OBJ } R \text{ the } OBJ' \text{ which is } R' \text{ of the } P \text{ OBJ}''! \quad (26)$$

Let us start from (24). We now make 9 successive applications of *Form specification*. We show only the associations and internal production rules used. By (20) and IL Rule 5 applied to (24):

$$A_1 \text{ the } S \sim (fa1 A_1 (io S))$$

By (19) and IL Rule 3:

$$A_1 \text{ the } S G \sim (fa1 A_1 (io (fo1 G S)))$$

By (23) and IL Rule 13:

$$A_1 \text{ the } S R O \sim (fa1 A_1 (io (fo1 (fr1 R O) S)))$$

By (20) and IL Rule 5:

$$A_1 \text{ the } S R \text{ the } S' \sim (fa1 A_1 (io (fo1 (fr1 R (io S')) S)))$$

By (18) and IL Rule 3:

$$A_1 \text{ the } S R \text{ the } S' \text{ which is } G \sim (fa1 A_1 (io (fo1 (fr1 R (io (fo1 G S')) S))) S)))$$

By (22) and IL Rule 13:

$$\begin{aligned} A_1 \text{ the } S R \text{ the } S' \text{ which is } R' \text{ of } O \sim \\ (fa1 A_1 (io (fo1 (fr1 R (io (fo1 (fr1 R' O) S')) S))) S))) \end{aligned}$$

By (20) and IL Rule 5:

$$\begin{aligned} A_1 \text{ the } S R \text{ the } S' \text{ which is } R' \text{ of the } S'' \sim \\ (fa1 A_1 (io (fo1 (fr1 R (io (fo1 (fr1 R' (io S'')) S')) S))) S))) \end{aligned}$$

By (21) and IL Rule 2:

$$A_1 \text{ the } P \text{ } S \text{ } R \text{ the } S' \text{ which is } R' \text{ of the } S'' \sim \\ (fa1 A_1 (io (fo1 (fr1 R (io (fo1 (fr1 R' (io S'')) S')) (fo1 P S))))))$$

By (9) and IL Rule 1:

$$A_1 \text{ the } P \text{ } OBJ \text{ } R \text{ the } OBJ' \text{ which is } R' \text{ of the } OBJ'' \sim \\ (fa1 A_1 (io (fo1 (fr1 R (io (fo1 (fr1 R' (io (fo1 OBJ'' *) (fo1 OBJ' *)))) (fo1 P (fo1 OBJ *))))))$$

Notice that this association matches the G.F. of (26).

4.2 Contiguity

Our learning algorithm is based on the idea that the structure of the internal language can be used to derive the constituent structure of natural language utterances. This idea easily invites the objection that the algorithm is successful for languages with only contiguous constituents but inevitably fails for languages with noncontiguous constituents. A case in point is the German command

$$\text{Heb die Schraube hoch, die gross ist!} \quad (1)$$

Note that in this command *Schraube die gross ist* forms one constituent from the standpoint of our internal language. This constituent, however, is not contiguous since it is interrupted by the constituent *hoch*. Now there is in German also a way to express the command expressed by (1) with contiguous constituents:

$$\text{Heb die Schraube, die gross ist, hoch!} \quad (2)$$

(We disregard for the moment that one could argue that (2) has also noncontiguous constituents since *heben* and *hoch* are forms of one word *hochheben*.) But (2) is considered less natural and the algorithm should be able to come up with a grammar on the basis of both (1) and (2). In fact, our algorithm is able to derive a grammar in both cases. The grammar derived from (2) will have among its rules the following one:

$$A_4 O \text{ ist } D \sim (fa4 A_4 D O) \quad (3)$$

The grammar derived from (1), however, will not have this rule but the following rule instead:

$$A_4 \text{ die } S \text{ } D \text{ die } P \text{ ist} \sim (fa4 A_4 (io (fo1 P S)) D)$$

Note that (2) is less general than (1). All the rules specific to German are of that kind.

4.3 Context-free, but not Regular Languages

Our axioms can generate grammars for languages that are context-free but not regular. This may happen because of the presence of so-called center-embedding structures exhibited by the following German example:

1. *Nimm die nahe der Mutter befindliche Schraube!*
2. *Nimm die nahe der hinter dem Loch befindlichen Mutter befindliche Schraube!*
3. *Nimm die nahe der hinter dem vor der Platte befindlichen Loch befindlichen Mutter befindliche Schraube!*

...

Our axioms are able to deal with center-embedding structures by having the following associations in memory:

$$R \text{ der } S' \text{ befindliche } S \sim (\text{fo1 (fr R (io } S')) S).$$

This amounts to having the following rule in a grammar for German:

$$S \rightarrow R \text{ der } S' \text{ befindliche } S.$$

4.4 Concept of Denotational Value

In the probabilistic theory of machine learning of natural language which we have been developing, we have encountered in a new form a standard problem in the analysis of the semantics of natural language, namely, how to handle words that are nondenoting. We do not mean nondenoting in some absolute sense, but relative to our standard set of semantical categories. It may well be that in some elaborate set-theoretical semantics of natural language, nondenoting words like the definite article *the* denote a complicated set-theoretical function. We have something simpler and closer to the common man's view of what denotations are. We take as denoting words color words, common nouns, familiar concrete action words, etc.

When a child learning a first language or an older person learning a second language first encounters utterances in that new language, there is no uniform way in which nondenoting words are marked. There is some evidence that various prosodic features are used in English and other languages to help the child. For example, in many utterances addressed to very young children the definite or indefinite article is not stressed but rather the common noun it modifies, as in the expression *Hand me the cup!* But such devices do not seem uniform and in any case are not naturally available to us in our machine-learning research, where we use written input of words without additional prosodic notation. A central feature of our approach to machine learning is the probabilistic association between words of the natural language being learned and internal symbols that denote, as explained earlier.

It is appropriate that at the beginning all words are treated equally and so the associations are formed from sampling based on a uniform distribution. On the other hand, after many words have been learned and a good piece of language has been acquired by the robot, it is very unnatural, and also inefficient, if the robot is now given the esoteric command *Get the astrolabe!* to have the internal symbol *\$ast* to be associated with equal probability with the definite article *the* and *astrolabe*—we assume here that the association of *get* is already correctly fixed. After much experience, what we want is that there is very little chance of associating the definite article *the* with any denoting symbol.

To incorporate such learning, many variant learning models are easily formulated. We have restricted ourselves in Axiom 2.3 to a familiar linear learning model (Estes and Suppes, 1959). We begin with each word w of the given natural language having the value $d_1(w) = 1$. It is obvious from Axiom 2.3 that for all trials n , $0 \leq d_n(w) \leq 1$. For reasons we shall not expand upon here we chose as the value of the learning parameter $\theta = 0.03$. We show in

English		Chinese		German	
word	d-value	word	d-value	word	d-value
plate	0.997	ban	0.999	Platte	0.998
nut	1.000	luomu	0.999	Mutter	0.999
large	0.967	dade	1.000	gross	1.000
place	0.986	fang	0.970	bring	0.989
in	0.972	jing	0.999	in	0.889
the	0.003	nage	0.007	die	0.002
a	0.003	yige	0.005	eine	0.002
		ba	0.001		
		na4	0.127		
		na4li	0.246		

Table 1: Selected Denotational Values

Table 1 the denotational values of selected words from English, Chinese and German after approximately 400 learning trials using the corpus of commands described in Section 5. As desired the denotational value of the denoting words in the sense defined above remain close to 1, and the nondenoting words approach 0.

4.5 Congruence of Nondenoting Words

To reduce the number of grammatical rules and facilitate comparison of rules for different languages, we introduced in the last part of Axiom 2.5 the concept of comprehension congruence, applied here only to the limited case of nondenoting words. The sense of congruence use has, as the formulation of the axiom shows, a specific and definite content. The idea of congruence used here was developed earlier in Suppes (1973, 1991). To illustrate ideas, consider this German case:

Nimm die Schraube! ~ (fa1 \$get (io (fo1 \$screw *)))
Nimm den Ring! ~ (fa1 \$get (io (fo1 \$washer *)))
Nimm das Loch! ~ (fa1 \$get (io (fo1 \$hole *)))

Now generalize to the grammatical forms without denoting words, only category symbols, and we have:

A_1 *die OBJ!* ~ (fa1 A1 (io (fo1 Obj *)))
 A_1 *den OBJ!* ~ (fa1 A1 (io (fo1 Obj *)))
 A_1 *das OBJ!* ~ (fa1 A1 (io (fo1 Obj *)))

Then, using γ -congruence, we may write as a single grammatical form

$$A_1 [die] OBJ! \sim (fa1 A1 (io (fo1 Obj *)))$$

where

$$[die]_i = (die, den, das)$$

and i is the rule number, which indexes the internal semantic form γ . We also note that in many cases, we include in the congruence class $[w]_i$ the empty string ϵ , if the corpus has generated such a variant grammatical form. For example, prior to computing congruence, we generated for Chinese the following three rules:

$$\begin{array}{lll}
A \rightarrow O A_1 & \sim & (\text{fal } A1 \text{ } O) \\
A \rightarrow ba \text{ } O A_1 & \sim & (\text{fal } A1 \text{ } O) \\
A \rightarrow xianzai \text{ } ba \text{ } O A_1 & \sim & (\text{fal } A1 \text{ } O)
\end{array}$$

Computing γ -congruence, we reduce these three rules to a single one:

$$A \rightarrow [ba] A_1 \text{ } O \sim (\text{fal } A1 \text{ } O)$$

where $[ba]_i = (ba, xianzai \text{ } ba, \epsilon)$.

In reporting the grammatical results for English, Chinese and German, we use an ordered triple notation for a given congruence class of nondenoting words of a rule common to all three languages: first the English words, then the Chinese words and finally the German words, with ϵ occurring as needed for each language. Thus the first common rule listed in Section 5 is

$$O \rightarrow [DA] S,$$

where $[DA] = (\text{the; nage, zhege; das, den, dem, der, die})$.

5 Grammars Generated

Comprehension grammars have been generated so far for English, Chinese, German, French, Dutch and Korean. Mainly because the first languages of the authors are English, German and Chinese respectively, we will concentrate on these three languages with remarks about the others. The corpus used has consisted of 456 commands, the last 60 of more complicated ones. In the first group commands range from the simple *Get the screw!* to *Place the screw on the plate!* In the last 60 a typical more complicated command is the following: *put a small nut on the screw behind the washer left of the plate; ba yige xiaode luomu fang zai nage ban zuobian de dianquan houmian de nage luosiding shang; tu eine klein-e Mutter auf die Schraube hinter dem Dichtungsring links von der Platte.*

Close translations of the 456 commands were used for the other languages. It was intended that the translations would be semantically equivalent but would be faithful to idiomatic constructions in the given language.

The details of the learning curves that are generated from our procedure of machine learning are as described in the axioms but will not be analyzed here, because we already have a very thorough discussion of learning curves in our earlier paper (Suppes, Liang, Böttner, 1992). We also do not analyze here in detail the learning of denotational value as characterized in Axiom 2.3. We do show in Table 1 the denotational values for some typical words, denoting and nondenoting in each of the three languages, English, Chinese and German, after approximately 400 learning trials for English and Chinese, and 800 for German. It is evident from the table that the denoting words have a denotational value that is close to 1, and the nondenoting words a value close to 0 as desired. The empirical details of the denotational model are being written up in another paper, but it is important to emphasize that just as we assume no grammar for any of the languages we study, we also do not assume in our machine learning an a priori classification of words as denoting or nondenoting.

We now turn to the three grammars generated for English, Chinese and German. The grammatical rules are written in context-free notation, with the associated internal semantic forms shown on the right. The congruence classes are listed separately. The generation particularly uses Axiom 2.4 on grammar computations. Using the axiom on congruence

(Axiom 2.5) to collapse rules that differ only in the occurrence of semantically equivalent non-denoting words, the number of rules for English is 18, for Chinese 19, and for German 20. What is of perhaps greater interest is the analysis of the structure of common rules in comparison with special rules for the particular languages. The basic numerical data are these. There are 9 rules that are common to English, Chinese and German where 'commonality' means that the category such as definite article now includes words from each of the three languages, in this particular case for example, *the*, *zhege*, *nage*, *den*, *das*, *die*, *dem*, and *der*. In addition, there are 2 rules common to English and Chinese that are not in the German grammar, there are 7 rules common to English and German that are not used in Chinese, and no rules common only to Chinese and German.

The 9 rules common to English, Chinese and German are the following:

1. $O \rightarrow [DA]S \sim (io\ S)$
2. $O \rightarrow [IA]S \sim (so\ S)$
3. $S \rightarrow P\ S \sim (fo1\ P\ S)$
4. $S \rightarrow OBJ \sim (fo1\ OBJ\ *)$
5. $G \rightarrow R\ [PO]O \sim (fr1\ R\ O)$
6. $D \rightarrow R \sim (fdir\ R)$
7. $P \rightarrow P\ [\&]P \sim (and\ P\ P')$
8. $P \rightarrow [\neg]P \sim (not\ P)$
9. $P \rightarrow P\ [V]P \sim (or\ P\ P')$

$[DA]_1 = (the; nage, zhege; das, dem, den, der, die),$

$[IA]_2 = (a; yige; eine, einem, einen, einer),$

$[PO]_5 = (of, \epsilon; \epsilon; von, \epsilon),$

$[\&]_7 = (and; he; und),$

$[\neg]_8 = (not; busi; nicht),$

$[V]_9 = (or; huozhe; oder)$

where the congruence classes for non-denoting words are shown below the nine rules.

What is remarkable about the above 9 rules is that none of them are high level rules for the generation of complete commands. The first 4 deal with the generation of object phrases, Rule 5 with the generation of a description of a region, Rule 6 with a direction, and the last 3 with properties.

There are two common rules for English and Chinese not used in the German grammar:

1. $A \rightarrow [ADV] A_4 D O \sim (fa4\ A4\ D\ O)$
 2. $A \rightarrow A_4 O \sim (fa4\ A4\ O)$
- $[ADV]_1 = (now, so, \epsilon; \epsilon)$

Note that these two rules are high level rules for generating commands.

The 7 rules common to English and German are the following:

1. $A \rightarrow [ADV] A_1 O [COP] \sim (fa1\ A1\ O)$
2. $A \rightarrow [ADV] A_2 G [COP] \sim (fa2\ A2\ G)$
3. $A \rightarrow G A_3 O \sim (fa3\ A3\ O\ G)$
4. $A \rightarrow A_3 O [COP] G \sim (fa3\ A3\ O\ G)$
5. $A \rightarrow [ADV] A_4 [COP] O D \sim (fa4\ A4\ D\ O)$
6. $S \rightarrow S [RP] P \sim (fo1\ P\ S)$
7. $S \rightarrow S [RP] G \sim (fo1\ G\ S)$

$[ADV]_1 = (now, so, \epsilon; nun, \epsilon)$
 $[COP]_1 = (\epsilon; ist, \epsilon)$
 $[ADV]_2 = (now, so, \epsilon; dann, jetzt, nun, \epsilon)$
 $[COP]_2 = (\epsilon; ist, \epsilon)$
 $[COP]_4 = (\epsilon; ist, \epsilon)$
 $[ADV]_5 = (now, so, \epsilon; \epsilon)$
 $[ADV]_5 = (\epsilon; jetzt, nun, \epsilon)$
 $[RP]_6 = (that\ is, which\ is; der, die, ist\ die)$
 $[RP]_7 = (that\ is, which\ is, \epsilon; der, die, die\ sich, \epsilon)$

Note that 5 of the 7 are high-level rules for generating commands. The remaining 2 are for generating object phrases.

The rules special for German and the congruence classes are the following:

1. $A \rightarrow A_4 [einen] S D [der] P [ist] \sim (fa5 A_4 D (so (fo1 P S)))$
2. $A \rightarrow A_4 [nun\ die] S D die G ist \sim (fa4 A_4 D (io (fo1 G S)))$
3. $A \rightarrow A_4 die S R von\ dem S D der G ist \sim (fa4 A_4 D (io (fo1 (fr1 R (io (fo1 G S))) S)))$
4. $A \rightarrow A_4 die S R der S D die P ist \sim (fa4 A_4 D (io (fo1 (fr1 R (io (fo1 P S))) S)))$

$[der]_1 = (der, die)$
 $[einen]_1 = (einen, jetzt\ eine)$
 $[ist]_1 = (ist, \epsilon)$
 $[nun]_2 = (nun\ die, die)$

The 8 rules special for Chinese, are the following:

1. $A \rightarrow [ba] O A_1 \sim (fa1 A_1 O)$
2. $A \rightarrow [xianzai] G [na4] A_2 \sim (fa2 A_2 G)$
3. $A \rightarrow zai G A_3 O \sim (fa3 A_3 O G)$
4. $A \rightarrow [ba] O A_3 [dao1] G \sim (fa3 A_3 O G)$
5. $A \rightarrow [ba] O A_4 D \sim (fa4 A_4 D O)$
6. $O \rightarrow [ba] S \sim (io S)$
7. $S \rightarrow G [de] S \sim (fo1 G S)$
8. $G \rightarrow O [de] R \sim (fr1 R O)$

$[ba]_1 = (ba, xianzai\ ba, \epsilon)$
 $[xianzai]_2 = (chao, name, xianzai)$
 $[na4]_2 = (na4, na4li)$
 $[ba]_4 = (ba, \epsilon)$
 $[dao1]_4 = (dao1, zai, \epsilon)$
 $[ba]_5 = (ba, xianzai\ ba, \epsilon)$
 $[ba]_6 = (ba, dao1, xianzai, zai, \epsilon)$
 $[de]_7 = (de, de\ nage)$
 $[de]_8 = (de, \epsilon)$

Five of these 8 are high-level rules for generating commands. Rules 6 and 7 are for generating object phrases, and Rule 8 for generating region description.

The big surprise is that no special grammatical rules are required for English. In no sense did we anticipate this outcome.

The congruence class for Rule 6 is intuitively incorrect. The reason is simple to explain. The Chinese particles *xianzai*, *dao1*, *zai*, and *ba* are not associated with object phrases, as

the rules suggest, but with other categories of words. For example, *zianzai* has a meaning that is close to the English adverb *now*, the particle *dao1* is associated with verbs, as is *ba*. An example from our corpus of the use of *zai* would be the following. *Get the screw on the plate.* has the following translation in Chinese:

ba nage zai (nage) ban shang de luosiding naqi.
the (the) plate on screw get.

In this example, we mention that the second occurrence of *nage* is deleted because it is awkward to say in Chinese *nage zai nage*. Note also in this example that although *ba* is associated with the verb *naqi*, *ba* comes at the beginning of the sentence, *naqi* at the end which would make for difficulties if we had to assign a specific semantic meaning to *ba*. What is important is that at the level of the commands we are considering, the non-intuitive Rule 6 is satisfactory for the purposes of comprehension. It would of course lead to very bad Chinese if applied to the production of utterances. We emphasize once again that in considering these Chinese examples, we are generating comprehension by uniform procedures that are the same across all languages. We are not claiming this can be done for production or even that for the ultimate reaches of comprehension it will be satisfactory. On the other hand it is important for machine learning purposes to understand how far uniform procedures can be satisfactorily exploited.

6 Problems not solved

Problems arise from the following assumptions: the association relation is one-one, and the internal language is a LISP regimentation of English lexical semantics, as already mentioned. Here are some examples.

1. Contraction of denoting and nondenoting

(a) German:

Geh zur Schraube! ~ (fal \$go (\$to (io (fol \$screw *))))
Geh zu der Schraube! ~ (fal \$go (\$to (io (fol \$screw *))))
Geh zum Loch! ~ (fal \$go (\$to (io (fol \$hole *))))
Geh zu dem Loch! ~ (fal \$go (\$to (io (fol \$hole *))))

(b) French:

Va au trou! ~ (fal \$go (\$to (io (fol \$hole *))))
Va à la vis! ~ (fal \$go (\$to (io (fol \$nut *))))

2. Only one word for different internal symbols (French)

Ferme la porte! ~ (fal \$close (io (fol \$door *)))
Ferme la porte! ~ (fal \$lock (io (fol \$door *)))

3. More than one synonymous word corresponding to same denoting symbol

Put the screw near the washer!
~ (fa3 \$put (io (fol \$screw *)) (fr1 \$near (io (fol \$washer *))))
Place the screw near the washer!
~ (fa3 \$put (io (fol \$screw *)) (fr1 \$near (io (fol \$washer *))))

4. More than one nonsynonymous word corresponding to same denoting symbol

(a) German:

Leg die Schraube in die Schachtel!

~ (fal \$put (fo \$screw *) fr1 \$in (io (fo \$box *))))

Steck die Schraube in das Loch!

~ (fal \$put (fo \$screw *) (fr1 \$in (io (fo \$hole *))))

(b) French:

vis ronde ~ (fo1 \$round (fo1 \$screw *))

douille cylindrique ~ (fo1 \$round (fo1 \$washer *))

5. Concepts that are expressed easily in one language can be more difficult to express in another language. For example, the English *go behind the plate* does not have a natural direct translation in Dutch, even though *go* can generally be translated by *gaan* in Dutch.

As a consequence, we made certain adjustments in the corpora of natural languages:

1. Since our system currently cannot identify different forms of the same denoting word, we mark the boundary between stem and ending by a hyphen, e.g., in French *vis rond-e* or German *dem rund-en Loch*
2. There are no words in French and German that correspond to the English verb *pick up* with respect to its "constructional range". We chose the German translation *hochheben* because it resembles the English construction at least for the imperatives under consideration where the verb gets split into two parts and *heben* can be associated with *pick* and *hoch* can be associated with *up*. On the other hand, in cases like *pick a screw!* the word *heben* does not appear to be usable and we switched to *nehmen*. In French we used *ramasser* as a translation for *pick up* and *saisir* as a translation for *pick*.
3. The English preposition *on* has no unique counterpart in German and French. Some contexts require it to be translated by *auf/sur* as in, e.g., *Put the screw on the plate!* for other contexts a translation by *an/à* would be more natural as in, e.g., *Put the nut on the screw!* We used *auf* and *sur* in all cases even in cases where the visual scene clearly suggests a different preposition.
4. If a language had no single word for a certain English expression but had only an expression which was longer than one word, we used this and marked it by hyphenations in a fashion that the machine would take it for one word: for French we used *de-taille-moyenne* for the English adjective *medium*.
5. In French we used *à* instead of *au* since *à* is needed as a counterpart for a denoting symbol of the IL. We also used *de le*, *de un*, and *de une* instead of *du*, *d'un*, and *d'une* respectively.

7 Acknowledgements

The research of Michael Böttner has been supported by grant 683/1-2 from the Deutsche Forschungsgemeinschaft. We are indebted to Michael Donio for the analysis of French and Paul Meyer for the analysis of Dutch.

8 References

Estes, W. K., and P. Suppes (1959) Foundations of linear models, R. R. Bush and W. K. Estes (eds.), *Studies in Mathematical Learning Theory*, Stanford University Press, Stanford

Suppes, P. (1973) Congruence of meaning. *Proceedings and Addresses of the American Philosophical Association*, 46, 21-38

Suppes, P. (1991) *Language for Humans and Robots*, Blackwell, Oxford

Suppes, P., Liang, L. and Böttner, M. (1991) Complexity issues in robotic machine learning of natural language, L. Lam and V. Naroditsky (eds.), *Modeling Complex Phenomena*, Springer, New York

An update semantics for dynamic predicate logic *

Paul Dekker

ILLC/University of Amsterdam

Abstract

In this paper I propose an update semantics for Groenendijk and Stokhof's dynamic predicate logic. In the proposed system, sentences are interpreted as (partial) update functions on a domain of information states which are sets of partial variable assignments. I show how adverbs of quantification (symmetric and asymmetric) and generalized quantifiers can be introduced in a perspicuous and uniform way.

1 Introduction

Groenendijk and Stokhof, in [1990], characterize the meaning of a sentence in a *static semantics* as the set of indices at which the sentence is true. Such a set of indices, also called an information state, defines the information content of the sentence. In a *dynamic semantics*, it is not the information content, but the information change potential of a sentence that is regarded as constituting its meaning. In a dynamic semantics the meaning of a sentence is a function on the domain of information states.

Groenendijk and Stokhof compare two examples of such a dynamic semantics, namely dynamic predicate logic (*DPL*, Groenendijk and Stokhof [1991]) and update semantics (*US*, Veltman [1990]). The two systems formalize different aspects of the dynamics of discourse and in their present formulations they have conflicting logical properties. This paper sets out to reconcile the two systems by formulating an update semantics for *DPL*.

I will proceed as follows. In the next section I sketch *DPL* and *US* and argue for a genuine update formulation of *DPL*. In section 3 such an update style *DPL*, *EDPL*, is presented and discussed, and in section 4 it is extended with quantifiers.

2 Two theories of dynamic semantics

2.1 Dynamic predicate logic

DPL gives a dynamic interpretation of the language of first order predicate logic that accounts, among other things, for intersentential anaphoric relationships like we find them in *A cowgirl meets a boy. She slaps him.* Like in discourse representation theory and file change semantics, natural language noun phrases in *DPL* are associated with

* I would like to thank David Beaver, Jeroen Groenendijk, Herman Hendriks, Martin Stokhof and Frank Veltman for pleasant and instructive discussion. I have had great benefit from their comments. Of course, remaining errors are mine.

variables, or discourse markers, and information states determine what values they can have given the conditions imposed on them in the course of a discourse.

Information states in *DPL*, then, contain information about the values of variables. They are modeled as sets of variable assignments that determine what are the possible values of the variables at a certain stage in the processing of a discourse. So, if D is a domain of individuals, and V the set of variables, then D^V is the set of variable assignments and $S = \mathcal{P}(D^V)$, the set of subsets of all variable assignments, is the set of information states.

The language of *DPL* is that of predicate logic, but for ease of exposition I disregard individual constants and identity. The semantics is defined with respect to a model $M = \langle D, F \rangle$ consisting of a non-empty set of individuals D and an interpretation function F that assigns sets of n -tuples of objects to n -ary relation expressions. (I omit reference to M whenever this does not lead to confusion.) The interpretation of formulas is a function on the domain of information states:

Definition 2.1 (Semantics of DPL)

- $s[Rx_1 \dots x_n] = \{i \in s \mid \langle i(x_1), \dots, i(x_n) \rangle \in F(R)\}$
- $s[\neg\phi] = s - \downarrow[\phi]$
- $s[\exists x\phi] = s[x][\phi]$
- $s[\phi \wedge \psi] = s[\phi][\psi]$

where

$$\begin{aligned} \downarrow[\phi] &= \{i \mid \{i\}[\phi] \neq \emptyset\} \\ s[x] &= \{j \mid \exists i \in s \exists d \in D: i[x/d] = j\} \end{aligned}$$

The interpretation of an atomic formula in a state s retains all those variable assignments in s with respect to which the atomic formula is true in a classical sense. The negation of ϕ subtracts those i in s which constitute a context $\{i\}$ with respect to which the interpretation of ϕ does not produce the absurd state $\emptyset$. Conjunction is just function composition. The characteristic clause concerns the interpretation of the existential quantifier. If a formula $\exists x\phi$ is interpreted in a state s , the variable x is reinstated, and next ϕ is interpreted.

DPL licenses the following equivalences:

Fact 2.1 (Weak donkey equivalences)

- $(\exists x\phi \wedge \psi) \Leftrightarrow \exists x(\phi \wedge \psi)$
- $((\phi \wedge \psi) \wedge \chi) \Leftrightarrow (\phi \wedge (\psi \wedge \chi))$

It is typical of *DPL* that the first equivalence also holds if the variable x is free in ψ . These equivalences therefore allow *DPL* to deal with the following textbook example (which explains one half of the equivalences' label):

- (1) A farmer owns a donkey. He beats it.

$$(\exists x(Fx \wedge \exists y(Dy \wedge Oxy)) \wedge Bxy) \Leftrightarrow \exists x(Fx \wedge \exists y(Dy \wedge (Oxy \wedge Bxy)))$$

This sequence of sentences turns out to be equivalent with the sentence *A farmer owns a donkey that he beats*.

The next fact follows from the weak donkey equivalence, on the classical definition of $\rightarrow$ and $\forall x$:

Fact 2.2 (Strong donkey equivalences)

- $(\exists x \phi \rightarrow \psi) = \neg(\exists x \phi \wedge \neg \psi) \Leftrightarrow \neg \exists x(\phi \wedge \neg \psi) = \forall x(\phi \rightarrow \psi)$
- $((\phi \wedge \psi) \rightarrow \chi) = \neg((\phi \wedge \psi) \wedge \neg \chi) \Leftrightarrow \neg(\phi \wedge (\psi \wedge \neg \chi)) = (\phi \rightarrow (\psi \rightarrow \chi))$

This enables *DPL* to deal with the museum piece donkey sentences:

- (2) If a farmer owns a donkey he beats it.
 $(\exists x(Fx \wedge \exists y(Dy \wedge Oxy)) \rightarrow Bxy) \Leftrightarrow \forall x(Fx \rightarrow \forall y((Dy \wedge Oxy) \rightarrow Bxy))$
- (3) Every farmer who owns a donkey beats it.
 $\forall x((Fx \wedge \exists y(Dy \wedge Oxy)) \rightarrow Bxy) \Leftrightarrow \forall x(Fx \rightarrow \forall y((Dy \wedge Oxy) \rightarrow Bxy))$

These sentences are assigned their so called ‘strong’ readings (hence the label of the equivalences). Both sentences state that every farmer beats every donkey he owns. It is the merit of *DPL* that it gives a compositional treatment of these examples.

A property of *DPL* that distinguishes it from classical, static, theories of interpretation, is that it has a non-eliminative semantics:

Fact 2.3 (Non-eliminativity)

- $s[\phi] \not\subseteq s$

Interpretation in *DPL* does not merely involve eliminating possibilities, but it may also involve introducing new possibilities, viz., by existential quantification. A characteristic property of *DPL* updates is furthermore that they are all distributive:

Fact 2.4 (Distributivity)

- $s[\phi] = \bigcup_{i \in s} \{i\}[\phi]$

Distributivity means that in the update of a state s with a formula ϕ only properties of the individual elements of s count.

Truth and entailment are defined as follows:

Definition 2.2 (Truth and entailment in DPL)

- ϕ is true in s , $s \models \phi$, iff $s \subseteq \downarrow[\phi]$
- $\phi_1, \dots, \phi_n \models \psi$ iff $\forall M, s: s[\phi_1]_M \dots [\phi_n]_M \models_M \psi$

A formula ϕ is true in a state s iff all possible variable assignments $i \in s$ constitute a state $\{i\}$ with respect to which ϕ can be successfully computed (i.e., with respect to which the interpretation of ϕ does not produce the absurd state $\emptyset$). A conclusion ψ follows from a sequence of premisses $\phi_1, \dots, \phi_n$, if the update of any information state s with $\phi_1, \dots, \phi_n$, respectively, produces a state in which ψ is true. The *DPL* notion of entailment is a dynamic one. For instance, $\exists x Fx$ entails Fx . This fact corresponds to the following line of elementary reasoning:

- (4) If a man comes from Rhodes, he likes pineapple-juice. A man I met yesterday comes from Rhodes. So, he likes pineapple-juice.
 $(\exists x(Mx \wedge Rx) \rightarrow Lx), \exists x(Mx \wedge Rx) \models Lx$

2.2 Update semantics

I now turn to Veltman’s first example of an update semantics in [1990]. It is sketched in its most rudimentary form as a (dynamic) propositional logic with an additional

sentential operator $\Diamond$.

The kind of information *US* deals with is information about the world. Information states are modeled as subsets of the set of possible worlds W . For someone in information state s , each world in s might correspond to the real world.

Interpretation in *US* involves update of information about the world. An atomic sentence p in *US* updates an information state s by eliminating the worlds in s which are inconsistent with p , and negation and conjunction are interpreted as set subtraction and composition, respectively. The interesting bit comes in with the operator $\Diamond$. In an information state s , $\Diamond\phi$ tests whether ϕ is still consistent with s . Like its natural language counterpart *might*, it reflects upon the present information state and expresses that that state can be consistently updated with ϕ .

The semantics of *US* is defined with respect to a model $M = \langle W, V \rangle$ consisting of a set of worlds W and an interpretation function V that assigns sets of worlds to proposition letters. (Again, reference to M is omitted when that does not lead to confusion.) The interpretation of a formula is defined as follows:

Definition 2.3 (Update semantics)

- $s[p] = \{i \in s \mid i \in V(p)\}$
- $s[\neg\phi] = s - s[\phi]$
- $s[\Diamond\phi] = \{i \in s \mid s[\phi] \neq \emptyset\}$
- $s[\phi \wedge \psi] = s[\phi][\psi]$

As is evident from the semantics of *US*, the result of interpreting a formula in a state s is always a subset of s . Interpretation can only *eliminate* possibilities:

Fact 2.5 (Eliminativity)

- $s[\phi] \subseteq s$

This fact implies that interpretation guarantees update of information.

The $\Diamond$ -operator reflects upon the specific stages in the process of information growth. The interpretation of $\Diamond\phi$ in a state s returns s if ϕ is acceptable in s , and the absurd state $\emptyset$ if ϕ is not acceptable in s . As a consequence, conjunction in *US* is non-commutative:

Fact 2.6 (Non-commutativity)

- $\phi \wedge \psi \not\equiv \psi \wedge \phi$

An example of a non-commutative conjunction is $\Diamond\phi \wedge \neg\phi$. This conjunction, with this order of conjuncts, is consistent. On the other hand, the commutation $\neg\phi \wedge \Diamond\phi$ of this conjunction is inconsistent. The following pair of examples exemplifies this pattern (granted that we know that John is not Mary):

- (5) Somebody is knocking at the door. It might be John. It's Mary.
- (6) *Somebody is knocking at the door. It's Mary. It might be John.

If somebody hears someone knocking at the door, he may of course be curious who it is and not exclude the possibility that it is, say, John. Still, in that situation it is perfectly possible for him to find out that it is Mary who is knocking, not John.

On the other hand, once he has found out that Mary is knocking on the door, it is excluded that it is John, and then it is quite absurd to say that, as far as he knows, it might be John who is knocking at that door.

Truth, or acceptance, and entailment are defined as follows:

Definition 2.4 (Truth and entailment in update semantics)

- ϕ is true in s , $s \models \phi$, iff $s \subseteq s[\phi]$
- $\phi_1, \dots, \phi_n \models \psi$ iff $\forall M, s: s[\phi_1]_M \dots [\phi_n]_M \models_M \psi$

A formula ϕ is true in s if after updating s with ϕ we still envisage the possibilities we envisaged in state s , i.e., if $s[\phi]$ doesn't contain more information than s ; $\phi_1, \dots, \phi_n$ entail ψ if always, if you update your information with $\phi_1 \dots \phi_n$, in that order, you arrive at a state of information to which update with ψ doesn't add anything more.

2.3 Update semantics and DPL

DPL and *US* are genuinely dynamic systems as can be seen from the fact in none of the two conjunction is commutative. The example $\Diamond\phi \wedge \neg\phi$ is a counterexample to commutativity in *US*, and in *DPL* a counterexample is $\exists x Fx \wedge Gx$. (Think of a little sequence like *A man walks in the park. He whistles.* This means something different than the example *He whistles. A man walks in the park.*)

As Groenendijk and Stokhof observe, the properties of (non-)eliminativity and (non-)distributivity serve to distinguish the two systems from classical theories of interpretation. It is fairly easily shown that a dynamic semantics in which all sentences are interpreted as eliminative and distributive updates is not really dynamic after all (Cf., Groenendijk and Stokhof [1990, p. 57] and van Benthem [1991, p. 137]). However, as we have seen above, *DPL* is non-eliminative, since after interpreting an existentially quantified formula $\exists x\phi$ in a state s , the information s has about the value of x is lost. Furthermore, *US* is non-distributive, since the *might*-operator expresses global properties of an information state which do not need to hold of all singleton subsets of that state.

It is important to notice that it is distinct properties that distinguish *DPL* and *US* from static theories, non-eliminativity and non-distributivity respectively. These two different ways in which *DPL* and *US* depart from the static paradigm are reflected by a difference in the respective notions of truth. For ψ to be true in s in *DPL*, each singleton subset s must allow update with ψ . This notion of truth is that of a distributive system. On the other hand, the *US* notion of truth is that of an eliminative system. For ψ to be true in s , *US* requires that the update of s with ϕ does not eliminate possibilities in s .

From these remarks it may already appear that the two different notions of truth and entailment should not be substituted for one another. For instance, if we adopt the *US* notion of truth in *DPL*, then the *DPL*-valid entailment $\exists x Px \models \exists y Py$ would no longer be valid. The reason is that, on the *US* notion of truth, $\exists y Py$ is true in s iff Py is true in s , and, clearly, $\exists x Px \not\models Py$ on any of the two notions of entailment. On the other hand, if we adopt the (distributive) *DPL* notion of truth in *US*, then the *US*-valid entailment $\Diamond\phi \models \Diamond\phi$ would no longer go through. The

reason is that, on the *DPL* notion of truth, $\Diamond\phi$ is true in s iff ϕ is true in s , and $\Diamond\phi \not\models \phi$ on any of the two notions of entailment.

We see that *DPL* and *US* are two really different systems of dynamic interpretation with conflicting characteristic properties. This is not to say that the two are incompatible though. As Groenendijk and Stokhof suggest, the two systems can be combined within a system that preserves the characteristic features of both and that, to some extent, gives a separate treatment of the two different kinds of (update of) information that the two systems deal with.

However, in this paper I want to show that it is worthwhile to remove one of the differences between the two systems, by adapting the logic of one of the two (*DPL*) to the format of that of the other (*US*). Doing thus, a combination of the two theories may have more the character of an *integration*, since it allows us to employ a singular notion of truth instead of the product of two, so to speak.

Two aspects of information growth

Characteristic feature of *DPL* is that it has a non-eliminative semantics. *DPL*'s non-eliminativity originates from existential quantifiers which effectuate a reinstantiation of variables, and this formalizes the phenomenon that indefinite noun phrases of natural language set up discourse referents for future anaphoric coreference.

However, the reinstantiation of a variable x in a state s at the same time implies the loss of information that s may have about x , and therefore, in *DPL*, the introduction of one discourse referent goes hand in hand with the exit of another. So, in fact, *DPL* fails an account of the fact the indefinites may introduce 'novel' objects, without interfering with (information about) other discourse referents introduced before.

It seems that, apart from information about the values of variables, another aspect of information about variables needs to be taken into account, viz., information about the domain of variables. When processing successive sentences in a discourse, the information that is passed on determines, first, which variables are under discussion, and, second, what the possible values of the variables are. Consequently, update of information need not solely consist of getting better informed about the values of the variables at issue, it may also involve the addition of variables to the universe of discourse. (In fact, in discourse representation theory and file change semantics, these two aspects have been separately encoded.) Clearly, with such a notion of information, the introduction of discourse referents can be conceived of as genuine updates.

In what follows, I will give a reformulation of *DPL* that gives formal expression to the insight that the introduction of discourse referents involves genuine updates. Like *DPL*, the system of interpretation that I propose (*EDPL*) gives a compositional interpretation of the language of predicate logic in terms of updates of information states. However, the information states which are used are more fine-grained than those in *DPL*, since they consist of partial variable assignments. Information states in *EDPL*, thus, determine a domain of variables whose values are at issue, viz.,

the joint domain of the assignments in such states. Besides that, information states contain information about the values of the variables in their domain.

Since information states in *EDPL* model two aspects of information about variables, we can also distinguish two basic kinds of update of information. Update of information consists either of restricting the set of partial variable assignments by elimination, or of extending the domain of partial variable assignments, or, of course, of a mixture of both. The general notion of information update that then results rules a state t a possible update of a state s iff all assignments in t are an extension of an assignment in s . Clearly, under such a notion of update, t is an update of s iff t contains the information that s has about the values of the variables in the domain of s , but t may contain more information about more variables.

It goes without saying that for indefinite noun phrase to extend the domain of a state s they have to be associated with variables which are not in the domain of s . In *DRT*, for this reason, the so-called discourse representation algorithm is used to ensure that indefinites always associate with novel variables and in *FCS* felicity-conditions serve the same purpose. Likewise, also in *EDPL* it is required, on pain of undefinedness, that noun phrases introduce new variables and, consequently, do not mess up with established information about variables in use.

In *EDPL* this is simply achieved by ruling the extension of the domain of a state s with a variable x undefined, if x is already in the domain of s . Consequently, *EDPL* involves partiality in two different respects. In the first place, *EDPL*'s semantics is stated as an update semantics for information states which are sets of partial variable assignments. In the second place, the interpretation function of *EDPL* itself is a partial function. Since the interpretation of a formula $\exists x\phi$ requires an extension of the domain of an information state with the variable x , it is undefined for a state that is already defined for x . Similarly, if a formula contains a free variable x , then the interpretation of the formula is undefined for a state that is undefined for x .

It must be stressed here that the envisaged partiality of *EDPL* has a technical motivation only. *EDPL*'s partial interpretation is not, at least not in the first place, intended to figure in an account of undefinedness phenomena in natural language, such as, for instance, presupposition failure (but, cf., Beaver [1992]). Partial interpretation in *EDPL*, like *DPL*'s non-eliminativity, merely expels nasty side effects of 'infelicit' translation.

3 Update semantics for DPL

In this section, I will first introduce the required notions of information and of information update in *EDPL*, and some operations on information states (section 3.1). In section 3.2 I give the semantics of *EDPL*, discuss some characteristic facts and show some examples. Section 3.3 treats entailment in *EDPL*, undefinedness in *EDPL*, and the relation with *DPL*.

3.1 Information states

If D is our domain of individuals and V the set of variables, then S^X , the set of information states about the values of $X \subseteq V$, and S , the set of all information states, are defined as follows:

Definition 3.1 (Information states)

- $S^X = \mathcal{P}(D^X)$
- $S = \bigcup_{X \subseteq V} S^X$

Given any domain of variables, the notions of minimal and maximal information states are as in *DPL* and *US*. For any domain of variables X , the minimal information state about the values of X is D^X (the state in which all valuations of X are possible). A maximal information state about the values of X is $\{i\}$ for any $i \in D^X$ (the state in which we know all about the values of variables in X). Furthermore, for any domain X , the absurd information state is $\emptyset$ (the information state that excludes every possibility).

A special set of information states is $S^\emptyset$, the possible states of information about the values of no variables. There are only two such states: the set containing the empty assignment, and the empty set. (In fact this is just the domain of truth values on their set-theoretic definition.) So, with respect to the empty domain the minimal and the maximal information state coincide. This reflects the fact that one can have no substantial information about no variables.

The fact that a state t contains more information than a state s , or, in other words, the fact that state t is an update of s is defined as follows:

Definition 3.2 (Update)

- t is an update of s , $t \leq s$, iff $\forall j \in t \exists i \in s: i \leq j$

If t is an update of s , then every assignments in t is an extension of some assignment in s . In that case t contains at least the information that s contains about the variables in the domain of s . Moreover, t may contain information about variables which s is silent about.

The update relation $\leq$ induces a partial order of information states:

Fact 3.1 $\leq$ is a partial order

- reflexive ($s \leq s$)
- transitive (if $s \leq s'$ and $s' \leq s''$ then $s \leq s''$)
- antisymmetric (if $s \leq s'$ and $s' \leq s$ then $s = s'$)

As we will see below, interpretation in *EDPL* is a process of information growth. For any formula ϕ we will find that $s[\phi] \leq s$.

I now define some more helpful notions:

Definition 3.3

- i is a partial element of s , $i \in s$, iff $\exists j \in s: i \leq j$

If i is a partial element of s , I will also say that i survives in s .

Definition 3.4

For all states $s \in S^X, t \in S^Y$, if $X \subseteq Y$:

- $s \sqcap t = \{i \in s \mid i \in t\}$
- $s - t = \{i \in s \mid i \notin t\}$
- $s \sqsubseteq t$ iff $\forall i \in s: i \in t$

Operation $\sqcap$ is an operation labeled 'state restriction'. The restriction of state s by t preserves the elements of s which survive in t , i.e., the assignments in s which have an extension in t . So, $s \sqcap t$ contains the information that s contains supplemented with the information that t contains about the variables in the domain of s . The operation $-$ is labelled 'state subtraction'. Subtracting t from s we preserve the assignments in s which do not survive in t , i.e., we *eliminate* the assignments in s which have an extension in t . So, $s - t$ contains the information that s has about the variables in the domain in s supplemented with the information excluded by t . Finally, $s \sqsubseteq t$, ' s is a substate of t ', iff all possibilities in s survive in t .

The substate relation is used in the definition of *EDPL*-entailment. For s to be a substate of t , s must contain as much information about the values of the variables in the domain of s as t . In other words, s is a subset of the restriction of s by t ($s \sqsubseteq t$ iff $s \subseteq s \sqcap t$ iff $s = s \sqcap t$). Notice that, if s is a substate of t , the latter may contain information about the values of variables which are not in the domain of s .

Like the update relation, the substate relation is a partial order:

Fact 3.2 $\sqsubseteq$ is a partial order

- reflexive ($s \sqsubseteq s$)
- transitive (if $s \sqsubseteq s'$ and $s' \sqsubseteq s''$ then $s \sqsubseteq s''$)
- antisymmetric (if $s \sqsubseteq s'$ and $s' \sqsubseteq s$ then $s = s'$)

We also have some kind of contraposition:

Fact 3.3

- $s - t \sqsubseteq z$ iff $s - z \sqsubseteq t$

So, if we substitute the absurd state for z , we find that $s - t \sqsubseteq \emptyset$ iff $s \sqsubseteq t$.

The last operation on states discussed here is the extension of the domain of a state. Let $i \leq_X j$ iff $i \leq j$ and the domain of j is the join of that of i with X . Then the extension of the domain of a state with a variable x is defined as follows:

Definition 3.5 (Domain extension)

- $\forall s \in S^X: s[x] = \{j \mid \exists i \in s: i \leq_{\{x\}} j\}$ if $x \notin X$; undefined otherwise

Extending the domain of s with x gives a state $s[x]$ that only differs from s in that it is defined for x . About the values of variables for which s is defined the new state $s[x]$ contains precisely the same information as s , and about x it is completely impartial: for each i in s , and for each d in D , there is an extension j of i in $s[x]$ that assigns d to x . What is added is only the information 'that x has a value'.

3.2 Semantics of EDPL

We are now ready to turn to the semantics of *EDPL*. Like a *DPL* model, an *EDPL* model is a pair $M = \langle D, F \rangle$ consisting of a non-empty set of individuals D and an interpretation function F that assigns sets of n -tuples of objects to n -ary relation expressions. (Again, I omit reference to M whenever this does not lead to confusion.) *EDPL* interpretation is defined as a partial update function on information states:

Definition 3.6 (Semantics of EDPL)

- $s[Rx_1 \dots x_n] = \{i \in s \mid \langle i(x_1), \dots, i(x_n) \rangle \in F(R)\}$ if defined
- $s[\neg\phi] = s - s[\phi]$
- $s[\exists x\phi] = s[x][\phi]$
- $s[\phi \wedge \psi] = s[\phi][\psi]$

The clauses in this definition closely correspond to those in the definitions of the semantics of *DPL* and *US*. The interpretation of an atomic formula is the same as in *DPL*. Negation is like in *US*, apart from the fact that we use *state*-, in stead of *set*-, subtraction. Like in *DPL*, the existential quantifier $\exists x$ introduces arbitrary values for x , the difference being that domain extension is used, instead of reinstantiation. Finally, like in *DPL* and *US*, conjunction is interpreted as function composition.

The most significant difference with *DPL* and *US* is that in *EDPL* the interpretation of a formula ϕ can be undefined for certain states. If an atomic formula contains a variable for which a state s is undefined, then the interpretation of the formula is undefined for that state s ; similarly, the interpretation of $\exists x\phi$ is undefined for a state s which is already defined for x . Furthermore, undefinedness persists in the following way. If (the interpretation of) ϕ is undefined for s , then $\neg\phi$ and $\phi \wedge \psi$ are undefined for s . Furthermore, if ϕ is undefined for $s[x]$, then $\exists x\phi$ is undefined for s , and if ψ is undefined for $s[\phi]$, then $\phi \wedge \psi$ is undefined for s . In section 3.3, I will give a more detailed account of the undefinedness phenomena in *EDPL*. Before that, I will assume definedness whenever this is unlikely to give rise to confusion.

Some characteristic facts Like *DPL*, *EDPL* has a distributive semantics. However, unlike *DPL*, *EDPL* is not really non-eliminative, although it is not really eliminative either. Instead of eliminativity, what is relevant here is that *EDPL* interpretation always yields updates:

Fact 3.4 (Update and distributivity)

- $s[\phi] \leq s$ if defined
- $s[\phi] = \bigcup_{i \in s} \{i\}[\phi]$ iff $\forall i \in s: \{i\}[\phi]$ is defined; undefined otherwise

EDPL updates are not purely eliminative, since a state $s[\phi]$ may contain *extensions* of assignments in s . Distributivity holds modulo definedness.

DPL's characteristic donkey equivalences are retained:

Fact 3.5 (Donkey equivalences)

- $(\exists x\phi \wedge \psi) \Leftrightarrow (\exists x(\phi \wedge \psi))$
- $((\phi \wedge \psi) \wedge \chi) \Leftrightarrow (\phi \wedge (\psi \wedge \chi))$

- $(\exists x\phi \rightarrow \psi) \Leftrightarrow (\forall x(\phi \rightarrow \psi))$
- $((\phi \wedge \psi) \rightarrow \chi) \Leftrightarrow (\phi \rightarrow (\psi \rightarrow \chi))$

So, like *DPL*, *EDPL* accounts for the fact that indefinite noun phrases (existential quantifiers) in one sentence may bind pronouns (free variables) in another. Furthermore, an indefinite in the antecedent of an implication gets universal force.

Negation *EDPL* uses a notion of negation that looks more like that of *US* than that of *DPL*. If we interpret $\neg\phi$ in a state s , ϕ is interpreted in s and the result is subtracted from s . Still, this notion of negation is an adequate reproduction of *DPL* negation, since *EDPL* interpretation, besides being distributive, has the update property. Recall that $s[\neg\phi]$ in *DPL* is the set of assignments $i \in s$ such that $\{i\}[\phi] = \emptyset$. In *EDPL*, $i \in s[\neg\phi]$ iff $i \in s$ and no $j \geq i$ is in $s[\phi]$. Now, distributivity and update, this means that $i \in s$ and $\{i\}[\phi] = \emptyset$, like in *DPL*.

Let us look at one example:

- (7) No man sees her
 $\neg\exists y(My \wedge S y x)$

The interpretation of this sentence in a state s is $s[\neg\exists y(My \wedge S y x)]$, which is $s - s[y][My][S y x]$. The state $s[y][My][S y x]$ is the following set of assignments:

- (8) $\{j \mid \exists k \in s: k \leq_{\{y\}} j \text{ \& } j(y) \in F(M) \text{ \& } \langle j(y), j(x) \rangle \in F(S)\}$

So, the state $s - s[y][My][S y x]$ is the following set of assignments:

- (9) $\{i \in s \mid \neg\exists j: i \leq_{\{y\}} j \text{ \& } j(y) \in F(M) \text{ \& } \langle j(y), j(x) \rangle \in F(S)\}$

In other words, the result of interpreting $\neg\exists y(My \wedge S y x)$ in s is the set of assignments in s that assign x an individual for which no other individual can be found that is a man and sees her.

Since the interpretation of a formula $\neg\phi$ in a state s returns those i in s which do not survive in $s[\phi]$, no formula $\neg\phi$ brings about extension of the domain of discourse. As a consequence, the law of double negation doesn't hold in *EDPL*. The double negation of a formula involves state restriction:

Fact 3.6

- $s[\neg\neg\phi] = s - (s - s[\phi]) = s \sqcap s[\phi]$

The double negation of a formula ϕ gives us the restriction of s by the update of s with ϕ . So, $\neg\neg\phi$ imposes the same restrictions as ϕ imposes on the assignments in a state s , but it cancels possible extensions of the domain of s brought about by ϕ . Like Groenendijk and Stokhof, I will call the double negation of ϕ the static closure of ϕ , and write $\downarrow\phi$ for $\neg\neg\phi$.

As usual, the universal quantifier and implication are defined in terms of the existential quantifier, negation and conjunction. Their interpretation is the following:

Fact 3.7

- $s[\forall x\phi] = s[\neg\exists x\neg\phi] = s - (s[x] - s[x][\phi])$
- $s[\phi \rightarrow \psi] = s[\neg(\phi \wedge \neg\psi)] = s - (s[\phi] - s[\phi][\psi])$

$$\bullet s[\forall x(\phi \rightarrow \psi)] = s[\neg \exists x \neg (\phi \wedge \neg \psi)] = s - (s[x][\phi] - s[x][\phi][\psi])$$

The interpretation of the formula $\forall x(\phi \rightarrow \psi)$ in a state s preserves all those assignments $i \in s$ such that every extension of i in $s[x][\phi]$ survives in $s[x][\phi][\psi]$. So, the interpretation of *Every farmer who owns a donkey beats it* in a state s , $s[\forall x((Fx \wedge \exists y(Dy \wedge Oxy)) \rightarrow Bxy)]$, is the set of assignments i in s such that every extension of i in $s[x][Fx][y][Dy][Oxy]$ has an extension in $s[x][Fx][y][Dy][Oxy][Bxy]$. Put more simply, this requires of an i in s that under every valuation of x and y such that x is a farmer and y a donkey that x owns, we find that x beats y .

Digression It has been argued, for instance by Schubert and Pelletier [1988], that the strong readings of donkey sentences are misguided, or, at least, not the only reading these sentences have. Schubert and Pelletier's favourite example is the following sentence, which I label the 'dime implication':

(10) If I have a dime in my pocket, I'll put it in the parking meter.

On its most natural reading this sentence says that if I have a dime in my pocket, then I will throw one in the meter. However, if we interpret the dime implication, like the donkey implication, as one of strong implication, then the sentence would imply that I throw *all* the dimes I have in my pocket in the meter. As concerns the present example, this strong reading seems quite odd.

It is very well possible to define a notion of weak implication that assigns conditional sentences the weak truth conditions that Schubert and Pelletier argue for:

Definition 3.7 (Weak implication)

$$\bullet s[\phi \hookrightarrow \psi] = (s - s[\phi]) \cup (s \sqcap s[\phi][\psi])$$

However, in section 4, we will see that we do not need such an independent notion of weak implication, since it in fact fits in a more general scheme of (universal) adverbial quantification. That concludes the digression.

3.3 Truth, undefinedness and the relation with DPL

Truth and entailment are defined as follows:

Definition 3.8 (Truth and entailment in EDPL)

- ϕ is true in s , $s \models \phi$, iff $s \sqsubseteq s[\phi]$
- ϕ is false in s , $s \models \neg \phi$, iff $s[\phi] = \emptyset$
- $\phi_1, \dots, \phi_n \models_X \psi$ iff $\forall M, s \in S^X: s[\phi_1]_M \dots [\phi_n]_M \models_M \psi$

A formula ϕ is true in a state s if the update of s with ϕ does not add any information about the variables in the domain of s . If a formula is true in a state s , its negation is false in s , and vice versa.

The notion of entailment, which has the dynamics of *DPL* entailment, is defined in terms of the *US*-style notion of truth. A sequence of premisses entails a conclusion if the update of any state of information about a certain domain of variables with the premisses always produces a state of information in which the conclusion is true.

Notice that a valid conclusion need not be true in the update of all states with the premisses. The conclusion of an entailment is only required to be true in the update with the premisses of all states with a certain domain of variables. The reason for this is that we should not completely exclude the possibility of undefinedness. For instance, for states with a domain Y such that $x \notin Y$, the interpretation of $x = x$ is undefined. Nevertheless, on the present definition of entailment $\models_X (x = x)$ is valid, that is, for any domain X such that $x \in X$. Similarly, for all states $s \in S^Y$, if $x \notin Y$, then $\exists y Fy$ is not guaranteed to be defined for $s[\exists x Fx]$. Still, the entailment $\exists x Fx \models_X \exists y Fy$ is valid, for a domain X such that $x, y \in X$.

Notice that a valid inference with respect to a domain of variables X may be *undefined* with respect to states in different domains, but can never be *falsified* in states with different domains. If $\phi_1, \dots, \phi_n$ entail ψ with respect to a domain X , and if for some state $s \notin S^X$ the update with $\phi_1, \dots, \phi_n, \psi$ is defined, then ψ is true in the state $s[\phi_1] \dots [\phi_n]$:

Fact 3.8

- If $\phi \models_X \psi$, then for all s , if $s[\phi][\psi]$ is defined, then $s[\phi] \models \psi$

For this reason I will drop the subscript X whenever irrelevant.

The deduction theorem holds in *EDPL*:

Fact 3.9

- $\Gamma, \phi \models \psi$ iff $\Gamma \models \phi \rightarrow \psi$

Contrary to the substate relation, *EDPL*-entailment is not transitive. Whereas $\exists x Fx$ entails $\exists y Fy$ and $\exists y Fy$ entails Fy , $\exists x Fx$ does not entail Fy .

After having discussed undefinedness in *EDPL*, I will give a restricted version of transitivity which holds of *EDPL* entailment.

Undefinedness Now let us look more specifically at the presuppositions of *EDPL* formulas and their projection behaviour in compound formulas. We have seen that presuppositions are triggered by free variable occurrences and existential quantifiers. A formula containing a free variable x presupposes that a state contains information about the value of x , and an existential quantifier $\exists x$ presupposes that a state is undefined for x . Furthermore, and this is characteristic of presuppositions, the presuppositions of the negation of a formula ϕ are the presuppositions of ϕ .

In the processing of conjunctions falsity persists and undefinedness persists. If $s[\phi]$ returns the absurd information state (or is undefined), then $s[\phi \wedge \psi]$ also returns the absurd state (cq. is undefined). This gives rise to a Karttunen/Heim-style presupposition projection behaviour (Karttunen [1974], Heim [1983]). If $\pi(\phi)$ expresses the presuppositions of ϕ , then, in Karttunen and Heim's systems, $\pi(\neg\phi) = \pi(\phi)$ and $\pi(\phi \wedge \psi) = \pi(\phi \rightarrow \psi) = \pi(\phi) \wedge (\phi \rightarrow \pi(\psi))$.

The presuppositions of *EDPL* formulas depend on the domain of states with respect to which they are interpreted. Therefore, the presuppositions of a formula must be calculated by means of an indexed function π_X that for any formula ϕ gives a formula $\pi_X(\phi)$ which states the presuppositions that ϕ carries concerning states of

information about a domain of variables X . This function is defined as follows (for ease of exposition I only consider atomic formulas with only one variable x):

Definition 3.9

- $\pi_X(Fx) = (x = x)$ if $x \in X$
 $= \forall x(x \neq x)$ if $x \notin X$
- $\pi_X(\exists x\phi) = (x \neq x)$ if $x \in X$
 $= \forall x \pi_{(X \cup \{x\})}(\phi)$ if $x \notin X$
- $\pi_X(\neg\phi) = \pi_X(\phi)$
- $\pi_X(\phi \wedge \psi) = \pi_X(\phi) \wedge (\phi \rightarrow \pi_X(\psi))$ if ϕ is a test

A formula ϕ is a test if ϕ is an atomic formula $Rx_1 \dots x_n$ or a negation $\neg\psi$. The presuppositions of conjunctions with other first conjuncts are computed using the equivalences $(\exists x\phi \wedge \psi) = \exists x(\phi \wedge \psi)$ and $((\phi \wedge \psi) \wedge \chi) = (\phi \wedge (\psi \wedge \chi))$.

Fact 3.10

- $\forall s \in S^X: s[\pi_X(\phi)][\phi]$ is defined

Fact 3.11

- $\forall s \in S^X: s[\phi]$ is defined iff $s \models \pi_X(\phi)$

So, for a state s with domain X , ϕ is defined for s if and only if the formula that expresses ϕ 's presuppositions is true in s .

Using the presupposition function π , we can now establish a restricted version of transitivity in *EDPL*:

Fact 3.12 (Restricted transitivity)

- If $\phi \models_X \psi$ and $\psi \models \chi$, then $\pi_X(\phi \wedge \chi), \phi \models_X \chi$

This fact says that if ϕ entails ψ with respect to X , and ψ entails χ , then ϕ entails χ with respect to X , whenever the presuppositions of $\phi \wedge \chi$ with respect to X are met. This restricted version of transitivity in effect excludes cases where the mediating formula ψ introduces variables which are free in χ .

Let us briefly look at the counterexample to transitivity given above. In *EDPL* we have that $\exists xFx \models_X \exists yFy$, where $x, y \notin X$, and $\exists yFy \models Fy$. Restricted transitivity gives $\pi_X(\exists xFx \wedge Fy), \exists xFx \models_X Fy$.

Computing the presuppositions of the two formulas with respect to X , we obtain $\pi_X(\exists xFx \wedge Fy) = \forall x((x = x) \wedge (Fx \rightarrow \forall y(y \neq y)))$. With respect to a state $s \in S^X$, this equals $\forall x \neg Fx$. So, restricted transitivity tells us that $\forall x \neg Fx, \exists xFx \models_X Fy$, which, clearly, is correct.

DPL and EDPL I now turn to the relation between *EDPL* and *DPL*. The relation between the two systems is partially characterized by the following fact (s^V is the set of total extensions of s , i.e., $s^V = \{g \in D^V \mid \exists i \in s: i \leq g\}$):

Fact 3.13

- $s \models \phi$ in *EDPL* iff $s^V \models \phi$ in *DPL* if defined

In order to complete the picture of the relation between *EDPL* and *DPL*, we only need to combine the facts 3.13 and 3.11. Fact 3.13 relates truth in *EDPL* to truth in *DPL* on the provision of definedness. Fact 3.11 relates definedness in *EDPL* to truth in *EDPL*. The facts 3.13 and 3.11, thus, jointly entail the following fact:

Fact 3.14

- $\forall s \in S^X: s \models \phi$ in *EDPL* iff $s^V \models (\pi_X(\phi) \wedge \phi)$ in *DPL*

4 Quantifiers

In this section I introduce quantifiers in *EDPL* and show that we can give a perspicuous and uniform interpretation of adnominal and adverbial quantifiers, symmetric as well as asymmetric. I start with adverbs of quantification.

4.1 Adverbs of quantification (unselective)

Lewis [1975] argues that in many cases adverbs of quantification (like *always*, *sometimes*, *usually*) unselectively quantify over the values of 'free variables' (or indefinite noun phrases) in their restrictive clause. The examples Lewis discusses are of the following form:

- (11) Sometimes/usually/always if a man owns a donkey, he beats it.

Lewis points out that the quantifying adverbs unselectively quantify over the values of the free variables, cq., indefinites, in the restrictive clause, and this phenomenon has been, quite elegantly, formalized in the frameworks of *FCS*, *DRT* and *DPL*. In *DPL*, for instance, the formula *Always*(ϕ)(ψ) tests, given an initial assignment g , whether all assignments that verify ϕ with respect to g are assignments with respect to which ψ is true.

EDPL, too, allows a straightforward interpretation of unselectively quantifying adverbs. For any adverb of quantification A , with its usual set-theoretic interpretation A' , the interpretation is defined as follows:

Definition 4.1 (Adverbs of quantification (symmetric))

- $s[A(\phi)(\psi)] = \{i \in s \mid A'(\{j \in s[\phi] \mid i \leq j\})(\{j \in s[\phi] \mid \psi\})\}$

So, if we interpret *If a farmer owns a donkey he always beats it* in s , we get $s[Always(\exists x(Fx \wedge \exists y(Dy \wedge Oxy))(Bxy))]$. This is the set of assignments i in s such that on every extension of i to x and y , if the value of x is a farmer who owns a donkey which is the value of y , then the value of x beats the value of y . In other words, this formula tests whether all pairs of a farmer and a donkey he owns are pairs of which the first element beats the second element. A second example is *If a man gives her a present, she usually thanks him for it*, *Usually*($\exists y(My \wedge \exists z(Pz \wedge Gyzx))(Txyz)$). Interpreted in a state s , this example gives all those i in s in return that assign x an individual that renders thanks in most cases in which a man gives her a present.

There are some interesting correspondences between the sentential connectives of *EDPL* and adverbs of quantification ($\downarrow\phi$ abbreviates $\neg\neg\phi$):

Fact 4.1

- $Sometimes(\phi)(\psi) \Leftrightarrow \downarrow(\phi \wedge \psi)$
- $Always(\phi)(\psi) \Leftrightarrow (\phi \rightarrow \psi)$
- $Never(\phi)(\psi) \Leftrightarrow \neg(\phi \wedge \psi)$

So conjunction (disregarding its external dynamics) and implication fit in the more general scheme of adverbial quantification.

4.2 Adverbs of quantification (asymmetric)

Adverbs of quantification not always unselectively quantify over the values of all variables introduced in their restriction. Several authors (Bäuerle and Egli [1985], Root [1986], Rooth [1987] and Kadmon [1987], see also Heim [1990] and Chierchia [1992]) have discussed examples in which adverbial quantifiers seem to involve quantification over the values of a proper *subset* of the introduced variables. Following Rooth and Kadmon, I call this kind of quantification *asymmetric*. We find it in the following sentences:

- (12) If a farmer owns a donkey, he is usually rich.
- (13) If a DRUMMER lives in an apartment complex, it is usually half empty.
- (14) If a drummer lives in an APARTMENT COMPLEX, it is usually half empty.

On its most natural reading, the adverb *usually* in the first example quantifies over farmers who own a donkey and not over farmer-donkey pairs. The sentence says that most farmers who own a donkey are rich. (If the adverb is taken to quantify unselectively, we get the different reading that for most pairs consisting of a farmer and a donkey he owns, it holds that the farmer is rich.) The second and third example are different for a similar reason. In the second example, with focal stress on *drummer*, we (may) find quantification over apartment complexes in which a drummer houses. The example then states that most apartment complexes where a drummer lives are usually half-empty. In the third example, where we find focal stress on *apartment complex*, the adverb may be taken to quantify over drummers. On this reading, the sentence says that most drummers that live in an apartment complex live in an half empty apartment complex.

In *EDPL*, asymmetric adverbs naturally fit into the general scheme of adverbial quantification. Since *EDPL* has the update property, we can (unselectively) quantify over the assignments that satisfy the restriction ϕ of an adverb by considering all extensions of i in $s[\phi]$, for any assignment i in an input state s . Now, in case of asymmetric quantification, we only need to take into account extensions of i that survive in $s[\phi]$, and test whether these extensions also survive in further update with the nuclear scope of the adverb.

So, assume that an asymmetric adverb of quantification comes with a set of selection indices X that selects the variables whose values the adverb quantifies over. Its interpretation then is defined as follows:

Definition 4.2 (Adverbs of quantification (asymmetric))

- $s[A_X(\phi)(\psi)] = \{i \in s \mid A'(\{j \in s[\phi] \mid i \leq_X j\})(\{j \in s[\phi][\psi]\})\}$

Let us briefly consider two mutually related examples.

(15) If a man gives her a PRESENT, she usually thanks him for it

$$Usually_{\{y\}}(\exists y(My \wedge \exists z(Pz \wedge Gyzx)))(Txyz)$$

Interpreted in a state s , this example returns all those i in s that assign x an individual that renders thanks to most men that gives her a present, irrespective of the number of presents given.

(16) If a MAN gives her a present, she usually thanks him for it

$$Usually_{\{z\}}(\exists y(My \wedge \exists z(Pz \wedge Gyzx)))(Txyz)$$

When interpreted in a state s , this example returns all those i in s that assign x an individual that renders thanks for most presents given by a man, irrespective of the number of men that give it.

We find the following equivalences:

Fact 4.2

- $Sometimes_X(\phi)(\psi) \Leftrightarrow Sometimes(\phi)(\psi)$
- $Never_X(\phi)(\psi) \Leftrightarrow Never(\phi)(\psi)$
- $Always_{\emptyset}(\phi)(\psi) \Leftrightarrow (\phi \hookrightarrow \psi)$

(for X a subset of the variables introduced by ϕ)

So, for the adverbs *sometimes* and *never* it makes no difference whether or not they select variables to quantify over. This is as it should be, since there seems to be no evidence whatsoever that there are distinct asymmetric readings of these adverbs.

On the other hand, for the adverbs *usually* and *always*, it does make a difference whether or not they select variables for asymmetric quantification, and which variables they select. Furthermore, we see that the weak implication ($\hookrightarrow$) addressed in the digression of section 3.2, now appears to be a borderline case of asymmetric adverbial quantification, i.e., universal quantification over the values of an empty set of selection indices.

That the weak implication is one of the many forms of asymmetric quantification, may be further substantiated by slightly varying the exemplary dime implication:

(17) If I have a dime in my pocket, I throw it in the parking meter.

$$(\exists y(Dy \wedge Piy) \hookrightarrow Tiy) \Leftrightarrow Always_{\emptyset}(\exists y(Dy \wedge Piy))(Tiy)$$

(18) If a man has a dime in his pocket, he throws it in the parking meter.

$$Always_{\{x\}}(\exists x(Mx \wedge \exists y(Dy \wedge Pxy)))(Txy)$$

On its most natural reading, the first example was argued to state that if I have a dime in my pocket, then I throw one in the meter, and this reading is captured by interpreting the conditional sentence as one of weak implication. However, the second example, which is a minor variation of the first, is most likely interpreted as stating that every man who has a dime, throws one in the meter. Neither the weak, nor the strong, reading of the implication gives us this. The preferred reading of this example is strong, i.e., universal, with respect to the parkers which have a dime in their pocket, and weak with respect to the dimes they throw in the meter. And, in fact, if the sentence is understood as asymmetrically quantifying over the parkers,

as indicated by its translation above, this reading results. Notice, that the two dime sentence, thus, can be interpreted in a uniform way.

4.3 Adnominal quantifiers

Internally dynamic generalized quantifiers, with an interpretation proposed by several authors (Chierchia [1992] to start with, see also van den Berg [1991] and van Eijck and de Vries [1991]) are easily introduced in a way that neatly fits in with the interpretation of quantifying adverbs. Let D be an arbitrary binary quantifier which has D' as its usual set-theoretic interpretation, then:

Definition 4.3 (Binary quantifiers)

- $s[Dx(\phi)(\psi)] = \{i \in s \mid D'(\{j \in s[x][\phi] \mid i \leq_{\{x\}} j\})(\{j \in s[x][\phi][\psi]\})\}$

We find the following correspondences with the unary quantifiers in *EDPL*:

Fact 4.3

- $An x(\phi)(\psi) \Leftrightarrow \downarrow \exists x(\phi \wedge \psi)$
- $No x(\phi)(\psi) \Leftrightarrow \neg \exists x(\phi \wedge \psi)$
- $Every x(\phi)(\psi) \Leftrightarrow \forall x(\phi \hookrightarrow \psi)$

We see that the binary determiners *a(n)* and *no* have the same truth-conditional content as their usual first order paraphrases. (The only difference is that the binary quantifiers defined above are externally static.) We also see that *EDPL* licenses a weak and a strong reading of the quantifier *every*, both of which are intuitively motivated. If we treat it as a binary generalized quantifier, the weak reading results. This reading is appropriate for the sentence *Every man who has a dime puts in the parking meter*. On the other hand, if we translate *every* with the unary first order quantifier as $\forall x(\phi \rightarrow \psi)$, then the strong reading results, and this is the proper reading of the (strong) donkey sentence.

It may be a bit tantalizing that we find two non-equivalent ways to translate natural language *every* in the language of *EDPL*. I must say I don't know of any knock down arguments for taking *every* to be ambiguous or for choosing one interpretation in favour of the other. On the other hand, it is especially the determiner *every* which has been associated with both weak and strong readings in the literature. Now it is not so much a merit of *EDPL* that it accounts for both kinds of readings in a principled way, but it is a relative merit that it does so without entailing ambiguity of all the other quantifiers.

A last observation concerns the relation between determiners and quantifying adverbs. If adverb A and determiner D are set-theoretically the same, then:

Fact 4.4

- $A_{\{x\}}(\exists x\phi)(\psi) \Leftrightarrow Dx(\phi)(\psi)$

So, here we see that the sentence *If a man owns a donkey, he is usually rich* on its asymmetric construal in which *usually* selects donkey owning men for quantification, is equivalent to the sentence *Most men that own a donkey are rich*.

References

- Bäuerle, R. & Egli, U., 1985, *Anapher, Nominalphrase und Eselsätze*, Sonderforschungsbericht 99., 105, Konstanz: Universität Konstanz
- Beaver, D., 1992, 'The kinematics of presupposition', in: Dekker, P. & Stokhof, M. (eds.), *Proceedings of the eighth Amsterdam colloquium, December 17–20, 1991*, Amsterdam: ILLC
- Benthem, J. van, 1991, *Language in action*, Amsterdam: North-Holland
- Berg, M. van den, 1991, 'Dynamic generalized quantifiers', in: Does, J. van der & Eijck, J. van (eds.), *Generalized quantifier theory and applications*, Amsterdam: Dutch Network for Language, Logic and Information, pp. 223–244
- Chierchia, G., 1992, 'Anaphora and dynamic binding', *Linguistics and Philosophy* 15, pp. 111–184
- Eijck, J. van & de Vries, F.-J., 1991, 'Dynamic interpretation and Hoare deduction', in: Does, J. van der & Eijck, J. van (eds.), *Generalized quantifier theory and applications*, Amsterdam: Dutch Network for Language, Logic and Information, pp. 85–113
- Groenendijk, J. & Stokhof, M., 1990, 'Two theories of dynamic semantics', in: Eijck, J. van (ed.), *Logics in AI, European Workshop JELIA '90*, Berlin: Springer Verlag, pp. 55–64
- Groenendijk, J. & Stokhof, M., 1991, 'Dynamic predicate logic', *Linguistics and Philosophy* 14, pp. 39–100
- Heim, I., 1982, *The semantics of definite and indefinite noun phrases*, dissertation, University of Massachusetts, Amherst
- Heim, I., 1983, 'On the projection problem for presuppositions', in: *Proceedings of the West Coast Conference on Formal Linguistics* 2, pp. 114–126
- Heim, I., 1990, 'E-type pronouns and donkey anaphora', *Linguistics and Philosophy* 13, pp. 137–178
- Kadmon, N., 1987, *On unique and non-unique reference and asymmetric quantification*, dissertation, University of Massachusetts, Amherst
- Kamp, H., 1981, 'A theory of truth and semantic representation', in: Groenendijk, J., Janssen, T. & Stokhof, M. (eds.), *Formal methods in the study of language*, Amsterdam: Mathematical Centre, pp. 277–322 [reprinted in: Groenendijk, J., Janssen, T. & Stokhof, M. (eds.), 1984, *Truth, interpretation and information*, Dordrecht: Foris, pp. 2–41]
- Karttunen, L., 1974, 'Presupposition and linguistic context', *Theoretical Linguistics* 1, pp. 181–194
- Lewis, D., 1975, 'Adverbs of quantification', in: Keenan, E. (ed.), *Formal semantics of natural language*, Cambridge: UP
- Pelletier, F.J. & Schubert, L., 1988, 'Generically speaking', in: Chierchia, G., Partee, B. & Turner, R. (eds.), *Properties, types and meaning II*, Dordrecht: Reidel, pp. 193–268
- Root, R., 1986, *The semantics of anaphora in discourse*, dissertation, University of Texas, Austin

- Rooth, M., 1987, 'Noun phrase interpretation in Montague grammar, file change semantics, and situation semantics', in: Gärdenfors, P. (ed.), *Generalized quantifiers*, Dordrecht: Reidel, pp. 237-269
- Stalnaker, R., 1974, 'Pragmatic presuppositions', in: Munitz, M. & Unger, P. (eds.), *Semantics and philosophy*, New York: University Press, pp. 197-213
- Veltman, F., 1990, 'Defaults in update semantics', in: Kamp, H. (ed.), *Conditionals, defaults and belief revision*, DYANA report R2.5.A, Edinburgh: Centre for Cognitive Science, University of Edinburgh, pp. 28-63

A Sound and Complete Calculus for Update Logic

Jan van Eijck (CWI, Amsterdam & OTS, Utrecht)

Fer-Jan de Vries (CWI, Amsterdam)

Abstract

This paper presents a sound and complete deduction system for Veltman's update logic in the style of the Hoare calculus with two kinds of correctness statements for dynamic assignment logic in Van Eijck and De Vries [13]. The Hoare correctness statements use modal propositional logic as assertion language and connect update logic to the modal propositional logic S5. The connection with modal propositional logic provides a clear link between the dynamic and the static semantics of update logic. The fact that update logic is decidable was noted already in [10]; the connection with S5 provides an alternative proof. The S5 connection can also be used for rephrasing the validity notions of update logic and for performing consistency checks.

1 Introduction

Recently the semantics of natural language has witnessed a shift of emphasis from static meaning (truth conditions) to dynamic meaning (information processing). In dynamic semantics (see for example Karttunen [7], Stalnaker [9], Kamp [6], Heim [4] and Barwise [1]) one thinks of the meaning of a text as the change it brings about in the information state of anyone who accepts the information conveyed by it. One perspective on dynamic semantics for natural language is to view this approach as a proposal to represent the meanings of natural language sentences not by means of formulae from static logic but by means of expressions from a dynamic action language.

The move from static logic to dynamic logic raises some interesting questions. In the first place, due to this transition we seem to have lost the deduction systems that static logical languages carry with them. This was noted by Barwise in [1], one of the first papers to propose an action language as representation medium for natural language meaning. Here the quest for a complete set of axioms for the dynamic inference notion that is engendered by the action language is put forward as an open problem. Next it can be asked what is the precise relation between the static semantics and the dynamic semantics of natural language. Are there systematic ways to derive the static meaning of a sentence from its dynamic representation?

These questions are intimately connected. Our contention is that they should be tackled together and moreover that theoretical computer science can guide the way. In computer science the view that the meaning of a program is a function from information states to information states is common ground. In the case of imperative programming languages this perspective has led to Hoare logic [5] as a successful means to construct deductive systems for reasoning about imperative programs. In this paper we will study Veltman's [14] update logic from this perspective.

To end this introduction, here is an overview of the contents and structure of the paper. In Section 2 Veltman's update logic is presented. Section 3 consists of a brief review of the tools from propositional modal logic that we will need. Section 4 contains the semantic definitions of the basic concepts of the paper that will serve as the starting point for our adaptation of Hoare's logic to update semantics. This section discusses two kinds of Hoare correctness statements with the two kinds of weakest preconditions that they give rise to. Section 5 links weakest preconditions to the next state conditions for update logic that were proposed by Van Benthem [10]. Section 6 contains the Hoare calculus engendered by the notion of weakest preconditions of update programs. In Section 7 we prove the soundness and in Section 8 the completeness of the Hoare calculus. In Section 9 we illustrate how the weakest preconditions analysis and the link to modal propositional logic can be used for reasoning about update logic. Finally, in Section 10, we demonstrate how weakest preconditions and the Hoare calculus that is based on them can be used for reasoning about consistency of update programs.

2 Update Logic: Syntax and Semantics

The characteristic feature of Veltman's update logic (see Veltman [14]) is the epistemic modal operator **might**. Due to the presence of this operator the meanings of 'update programs' have to be phrased in terms of input information *sets*, and have to be phrased dynamically. A program π maps an information state I to a new information state $\llbracket \pi \rrbracket(I)$. To see that **might** is the key feature, note that the semantics for the fragment of update logic without **might** can be given by means of a dynamic yes/no function for individual propositional valuations, which reduces the semantics immediately to ordinary static propositional logic.

Following Veltman [14] (and in fact, slightly simplifying his syntax), we can define the language of epistemic update logic over a set of proposition letters P as the smallest set L_P such that the following hold:

Definition 1 (Syntax of Update Logic L_P)

1. $\perp \in L_P$.
2. If $p \in P$ then $p \in L_P$.
3. If π and $\pi' \in L_P$, then $(\pi; \pi') \in L_P, (\pi \cup \pi') \in L_P$.
4. If $\pi \in L_P$, then $\neg\pi \in L_P, \text{might } \pi \in L_P$.
5. Nothing else is in L_P .

The semantics of L_P is given in terms of input-output behaviour. We take the set W of worlds over P to be the set $\mathcal{P}P$. Any subset of W is an information state. Programs are interpreted as functions from information states to information states, i.e., as functions in $\mathcal{P}W \rightarrow \mathcal{P}W$. The clauses are as follows:

Definition 2 (Semantics of Update Logic)

1. $\llbracket \perp \rrbracket(I) = \emptyset$.
2. $\llbracket p \rrbracket(I) = I \cap \{w \mid p \in w\}$.
3. $\llbracket \pi; \pi' \rrbracket(I) = \llbracket \pi' \rrbracket(\llbracket \pi \rrbracket(I))$.
4. $\llbracket \pi \cup \pi' \rrbracket(I) = \llbracket \pi \rrbracket(I) \cup \llbracket \pi' \rrbracket(I)$.
5. $\llbracket \neg \pi \rrbracket(I) = I - \llbracket \pi \rrbracket(I)$.
6. $\llbracket \text{might } \pi \rrbracket(I) = \begin{cases} I & \text{if } \llbracket \pi \rrbracket(I) \neq \emptyset, \\ \emptyset & \text{otherwise.} \end{cases}$

We will follow the usual conventions and drop outermost parentheses as much as possible. Also, since sequential composition is associative we will write both $\pi_1; (\pi_2; \pi_3)$ and $(\pi_1; \pi_2); \pi_3$ as $\pi_1; \pi_2; \pi_3$.

Intuitively, a program of the form **might** π does not provide information about the world but about available information. A program **might** π is acceptable, given an information state I , if there is at least one world $w \in I$ for which π is accepted in the sense that $w \in \llbracket \pi \rrbracket(I)$. If such a w can be found, the output information state of **might** π is equal to its input information state; this agrees with the intuition that **might** π does not say anything at all about what the world is like. In the other case, i.e., the case were $\llbracket \pi \rrbracket(I) = \emptyset$, the output information state of **might** π equals $\emptyset$.

As was mentioned already, the **might** operator is the key feature of update logic. Yet another way to see this is to note that the semantic clause for **might** π introduces an element of *non distributivity* (this terminology is taken from [3]) into the semantics, in the sense that unions of input states do not distribute over output states: (1) does not in general hold.

$$(1) \quad \llbracket \pi \rrbracket(I) = \bigcup_{i \in I} \llbracket \pi \rrbracket(\{i\}).$$

More specifically, it does not in general hold that $\llbracket \pi \rrbracket(I) \subseteq \bigcup_{i \in I} \llbracket \pi \rrbracket(\{i\})$. Counterexample: take π equal to **might** p and let $I = \{w, w'\}$ with $p \in w$ and $p \notin w'$. Then $\llbracket \text{might } p \rrbracket(\{w\}) \cup \llbracket \text{might } p \rrbracket(\{w'\}) = \{w\}$, but $\llbracket \text{might } p \rrbracket(I) = I = \{w, w'\}$.

On the other hand, a simple induction on the complexity of π shows that Lemma 1 holds for all $\pi \in L_P$ and all information states I . In the terminology of Groenendijk & Stokhof [3]: epistemic update logic is *eliminative*.

Lemma 1 (Elimination Lemma) *For all I : $\llbracket \pi \rrbracket(I) \subseteq I$.*

3 Modal Logic

The presence of the modal **might** operator in a dynamic setting which is otherwise fully propositional strongly suggests the use a modal propositional logic as language to make static assertions about update programs in, i.e., as assertion language. The syntax of our assertion language mlp is as follows:

Definition 3 (Syntax of ml_P)

1. $\perp \in ml_P$.
2. If $p \in P$ then $p \in ml_P$.
3. If $\varphi, \psi \in ml_P$, then $(\varphi \wedge \psi), \neg\varphi, \Box\varphi \in ml_P$.
4. Nothing else is in ml_P .

As is customary, we abbreviate $\neg\perp$ as $\top$, $\neg(\neg\varphi \wedge \neg\psi)$ as $(\varphi \vee \psi)$, $\neg(\varphi \wedge \neg\psi)$ as $(\varphi \rightarrow \psi)$, and $\neg\Box\neg\varphi$ as $\Diamond\varphi$. Also, we omit outermost parentheses for readability.

We consider information states $I \in \mathcal{PW}$ as universal Kripke models; thus, I is considered as the Kripke model with accessibility relation $I \times I$. Recall from the literature (see e.g. [2]) that the modal logic determined by the class of finite universal frames is S5. Moreover, for any finite universal model (a universal frame with valuations assigned to all of its worlds) there is a finite subset I of W validating the same formulae. I can be got by throwing away the extra copies of the worlds with identical valuations: because of the universal accessibility this makes no difference to validity.

The notion $I, w \Vdash \varphi$ (world w forces formula φ in information state I) is defined in the usual way. It is convenient to also define the interpretation of a formula with respect to an information state.

Definition 4 (Interpreting φ with respect to I) $[\varphi]_I = \{w \in I : I, w \Vdash \varphi\}$.

In Hoare style reasoning about update logic the notions of relativisation and localisation of modal formulae play an important role. Localisations of modal formulae are defined in Kracht [8]. If φ, ψ are in ml_P , then $\varphi \downarrow \psi$, the localisation of φ to ψ , is given by the following definition.

Definition 5 (Localised modal formulae $\varphi \downarrow \psi$)

$$\begin{aligned}
 \perp \downarrow \psi &= \perp \\
 p \downarrow \psi &= p \wedge \psi \\
 (\varphi_1 \wedge \varphi_2) \downarrow \psi &= (\varphi_1 \downarrow \psi) \wedge (\varphi_2 \downarrow \psi) \\
 (\neg\varphi) \downarrow \psi &= \psi \wedge \neg(\varphi \downarrow \psi) \\
 (\Box\varphi) \downarrow \psi &= \psi \wedge \Box(\psi \rightarrow (\varphi \downarrow \psi))
 \end{aligned}$$

Localisation is closely related to the usual notion of a relativised modal formula.

Definition 6 (Relativised modal formulae φ^ψ)

$$\begin{aligned}
 \perp^\psi &= \perp \\
 p^\psi &= \psi \rightarrow p \\
 (\varphi_1 \wedge \varphi_2)^\psi &= \varphi_1^\psi \wedge \varphi_2^\psi \\
 (\neg\varphi)^\psi &= \neg(\varphi^\psi) \\
 (\Box\varphi)^\psi &= \Box(\psi \rightarrow \varphi^\psi)
 \end{aligned}$$

The connection between the two notions is given by the following lemma.

Lemma 2 (Van Benthem) $\varphi \downarrow \psi$ iff $\varphi^\psi \wedge \psi$.

Proof: Induction on the structure of φ . ■

Given this connection, the following lemma will not come as a surprise (the first two items are from the modal folklore, the third and the fourth from Kracht [8]).

Lemma 3 (Relativisation and Localisation)

1. $I, w \vdash \varphi^\psi$ iff $I, w \vdash \psi$ implies $\{u \in I : I, u \vdash \psi\}, w \vdash \varphi$.
2. $[\varphi^\psi]_I = (I - [\psi]_I) \cup [\varphi]_{[\psi]_I}$.
3. $I, w \vdash \varphi \downarrow \psi$ iff $I, w \vdash \psi$ and $\{u \in I : I, u \vdash \psi\}, w \vdash \varphi$.
4. $[\varphi \downarrow \psi]_I = [\varphi]_{[\psi]_I}$.

Proof: First note that 2 is a reformulation of 1 and 4 a reformulation of 3. Both 1 and 3 are proved by induction on the complexity of φ . ■

4 Correctness Statements for Update Logic

As in the calculus for dynamic predicate logic presented in Van Eijck and De Vries [13], the deductive system for update logic is a hybrid calculus, with statements characterising information states (we call these: statements of the assertion language), plus two kinds of correctness statements, which we call *disjunctive* and *conjunctive* correctness statements. Because of the presence of a modal operator **might** in the language, we need modal operators in the assertion language. The following clauses define the language of H_P of Hoare statements over P .

Definition 7 (Hoare Logic over P : Syntax of H_P)

1. If φ is a formula of ml_P , then φ is a formula of H_P .
2. If φ, ψ are formulae of ml_P and π is a program of L_P , then $\langle \varphi \rangle \pi \langle \psi \rangle$ and $\{\varphi\} \pi \{\psi\}$ are formulae of H_P .
3. Nothing else is a formula of H_P .

The H_P statements of the form φ are used for making assertions about information states I (viewed as universal Kripke models). The H_P statements of the form $\{\varphi\} \pi \{\psi\}$ are *disjunctive* correctness statements. The statement $\{\varphi\} \pi \{\psi\}$ expresses that all information states I and all $w \in W$ such that $I, w \vdash \varphi$ have the property that either $w \notin [\pi](I)$ or $[\pi](I), w \vdash \psi$. The H_P statements of the form $\langle \varphi \rangle \pi \langle \psi \rangle$ are *conjunctive* correctness statements. The statement $\langle \varphi \rangle \pi \langle \psi \rangle$ expresses that all input information states I and all $w \in I$ such that $I, w \vdash \varphi$ have the property that $w \in [\pi](I)$ and $I, w \vdash \psi$. Formally:

Definition 8 (Validity of H_P formulae)

1. If F has the form $\varphi \in mlp$ then $\models F$ if for all $I \subseteq W$ and all $w \in I$ it holds that $I, w \models F$.
2. If F has the form $\{\varphi\} \pi \{\psi\}$ then $\models F$ if for all $I \subseteq W$ and all $w \in I$ such that $I, w \models \varphi$ it holds that $w \notin [\pi](I)$ or $[\pi](I), w \models \psi$.
3. If F has the form $\langle \varphi \rangle \pi \langle \psi \rangle$ then $\models F$ if for all $I \subseteq W$ and all $w \in I$ such that $I, w \models \varphi$ it holds that $w \in [\pi](I)$ and $[\pi](I), w \models \psi$.

The next lemma gives a useful reformulation.

Lemma 4

1. $\models \{\varphi\} \pi \{\psi\}$ iff for all I : $[\varphi]_I \subseteq (I - [\pi](I)) \cup [\psi]_{[\pi](I)}$.
2. $\models \langle \varphi \rangle \pi \langle \psi \rangle$ iff for all I : $[\varphi]_I \subseteq [\psi]_{[\pi](I)}$.

Proof: Immediate from Definition 4, Definition 8 and Lemma 1. ■

Next, let us explain why we need both kinds of correctness statements in our calculus. Suppose we want to express in Hoare logic that for all I , $[\varphi]_I$ is equal to $[\pi](I)$. Then the conjunction of $\models \langle \varphi \rangle \pi \langle \top \rangle$ and $\models \{\neg \varphi\} \pi \{\perp\}$ does the job, witness the following lemma.

Lemma 5 $\models \langle \varphi \rangle \pi \langle \top \rangle$ and $\models \{\neg \varphi\} \pi \{\perp\}$ iff for all I : $[\varphi]_I = [\pi](I)$.

Proof: Note the following equivalences:

$$\begin{aligned} \models \langle \varphi \rangle \pi \langle \top \rangle & \text{ iff } \text{Lemma 4} & \text{ for all } I: [\varphi]_I \subseteq [\top]_{[\pi](I)} \\ & \text{ iff } []_I \text{ def} & \text{ for all } I: [\varphi]_I \subseteq [\pi](I). \end{aligned}$$

Also:

$$\begin{aligned} \models \{\neg \varphi\} \pi \{\perp\} & \text{ iff } \text{Lemma 4} & \text{ for all } I: [\neg \varphi]_I \subseteq (I - [\pi](I)) \cup [\perp]_I \\ & \text{ iff } []_I \text{ def} & \text{ for all } I: I - [\varphi]_I \subseteq I - [\pi](I) \\ & \text{ iff } \text{Lemma 1} & \text{ for all } I: [\varphi]_I \supseteq [\pi](I). \end{aligned}$$

Combine these results and the lemma follows. ■

Corresponding with the two kinds of correctness statements we have two notions of weakest precondition. We will call them respectively weakest conjunctive precondition and weakest disjunctive precondition.

Definition 9 A formula $\varphi \in mlp$ is a weakest disjunctive precondition (WDP) of the program $\pi \in L_P$ and the formula $\psi \in mlp$ if for all I and all $w \in I$:

$$I, w \models \varphi \text{ iff } w \notin [\pi](I) \text{ or } [\pi](I), w \models \psi.$$

A formula $\varphi \in mlp$ is a weakest conjunctive precondition (WCP) of the program $\pi \in L_P$ and the formula $\psi \in mlp$ if for all I and all $w \in I$:

$$I, w \models \varphi \text{ iff } w \in [\pi](I) \text{ and } [\pi](I), w \models \psi.$$

The following lemma gives a useful reformulation of this definition.

Lemma 6

1. φ is a WDP of π and ψ iff for all I : $[\varphi]_I = (I - [\pi](I)) \cup [\psi]_{[\pi](I)}$.
2. φ is a WCP of π and ψ iff for all I : $[\varphi]_I = [\psi]_{[\pi](I)}$.

Proof: Immediate from Lemma 4 and Definition 9. ■

We can say a bit more about the relation between WCP and WDP.

Lemma 7 φ is a WCP of π and ψ iff $\neg\varphi$ is a WDP of π and $\neg\psi$.

Proof:

- | | | |
|-----|--|--|
| | φ is a WCP of π and ψ | |
| iff | Lemma 6 | for all I : $[\varphi]_I = [\psi]_{[\pi](I)}$ |
| iff | | for all I : $I - [\varphi]_I = I - [\psi]_{[\pi](I)}$ |
| iff | Lemma 1 | for all I : $I - [\varphi]_I = (I - [\pi](I)) \cup ([\pi](I) - [\psi]_{[\pi](I)})$ |
| iff | $[\]_I$ def | for all I : $[\neg\varphi]_I = (I - [\pi](I)) \cup [\neg\psi]_{[\pi](I)}$ |
| iff | Lemma 6 | $\neg\varphi$ is a WDP of π and $\neg\psi$. |
-

It is not obvious at first sight that WDPs and WCPs of an L_P program π and an ml_P formula ψ always exist (as formulae of ml_P). We will demonstrate now that they do, by first inductively defining a function $\mathbf{wcp}(\pi, \psi)$, of which we will show that it expresses a WCP of π and ψ , and then defining a function $\mathbf{wdp}(\pi, \psi)$ in terms of $\mathbf{wcp}(\pi, \psi)$ which expresses a WDP of π and ψ .

Definition 10 ($\mathbf{wcp}$)

1. $\mathbf{wcp}(\perp, \psi) = \perp$.
2. $\mathbf{wcp}(p, \psi) = \psi \downarrow p$.
3. $\mathbf{wcp}(\pi_1; \pi_2, \psi) = \mathbf{wcp}(\pi_1, \mathbf{wcp}(\pi_2, \psi))$.
4. $\mathbf{wcp}(\pi_1 \cup \pi_2, \psi) = \psi \downarrow (\mathbf{wcp}(\pi_1, \top) \vee \mathbf{wcp}(\pi_2, \top))$.
5. $\mathbf{wcp}(\neg\pi, \psi) = \psi \downarrow \neg\mathbf{wcp}(\pi, \top)$.
6. $\mathbf{wcp}(\mathbf{might} \pi, \psi) = \Diamond \mathbf{wcp}(\pi, \top) \wedge \psi$.

Lemma 8 shows that the function $\mathbf{wcp}(\pi, \psi)$ does indeed express a WCP of a program π and a modal propositional formula ψ .

Lemma 8 ($\mathbf{wcp}$ adequacy) $[\mathbf{wcp}(\pi, \psi)]_I = [\psi]_{[\pi](I)}$.

Proof: We prove the claim with induction on the structure of π .

$$\begin{aligned}
[\mathbf{wcp}(\perp, \psi)]_I &= \text{wcp def} \quad [\perp]_I \\
&= [\]_I \text{ def} \quad \emptyset \\
&= [\] \text{ def} \quad [\psi]_{[\perp](I)}.
\end{aligned}$$

$$\begin{aligned}
[\mathbf{wcp}(p, \psi)]_I &= \text{wcp def} \quad [\psi \downarrow p]_I \\
&= \text{loc lemma} \quad [\psi]_{[p]_I} \\
&= [\] \text{ def} \quad [\psi]_{[p](I)}.
\end{aligned}$$

$$\begin{aligned}
[\mathbf{wcp}(\pi_1; \pi_2, \psi)]_I &= \text{wcp def} \quad [\mathbf{wcp}(\pi_1, \mathbf{wcp}(\pi_2, \psi))]_I \\
&= \text{ind hyp} \quad [\mathbf{wcp}(\pi_2, \psi)]_{[\pi_1](I)} \\
&= \text{ind hyp} \quad [\psi]_{[\pi_2]([\pi_1](I))} \\
&= [\] \text{ def} \quad [\psi]_{[\pi_1; \pi_2](I)}.
\end{aligned}$$

$$\begin{aligned}
[\mathbf{wcp}(\pi_1 \cup \pi_2, \psi)]_I &= \text{wcp def} \quad [\psi \downarrow (\mathbf{wcp}(\pi_1, \top) \vee \mathbf{wcp}(\pi_2, \top))]_I \\
&= \text{loc lemma} \quad [\psi]_{[\mathbf{wcp}(\pi_1, \top) \vee \mathbf{wcp}(\pi_2, \top)]_I} \\
&= \text{ind hyp}, [\]_I \text{ def} \quad [\psi]_{[\pi_1](I) \cup [\pi_2](I)} \\
&= [\] \text{ def} \quad [\psi]_{[\pi_1 \cup \pi_2](I)}.
\end{aligned}$$

$$\begin{aligned}
[\mathbf{wcp}(\neg\pi, \psi)]_I &= \text{wcp def} \quad [\psi \downarrow \neg\mathbf{wcp}(\pi, \top)]_I \\
&= \text{loc lemma} \quad [\psi]_{[\neg\mathbf{wcp}(\pi, \top)]_I} \\
&= [\]_I \text{ def} \quad [\psi]_{I - [\mathbf{wcp}(\pi, \top)]_I} \\
&= \text{ind hyp}, [\]_I \text{ def} \quad [\psi]_{I - [\pi](I)} \\
&= [\] \text{ def} \quad [[\psi]_{[\neg\pi](I)}].
\end{aligned}$$

$$\begin{aligned}
[\mathbf{wcp}(\text{might } \pi, \psi)]_I &= \text{wcp def} \quad [\Diamond \mathbf{wcp}(\pi, \top) \wedge \psi]_I \\
&= [\]_I \text{ def} \quad [\Diamond \mathbf{wcp}(\pi, \top)]_I \cap [\psi]_I \\
&= \Diamond \text{ def in } S5 \quad \begin{cases} [\psi]_I & \text{if } [\mathbf{wcp}(\pi, \top)]_I \neq \emptyset, \\ \emptyset & \text{otherwise} \end{cases} \\
&= \text{ind hyp}, [\]_I \text{ def} \quad \begin{cases} [\psi]_I & \text{if } [\pi](I) \neq \emptyset, \\ \emptyset & \text{otherwise} \end{cases} \\
&= [\] \text{ def} \quad [\psi]_{[\text{might } \pi](I)}.
\end{aligned}$$

This completes the proof of the lemma. ■

Definition 11 $\mathbf{wdp}(\pi, \psi) \stackrel{\text{def}}{=} \neg\mathbf{wcp}(\pi, \neg\psi)$.

Lemma 9 (wdp adequacy) $[\mathbf{wdp}(\pi, \psi)]_I = (I - [\pi](I)) \cup [\psi]_{[\pi](I)}$.

Proof: Immediate from Lemma 7 and the **wcp** adequacy lemma. ■

Lemma 10

1. $[\mathbf{wcp}(\pi, \top)]_I = [\pi](I)$.
2. $[\mathbf{wdp}(\pi, \perp)]_I = I - [\pi](I)$.
3. $\models \langle \varphi \rangle \pi \langle \top \rangle$ and $\models \{\neg\varphi\} \pi \{\perp\}$ iff for all I : $[\mathbf{wcp}(\pi, \top)]_I = [\varphi]_I$.

Proof:

The first item:

$$\begin{aligned} [\mathbf{wcp}(\pi, \top)]_I &= \mathbf{wcp} \text{ adeq} \quad [\top]_{[\pi](I)} \\ &= [\]_I \text{ def} \quad [\pi](I). \end{aligned}$$

The second item:

$$\begin{aligned} [\mathbf{wdp}(\pi, \perp)]_I &= \mathbf{wdp} \text{ def} \quad [\neg \mathbf{wcp}(\pi, \top)]_I \\ &= [\]_I \text{ def} \quad I - [\mathbf{wcp}(\pi, \top)]_I \\ &= \text{previous item} \quad I - [\pi](I). \end{aligned}$$

The third item: immediate from Lemma 5 and the first item. ■

Lemma 11 $[\mathbf{wcp}(\pi, \psi)]_I = [\psi \downarrow \mathbf{wcp}(\pi, \top)]_I$.

Proof:

$$\begin{aligned} [\mathbf{wcp}(\pi, \psi)]_I &= \mathbf{wcp} \text{ adeq} \quad [\psi]_{[\pi](I)} \\ &= \text{Lemma 10} \quad [\psi]_{[\mathbf{wcp}(\pi, \top)]_I} \\ &= \text{loc lemma} \quad [\psi \downarrow \mathbf{wcp}(\pi, \top)]_I. \end{aligned}$$
■

5 Weakest Preconditions and Next State Conditions

In [10] and [11], Van Benthem has studied update logic by looking at update programs π as functions of the form $\lambda I \cdot \text{NEXT STATE}(I, \pi)$, where NEXT STATE is the function producing the information state which results from processing π in information state I , i.e., π is considered as $\lambda I \cdot [\pi](I)$. The investigation in [10, 11] was carried out in semantic terms, without reference to a specific assertion language, but it can easily be transposed in a setting of assertions from modal propositional logic. Some illuminating conversations between Johan van Benthem and the authors, backed up by an exchange of letters of explanation and consecutive drafts of the present paper, have fully cleared up the connection between his perspective and ours. His generous help in clarifying the issues raised in this section is herewith gratefully acknowledged.

Definition 12 A formula $\psi \in \text{mlp}$ is a next state condition (NSC) of the formula $\varphi \in \text{mlp}$ and the program $\pi \in L_P$ if for all I and all $w \in I$:

$$I, w \vdash \psi \text{ iff } w \in [\pi]([\varphi]_I).$$

Again, it is helpful to reformulate this. Lemma 12 follows immediately from the definition.

Lemma 12 ψ is a NSC of φ and π iff for all I : $[\psi]_I = [\pi]([\varphi]_I)$.

The following function nsc is a reformulation in modal logic of Van Benthem's characterisation of the next state function.

Definition 13 (nsc)

1. $\text{nsc}(\varphi, \perp) = \perp$.
2. $\text{nsc}(\varphi, p) = p \wedge \varphi$.
3. $\text{nsc}(\varphi, \pi_1; \pi_2) = \text{nsc}(\text{nsc}(\varphi, \pi_1), \pi_2)$.
4. $\text{nsc}(\varphi, \pi_1 \cup \pi_2) = \text{nsc}(\varphi, \pi_1) \vee \text{nsc}(\varphi, \pi_2)$.
5. $\text{nsc}(\varphi, \neg\pi) = \neg\text{nsc}(\varphi, \pi) \wedge \varphi$.
6. $\text{nsc}(\varphi, \text{might } \pi) = \Diamond \text{nsc}(\varphi, \pi) \wedge \varphi$.

Lemma 13 (nsc adequacy) $[\text{nsc}(\varphi, \pi)]_I = \llbracket \pi \rrbracket([\varphi]_I)$.

Induction on the structure of π .

$$\begin{aligned} [\text{nsc}(\varphi, \perp)]_I &= \text{nsc def } [\perp]_I \\ &= [\]_I \text{ def } \emptyset \\ &= \llbracket \] \text{ def } \llbracket \perp \rrbracket([\varphi]_I). \end{aligned}$$

$$\begin{aligned} [\text{nsc}(\varphi, p)]_I &= \text{nsc def } [p \wedge \varphi]_I \\ &= [\]_I \text{ def } [p]_I \cap [\varphi]_I \\ &= \llbracket \] \text{ def } [p]([\varphi]_I). \end{aligned}$$

$$\begin{aligned} [\text{nsc}(\varphi, \pi_1; \pi_2)]_I &= \text{nsc def } [\text{nsc}(\text{nsc}(\varphi, \pi_1), \pi_2)]_I \\ &= \text{ind hyp } \llbracket \pi_2 \rrbracket([\text{nsc}(\varphi, \pi_1)]_I) \\ &= \text{ind hyp } \llbracket \pi_2 \rrbracket(\llbracket \pi_1 \rrbracket([\varphi]_I)) \\ &= \llbracket \] \text{ def } \llbracket \pi_1; \pi_2 \rrbracket([\varphi]_I). \end{aligned}$$

$$\begin{aligned} [\text{nsc}(\varphi, \pi_1 \cup \pi_2)]_I &= \text{nsc def } [\text{nsc}(\varphi, \pi_1) \vee \text{nsc}(\varphi, \pi_2)]_I \\ &= \text{ind hyp } \llbracket \pi_1 \rrbracket([\varphi]_I) \cup \llbracket \pi_2 \rrbracket([\varphi]_I) \\ &= \llbracket \] \text{ def } \llbracket \pi_1 \cup \pi_2 \rrbracket([\varphi]_I). \end{aligned}$$

$$\begin{aligned} [\text{nsc}(\varphi, \neg\pi)]_I &= \text{nsc def } [\neg\text{nsc}(\varphi, \pi) \wedge \varphi]_I \\ &= [\]_I \text{ def } (I - [\text{nsc}(\varphi, \pi)]_I) \cap [\varphi]_I \\ &= \text{ind hyp } (I - \llbracket \pi \rrbracket([\varphi]_I)) \cap [\varphi]_I \\ &= \text{elim lemma } ([\varphi]_I - \llbracket \pi \rrbracket([\varphi]_I)) \cap [\varphi]_I \\ &= \llbracket \] \text{ def } \llbracket \neg\pi \rrbracket([\varphi]_I). \end{aligned}$$

$$\begin{aligned} [\text{nsc}(\varphi, \text{might } \pi)]_I &= \text{nsc def } [\Diamond \text{nsc}(\varphi, \pi) \wedge \varphi]_I \\ &= [\]_I \text{ def } [\Diamond \text{nsc}(\varphi, \pi)]_I \cap [\varphi]_I \\ &= \Diamond \text{ def in } S5 \begin{cases} [\varphi]_I & \text{if } [\text{nsc}(\varphi, \pi)]_I \neq \emptyset, \\ \emptyset & \text{otherwise} \end{cases} \\ &= \text{ind hyp } \begin{cases} [\varphi]_I & \text{if } \llbracket \pi \rrbracket([\varphi]_I) \neq \emptyset, \\ \emptyset & \text{otherwise} \end{cases} \\ &= \llbracket \] \text{ def } \llbracket \text{might } \pi \rrbracket([\varphi]_I). \end{aligned}$$

This completes the proof of the lemma. ■

Lemma 14 $[\text{nsc}(\top, \pi)]_I = \llbracket \pi \rrbracket(I)$.

Proof:

$$\begin{aligned} [\text{nsc}(\top, \pi)]_I &= \text{nsc adeq} \quad \llbracket \pi \rrbracket([\top]_I) \\ &= [\]_I \text{ def} \quad \llbracket \pi \rrbracket(I). \end{aligned}$$
■

The following theorem gives the precise connections between WCPs and NSCs.

Theorem 15

1. $[\text{nsc}(\varphi, \pi)]_I = [\text{wcp}(\pi, \top) \downarrow \varphi]_I$.
2. $[\text{wcp}(\pi, \psi)]_I = [\psi \downarrow \text{nsc}(\top, \pi)]_I$.

Proof:

The first item:

$$\begin{aligned} [\text{nsc}(\varphi, \pi)]_I &= \text{nsc adeq} \quad \llbracket \pi \rrbracket([\varphi]_I) \\ &= \text{Lemma 10} \quad [\text{wcp}(\pi, \top)]_{[\varphi]_I} \\ &= \text{loc lemma} \quad [\text{wcp}(\pi, \top) \downarrow \varphi]_I. \end{aligned}$$

The second item:

$$\begin{aligned} [\text{wcp}(\pi, \psi)]_I &= \text{wcp adeq} \quad [\psi]_{\llbracket \pi \rrbracket(I)} \\ &= \text{Lemma 14} \quad [\psi]_{[\text{nsc}(\top, \pi)]_I} \\ &= \text{loc lemma} \quad [\psi \downarrow \text{nsc}(\top, \pi)]_I. \end{aligned}$$
■

6 A Hoare Calculus for Update Logic

We now present the axioms and rules of a deduction system for update logic based on the concepts of WCPs and WDPs from Section 4. As was explained there, the need to have two kinds of correctness statements arises from the fact that we have to specify weakest preconditions for success of a program π . For example, in the rule for negation, to specify that $\neg\varphi$ is the weakest precondition for success of π we use a conjunctive premiss $\langle \neg\varphi \rangle \pi \langle \top \rangle$ together with a disjunctive premiss $\{\varphi\} \pi \{\perp\}$ (see Lemma 5).

6.1 The rules for $\perp$

$$\overline{\{\top\} \perp \{\perp\}} \quad \overline{\langle \perp \rangle \perp \langle \perp \rangle}$$

6.2 The rules for p

$$\frac{}{\{p \rightarrow (\varphi \downarrow p)\} p \{ \varphi \}} \quad \frac{}{\langle \varphi \downarrow p \rangle p \langle \varphi \rangle}$$

6.3 The oracle rule

For every formula derivable in S5 we add the following rule (recall that derivability in S5 is decidable):

$$\frac{}{\varphi} \text{ oracle rule}$$

6.4 The consequence rules

$$\frac{\varphi \rightarrow \psi \quad \{ \psi \} \pi \{ \chi \} \quad \chi \rightarrow \xi}{\{ \varphi \} \pi \{ \xi \}}$$

$$\frac{\varphi \rightarrow \psi \quad \langle \psi \rangle \pi \langle \chi \rangle \quad \chi \rightarrow \xi}{\langle \varphi \rangle \pi \langle \xi \rangle}$$

6.5 The rules for sequential composition

$$\frac{\{ \varphi \} \pi_1 \{ \psi \} \quad \{ \psi \} \pi_2 \{ \chi \}}{\{ \varphi \} (\pi_1; \pi_2) \{ \chi \}}$$

$$\frac{\langle \varphi \rangle \pi_1 \langle \psi \rangle \quad \langle \psi \rangle \pi_2 \langle \chi \rangle}{\langle \varphi \rangle (\pi_1; \pi_2) \langle \chi \rangle}$$

6.6 Rules for disjunction

$$\frac{\langle \varphi \rangle \pi_1 \langle \top \rangle \quad \{ \neg \varphi \} \pi_1 \{ \perp \} \quad \langle \psi \rangle \pi_2 \langle \top \rangle \quad \{ \neg \psi \} \pi_2 \{ \perp \}}{\{ (\varphi \vee \psi) \rightarrow (\chi \downarrow (\varphi \vee \psi)) \} (\pi_1 \cup \pi_2) \{ \chi \}}$$

$$\frac{\langle \varphi \rangle \pi_1 \langle \top \rangle \quad \{ \neg \varphi \} \pi_1 \{ \perp \} \quad \langle \psi \rangle \pi_2 \langle \top \rangle \quad \{ \neg \psi \} \pi_2 \{ \perp \}}{\langle \chi \downarrow (\varphi \vee \psi) \rangle (\pi_1 \cup \pi_2) \langle \chi \rangle}$$

6.7 Rules for negation

$$\frac{\{ \varphi \} \pi \{ \perp \} \quad \langle \neg \varphi \rangle \pi \langle \top \rangle}{\{ \varphi \rightarrow (\psi \downarrow \varphi) \} \neg \pi \{ \psi \}}$$

$$\frac{\{ \varphi \} \pi \{ \perp \} \quad \langle \neg \varphi \rangle \pi \langle \top \rangle}{\langle \psi \downarrow \varphi \rangle \neg \pi \langle \psi \rangle}$$

6.8 Rules for might

$$\frac{\{\varphi\} \pi \{\perp\}}{\{\Box\varphi \vee \psi\} \text{ might } \pi \{\psi\}}$$

$$\frac{\langle\varphi\rangle \pi \langle\top\rangle}{\langle\Diamond\varphi \wedge \psi\rangle \text{ might } \pi \langle\psi\rangle}$$

If $F_1, \dots, F_n$ is a sequence of H_P statements each member of which is either an axiom or a statement following by the rules of the calculus from preceding statements in the sequence, then $F_1, \dots, F_n$ is a *derivation* in the calculus. If F equals F_n for some derivation $F_1, \dots, F_n$, then F is a *theorem* of the calculus (notation: $\vdash F$), and n is the *length* of the derivation of F . Note that we consider only derivations without hypotheses.

7 Soundness of the Calculus

To prove that the calculus is sound, we have to prove that if $F \in H_P$ is a theorem of the calculus, then F is valid (in the sense defined in Section 4). As usual, soundness is proved by induction on the length of the derivation of F . We are done if we can show that every axiom of the calculus is valid and that the rules of the calculus preserve validity.

Lemma 16 (wcp/wdp validity)

1. $\models \{\text{wdp}(\pi, \psi)\} \pi \{\psi\}$.
2. $\models \langle \text{wcp}(\pi, \psi) \rangle \pi \langle \psi \rangle$.

Proof: Immediate from Lemma 6, Lemma 8 and Definition 11. ■

Theorem 17 (Soundness) For all $F \in H_P$: If $\vdash F$ then $\models F$.

Proof: Soundness of the test axioms follows immediately from the wcp/wdp validity lemma: test axioms have the form $\{\text{wdp}(\pi, \psi)\} \pi \{\psi\}$ or $\langle \text{wcp}(\pi, \psi) \rangle \pi \langle \psi \rangle$.

The soundness of the oracle rule follows from the fact that for every S5 theorem φ and every information state I it holds, by the soundness of S5, that $[\varphi]_I = I$.

For the other rules we only give the argument for the conjunctive cases, the disjunctive cases being similar. For the soundness of the conjunctive consequence rule, assume that $\vdash \varphi \rightarrow \psi$, $\vdash \langle \psi \rangle \pi \langle \chi \rangle$ and $\vdash \chi \rightarrow \xi$. By the induction hypothesis we get from these three premisses respectively the following, for every information state I :

- (2) $[\varphi]_I \subseteq [\psi]_I$.
- (3) $[\psi]_I \subseteq [\chi]_{[\pi](I)}$.
- (4) $[\chi]_{[\pi](I)} \subseteq [\xi]_{[\pi](I)}$.

This yields conclusion (5).

$$(5) \quad [\varphi]_I \subseteq [\xi]_{[\pi](I)}.$$

Because I was arbitrary, conclude that $\models \langle \varphi \rangle \pi \langle \xi \rangle$.

For the soundness of the conjunctive rule of composition, assume $\vdash \langle \varphi \rangle \pi_1 \langle \psi \rangle$ and $\vdash \langle \psi \rangle \pi_2 \langle \chi \rangle$. By the induction hypothesis, these premisses give the following, again for every I :

$$(6) \quad [\varphi]_I \subseteq [\psi]_{[\pi_1](I)}.$$

$$(7) \quad [\psi]_{[\pi_1](I)} \subseteq [\chi]_{[\pi_2]([\pi_1](I))}.$$

Combining these two we get (8).

$$(8) \quad [\varphi]_I \subseteq [\chi]_{[\pi_2]([\pi_1](I))}.$$

From this by the semantic clause for sequential composition:

$$(9) \quad [\varphi]_I \subseteq [\chi]_{[\pi_1; \pi_2](I)}.$$

Since I was arbitrary, this establishes $\models \langle \varphi \rangle \pi_1; \pi_2 \langle \chi \rangle$.

For the soundness of the conjunctive rule of union, assume $\vdash \langle \varphi \rangle \pi_1 \langle \top \rangle$, $\vdash \{\neg \varphi\} \pi_1 \{\perp\}$, $\vdash \langle \psi \rangle \pi_2 \langle \top \rangle$, $\vdash \{\neg \psi\} \pi_2 \{\perp\}$. From these four premisses, with the induction hypothesis, Lemma 5 and Lemma 10:

$$(10) \quad \models \varphi \leftrightarrow \mathbf{wcp}(\pi_1, \top).$$

$$(11) \quad \models \psi \leftrightarrow \mathbf{wcp}(\pi_2, \top).$$

Now use the **wcp** validity lemma to conclude from this that (12).

$$(12) \quad \models \langle \chi \downarrow (\varphi \vee \psi) \rangle \pi_1 \cup \pi_2 \langle \chi \rangle.$$

For the soundness of the conjunctive rule of negation, the reasoning follows the same pattern.

For the soundness of the conjunctive rule of **might**, assume $\vdash \langle \varphi \rangle \pi \langle \top \rangle$. By the induction hypothesis this yields (13), for every I .

$$(13) \quad [\varphi]_I \subseteq [\pi](I).$$

From (13) by the S5 semantics of $\diamond$ and the semantic clause for **might**, (14).

$$(14) \quad [\diamond \varphi]_I \subseteq [\mathbf{might} \pi](I).$$

From this by set theoretic reasoning and Lemma 1, (15).

$$(15) \quad [\diamond \varphi]_I \cap [\psi]_I \subseteq [\psi]_{[\mathbf{might} \pi](I)}.$$

By modal propositional logic, (16):

$$(16) \quad [\diamond \varphi \wedge \psi]_I \subseteq [\psi]_{[\mathbf{might} \pi](I)}.$$

Since I was arbitrary, it follows that $\models \langle \diamond \varphi \wedge \psi \rangle \mathbf{might} \pi \langle \psi \rangle$. ■

8 Completeness of the Calculus

As a preliminary to the proof that the calculus is complete, we first show that statements of the form $\{\mathbf{wdp}(\pi, \psi)\} \pi \{\psi\}$ and $\langle \mathbf{wcp}(\pi, \psi) \rangle \pi \langle \psi \rangle$ are theorems of the calculus.

Lemma 18 (wcp/wdp derivability)

1. $\vdash \{\mathbf{wdp}(\pi, \psi)\} \pi \{\psi\}$.
2. $\vdash \langle \mathbf{wcp}(\pi, \psi) \rangle \pi \langle \psi \rangle$.

Proof: Induction on the structure of π , simultaneously for 1 and 2.

For atomic programs, the test axioms immediately give:

- (17) $\vdash \{\mathbf{wdp}(\perp, \psi)\} \perp \{\psi\}$.
- (18) $\vdash \langle \mathbf{wcp}(\perp, \psi) \rangle \perp \langle \psi \rangle$.
- (19) $\vdash \{\mathbf{wdp}(p, \psi)\} p \{\psi\}$.
- (20) $\vdash \langle \mathbf{wcp}(p, \psi) \rangle p \langle \psi \rangle$.

The cases of composition and union are left to the reader.

For negation, conjunctive case, we have to establish $\vdash \langle \mathbf{wcp}(\neg\pi, \psi) \rangle \neg\pi \langle \psi \rangle$. By the definition of $\mathbf{wcp}$, this is equivalent to: $\vdash \langle \psi \downarrow \mathbf{wdp}(\pi, \perp) \rangle \neg\pi \langle \psi \rangle$. By the induction hypothesis we have:

- (21) $\vdash \{\mathbf{wdp}(\pi, \perp)\} \pi \{\perp\}$.
- (22) $\vdash \langle \mathbf{wcp}(\pi, \top) \rangle \pi \langle \top \rangle$.

Using Definition 11 and the conjunctive consequence rule we get from (22):

- (23) $\vdash \langle \neg\mathbf{wdp}(\pi, \perp) \rangle \pi \langle \top \rangle$.

The first negation rule can now be applied to (21) and (23) to get:

- (24) $\vdash \langle \psi \downarrow \mathbf{wdp}(\pi, \perp) \rangle \neg\pi \langle \psi \rangle$.

For the disjunctive case of negation the reasoning is similar.

For **might**, conjunctive case, we have to establish $\vdash \langle \mathbf{wcp}(\mathbf{might} \pi, \psi) \rangle \mathbf{might} \pi \langle \psi \rangle$. By the definition of $\mathbf{wcp}$, this is equivalent to: $\vdash \langle \Diamond \mathbf{wcp}(\pi, \top) \wedge \psi \rangle \mathbf{might} \pi \langle \psi \rangle$. By the induction hypothesis we have (22), so the conjunctive rule of **might** gives the desired conclusion. The disjunctive case of **might** is similar. ■

The completeness theorem follows almost immediately.

Theorem 19 (Completeness) For all $F \in H_P$: If $\models F$ then $\vdash F$.

Proof: Assume F is a formula φ from mlp . Then $\models \varphi$ entails that φ is a theorem of S5, and therefore an axiom of the calculus, so $\vdash \varphi$.

Assume F has the form $\{\varphi\} \pi \{\psi\}$ and suppose:

$$(25) \quad \models \{\varphi\} \pi \{\psi\}.$$

From this and the **wcp/wdp** validity lemma:

$$(26) \quad \models \varphi \rightarrow \mathbf{wdp}(\pi, \psi).$$

All S5 validities are axioms, so from (26):

$$(27) \quad \vdash \varphi \rightarrow \mathbf{wdp}(\pi, \psi).$$

From the **wcp/wdp** derivability lemma:

$$(28) \quad \vdash \{\mathbf{wdp}(\pi, \psi)\} \pi \{\psi\}.$$

From (27) and (28) by the disjunctive consequence rule:

$$(29) \quad \vdash \{\varphi\} \pi \{\psi\}.$$

The reasoning for the case of $\langle \varphi \rangle \pi \langle \psi \rangle$ is similar. ■

9 Reasoning about Update Logic via S5

Just for the record we mention a fact about update logic which follows immediately from our 'reduction to S5' (but note that this fact was already proved in [10]).

Theorem 20 *Update logic is decidable.*

Proof: The decision problem for update logic is the question: which $\pi \in L_P$ have the property that they are valid (accepted in every input state I)? In other words: which π have the property that for all I it holds that $[\pi](I) = I$? The decision procedure for π is as follows. Use the definition of **wcp** to find $\mathbf{wcp}(\pi, \top)$. By Lemma 10 we know that $[\pi](I) = [\mathbf{wcp}(\pi, \top)]_I$, so the decision problem for π reduces to the question whether $\mathbf{wcp}(\pi, \top)$ is S5-valid. Use the decision procedure for S5 to settle this question. ■

In update logic there is a distinction between acceptable and accepted information, witness the following definition.

Definition 14

1. A program π is accepted in I if $I = [\pi](I)$.
2. A program π is acceptable in I if $[\pi](I) \neq \emptyset$.

It is the universal version of the first of these which is taken as the notion for universal validity, but one might consider the universal version of the second one just as well.

Definition 15

1. A program π is *always accepted* (or *valid*) if for all I it holds that $[\pi](I) = I$.
2. A program π is *always acceptable* if for all $I \neq \emptyset$ it holds that $[\pi](I) \neq \emptyset$.

An obvious question suggests itself: are the notions of being always accepted and being always acceptable equivalent? Using the S5 connection it is easy to see the relation between these two notions. Now we need not concern ourselves with the case of $I = \emptyset$, for the elimination lemma forces $[\pi](\emptyset) = \emptyset$ for every π . Thus, there is no harm in adopting the usual convention that S5 models have a non-empty set of worlds. The ‘static’ version of π is *always accepted* is (30).

$$(30) \quad S5 \models \mathbf{wcp}(\pi, \top).$$

The ‘static’ version for π is *always acceptable*, on the other hand, is (31). Note that this translation hinges on the assumption of non-emptiness of S5 models.

$$(31) \quad S5 \models \Diamond \mathbf{wcp}(\pi, \top).$$

So the notion of being always acceptable is decidable as well, but it does not coincide with the notion of being always accepted. Indeed, we have that $S5 \models \varphi$ implies $S5 \models \Diamond \varphi$, because of the reflexivity of accessibility, so (30) implies (31), but not the other way around. Take φ equal to $\Diamond p \rightarrow p$ for a simple counterexample. We have $S5 \not\models \Diamond p \rightarrow p$ (take a non- p world in a model containing both p and non- p worlds), but $S5 \models \Diamond(\Diamond p \rightarrow p)$. To see this latter fact, take an arbitrary w in an arbitrary universal S5 model I . If there are no p worlds, then $I, w \vdash \Diamond(\Diamond p \rightarrow p)$; if there are p worlds, then there is a p world w' for which $I, w' \vdash \Diamond p \rightarrow p$, so by the fact that accessibility is universal again $I, w \vdash \Diamond(\Diamond p \rightarrow p)$. Note, by the way, that $\Diamond(\Diamond p \rightarrow p)$ is the modal counterpart of a predicate logical sentence that philosophical logicians sometimes refer to as ‘Plato’s principle’: $\exists x(\exists x Px \rightarrow Px)$.

The S5 counterexample can be transposed to update logic, of course: $p \cup \neg \text{might } p$ is an example of a program which is always acceptable but not always accepted.

For a next illustration of reasoning about update logic via S5 we take a quick look at valid consequence in update logic. In his paper [14] Veltman discusses various notions of valid consequence. He distinguishes the following three definitions.

Definition 16

1. $\pi_1 \models_1 \pi_2$ if for all I it holds that $[\pi_1](I) = I$ implies $[\pi_2](I) = I$.
2. $\pi_1 \models_2 \pi_2$ if for all I it holds that $[\pi_1](I) = [\pi_2]([\pi_1](I))$.
3. $\pi_1 \models_3 \pi_2$ if $[\pi_1](W) = [\pi_2]([\pi_1](W))$.

The following proposition reduces these notions to S5.

Proposition 21

1. $\pi_1 \models_1 \pi_2$ iff $S5 \vdash \Box \mathbf{wcp}(\pi_1, \top) \rightarrow \Box \mathbf{wcp}(\pi_2, \top)$.

2. $\pi_1 \models_2 \pi_2$ iff $S5 \vdash \mathbf{wcp}(\pi_1, \top) \leftrightarrow \mathbf{wcp}(\pi_1; \pi_2, \top)$.
3. $\pi_1 \models_3 \pi_2$ iff $S5 \vdash (\mathbb{A} \Diamond \varphi_i) \rightarrow (\mathbf{wcp}(\pi_1, \top) \leftrightarrow \mathbf{wcp}(\pi_1; \pi_2, \top))$,
where the φ_i are all conjunctions of the form $(\neg)p_1 \wedge \dots \wedge (\neg)p_n$, with $p_1, \dots, p_n$
the list of proposition letters occurring in π_1 or π_2 .

Proof: The first item:

For all I : $[\pi_1](I) = I$ implies $[\pi_2](I) = I$
iff for all I : if $\forall w \in I: I, w \vdash \mathbf{wcp}(\pi_1, \top)$ then $\forall w \in I: I, w \vdash \mathbf{wcp}(\pi_2, \top)$
iff for all I : $I \models \Box \mathbf{wcp}(\pi_1, \top) \rightarrow \Box \mathbf{wcp}(\pi_2, \top)$
iff $S5 \vdash \Box \mathbf{wcp}(\pi_1, \top) \rightarrow \Box \mathbf{wcp}(\pi_2, \top)$.

The second item is immediate from the definitions of the validity notions, the $\mathbf{wcp}$ adequacy lemma and the completeness of S5.

For the third item, note that $I \models \mathbb{A} \Diamond \varphi_i$ (where $\models$ denotes S5 validity) for precisely those information sets I that express total ignorance with respect to all proposition letters in π_1 and π_2 , i.e., for the sets I that are indistinguishable from W as far as π_1 and π_2 are concerned. The S5 formula expresses that for such I , all worlds in $[\pi_1; \pi_2](I)$ are worlds in $[\pi_2](I)$ and vice versa. This is precisely what the validity notion $\models_3$ expresses. ■

10 Calculations of Consistency

Veltman calls a program π of L_P consistent if there is some information state I for which $[\pi](I) \neq \emptyset$. Intuitively, consistent programs are programs that can be used to convey information. By the soundness of the Hoare calculus, consistency of an update program π boils down to the question whether there is some S5 consistent φ such that $\vdash \langle \varphi \rangle \pi \langle \top \rangle$. We illustrate how to check consistency for two examples taken from Veltman [14]. We calculate with WCPs, but by virtue of the fact that WCP reasoning and NSC reasoning are equivalent (Theorem 15), calculations with NSCs work just as well.

Example 1 *might p ; $\neg p$ is consistent.*

Proof:

$$\begin{aligned}
\mathbf{wcp}(\text{might } p; \neg p, \top) &= \mathbf{wcp}(\text{might } p; \mathbf{wcp}(\neg p, \top)) \\
&= \mathbf{wcp}(\text{might } p; \neg \mathbf{wcp}(p, \top)) \\
&= \mathbf{wcp}(\text{might } p; \neg p) \\
&= \Diamond p \wedge \neg p
\end{aligned}$$

Since $\Diamond p \wedge \neg p$ does have S5 models, so it is not S5-provably equivalent to $\perp$. ■

Example 2 *$\neg p$; might p is not consistent.*

Proof:

$$\begin{aligned}
\text{wcp}(\neg p ; \text{might } p, \top) &= \text{wcp}(\neg p; \text{wcp}(\text{might } p, \top)) \\
&= \text{wcp}(\neg p, \Diamond \text{wcp}(p, \top)) \\
&= \text{wcp}(\neg p, \Diamond p) \\
&= \Diamond p \downarrow \neg \text{wcp}(p, \top) \\
&= \Diamond p \downarrow \neg p \\
&= \neg p \wedge \Diamond(p \wedge (p \downarrow \neg p)) \\
&= \neg p \wedge \Diamond(p \wedge \neg p) \\
&= \neg p \wedge \perp \\
&= \perp
\end{aligned}$$

■

It is a fact about the Hoare calculus that it yields weakest preconditions, provided that the consequence rules are not used for precondition strengthening. For good measure, we also give the Hoare deductions of weakest preconditions for the two examples.

$$\begin{array}{c}
\frac{\langle \Diamond p \wedge \neg p \rangle \text{ might } p \langle \neg p \rangle \quad \langle \neg p \rangle \neg p \langle \top \rangle}{\langle \Diamond p \wedge \neg p \rangle \text{ might } p; \neg p \langle \top \rangle.} \\
\frac{\langle \Diamond p \downarrow \neg p \rangle \neg p \langle \Diamond p \rangle \quad \langle \Diamond p \rangle \text{ might } p \langle \top \rangle}{\langle \Diamond p \downarrow \neg p \rangle \neg p; \text{ might } p \langle \top \rangle.}
\end{array}$$

11 Conclusion

By our construction of a calculus for update logic we claim to have established a clean connection between a species of dynamic logic and good old static S5. There is scope for quite a bit of further work. In the first place, one might wish to consider extensions of update logic with new operations, such as ‘downdating’ (retracting information), and study how these could be linked to static modal notions. Also, a modal study of the preference relation on information states that Veltman proposes seems to be worthwhile. In a different direction, one may study the combination of the calculus given here with the calculus from [13] in a system of quantificational update logic satisfying the desiderata which Groenendijk and Stokhof list in [3]. Giving a Hoare style axiomatisation of a system of dynamic assignment logic with epistemic modalities, using modal predicate logic as the assertion language, is a feasible task now, witness Van Eijck and Cepparello [12].

Acknowledgement

This paper has benefited from helpful comments by Johan van Benthem, Giovanna Cepparello, Tim Fernando, Willem Groeneveld, Marcus Kracht, Frank Veltman, Kees Vermeulen, Jørgen Villadsen and Albert Visser.

References

- [1] J. Barwise. Noun phrases, generalized quantifiers and anaphora. In P. Gärdenfors, editor, *Generalized Quantifiers: linguistic and logical approaches*, pages 1–30. D. Reidel Publishing Company, Dordrecht, 1987.
- [2] R. Goldblatt. *Logics of Time and Computation*, volume 7 of *CSLI Lecture Notes*. CSLI, Stanford, 1987. Distributed by University of Chicago Press.
- [3] J. Groenendijk and M. Stokhof. Two theories of dynamic semantics. In J. van Eijck, editor, *Logics in AI—European Workshop JELIA '90*, pages 55–64, Berlin, 1991. Springer Lecture Notes in Artificial Intelligence.
- [4] I. Heim. *The Semantics of Definite and Indefinite Noun Phrases*. PhD thesis, University of Massachusetts, Amherst, 1982.
- [5] C.A.R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):567–580, 583, 1969.
- [6] H. Kamp. A theory of truth and semantic representation. In J. Groenendijk et al., editors, *Formal Methods in the Study of Language*. Mathematisch Centrum, 1981.
- [7] L. Karttunen. Discourse referents. In J. McCawley, editor, *Syntax and Semantics 7*, pages 363–385. Academic Press, 1976.
- [8] M. Kracht. Splittings and the finite model property. Logic group preprint series, Department of Philosophy, University of Utrecht, 1991. To appear in the JSL.
- [9] R. Stalnaker. Pragmatics. In D. Davidson and G. Harman, editors, *Semantics of Natural Language*, pages 380–397. Reidel, 1972.
- [10] J. van Benthem. Semantic parallels in natural language and computation. In H.-D. Ebbinghaus et al., editors, *Logic Colloquium, Granada, 1987*, pages 331–375, Amsterdam, 1989. Elsevier.
- [11] J. van Benthem. General dynamics. Technical Report LP-90-11, ITLI, University of Amsterdam, 1990.
- [12] J. van Eijck and G. Cepparello. Dynamic modal predicate logic. Manuscript, CWI, Amsterdam, 1992.
- [13] J. van Eijck and F.J. de Vries. Dynamic interpretation and Hoare deduction. *Journal of Logic, Language, and Information*, 1:1–44, 1992.
- [14] F. Veltman. Defaults in update semantics. Manuscript, Department of Philosophy, University of Amsterdam, 1991. To appear in the *Journal of Philosophical Logic*.

Towards a dynamic and compositional treatment of temporal expressions

Frank Van Eynde

Nationaal Fonds voor Wetenschappelijk Onderzoek
België

This paper presents a format for the representation and interpretation of temporal relations in texts. It covers the same phenomena as Discourse Representation Theory, but it is more in line with the principle of compositionality: the temporal structure of a text is described as a function of the meanings of the individual clauses and the temporal expressions they contain, rather than the other way round.

The format is highly modular: it has a compositional core consisting of two modules (tense and aspect) and a dynamic scheme for the interpretation of cross-clausal relations, as expressed by temporal anaphora and the like. The format will be presented in sections 4 and 5, and compared with the DRT treatment in section 6. The first three sections prepare the ground with a discussion of the notions "time of reference" and "point of perspective", and with an overview of the main surface markers of temporal relations, i.e. the grammaticalised forms of tense and aspect.

1. The time of reference

The notion of "reference time" was first introduced by Hans Reichenbach, and then adopted by many linguists and logicians. As most other popular terms it has come to mean a number of different things, depending on the system in which it functions. Two of these systems will be presented and compared: the original Reichenbach one (1.1) and the one of Discourse Representation Theory (1.2). Taking into account some of the criticisms which the latter has recently elicited I will then define the role which the reference time plays in my treatment (1.3).

1.1. In Reichenbach 1947 the time of reference serves three different functions.

The first, and probably the most familiar, concerns the semantic analysis of the grammaticalised forms of tense and aspect. Reichenbach claimed that those forms cannot simply be analysed as relations between a time of event (E) and a time of speech (S). Indeed, taking a linearly ordered set of time points as a model, there are just three possible relations between E and S: E is before S ($E < S$), E is after S ($E > S$) and E coincides with S ($E = S$). It follows that the English past tenses, for instance, which include a.o. the simple past, the pluperfect, the present perfect and the past progressive, will be analysed as $E < S$, thus ignoring the fact that they have mutually distinct meanings.

Reichenbach's proposal to make use of a point of reference (R), intermediating between E and S, enhances the expressive power of the formalism: instead of three relations between E and S, he uses three relations between R and S, plus three relations between E and R, resulting in nine different combinations. Analysing the past tenses in terms of both relations, Reichenbach is able to make the required differentiations:

Simple Past	$R < S$	$E = R$	$R, E \text{ — } S$
Present Perfect	$R = S$	$E < R$	$E \text{ — } S, R$
Pluperfect	$R < S$	$E < R$	$E \text{ — } R \text{ — } S$

The line stands for precedence, the comma for coincidence

The further distinction between the simple past and the past progressive is made by allowing E to be either a moment or a period of some length:

Past Progressive $R < S$ $E \supset R$ $\frac{E}{R} \text{---} S$

The second function of the time of reference is to serve as the denotatum of time adverbials, like "now", "yesterday", "in 1678", "when John came", etc. These adverbials cannot generally be interpreted as specifications of the time of event. In

(1) at 5 o'clock he had already disappeared
the time denoted by "at 5 o'clock" is not the time of the disappearance (E) nor the time of speech (S), but rather a time at which the disappearance had already taken place. Given the analysis of the pluperfect it is obvious to treat it as a specification of the intervening time of reference. Similarly, in

(2) at 5 o'clock they were all dancing
the adverbial does not denote the time of dancing, which is a period of some length, but rather an instant which is included in the dancing time, and given the analysis of the past progressive the obvious candidate for being this instant is again the time of reference.

The third function, and the one which will interest us most in the rest of the paper, concerns the interpretation of sequences of sentences in discourse. In the following piece of text, taken from W. Somerset Maugham's "Of Human Bondage" and discussed in Reichenbach 1947, we can identify six events E_1 :

(3) But Philip ceased to think of her for a moment after he had settled down in his carriage. He thought only of the future. He had written to Mrs. Otter, the massière to whom Hayward had given him an introduction, and had in his pocket an invitation to tea the following day.

E_1 = Philip cease to think of her

E_2 = Philip settle down in his carriage

E_3 = Philip think only of the future

E_4 = Philip write to Mrs. Otter, the massière

E_5 = Hayward give Philip an introduction to Mrs. Otter

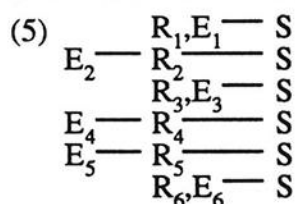
E_6 = Philip have an invitation to tea in his pocket

The clauses with a simple past are analysed as $R, E \text{---} S$, and those with a pluperfect as $E \text{---} R \text{---} S$. This information from the tenses does not -in itself- say anything about the relations between the times of event of the different clauses; indeed, even if we would accept the rather obvious stipulation that the time of speech is the same for the whole fragment, there is nothing in the analysis of the simple past and the pluperfect which prevents us from deriving a representation like

(4) $R_3, E_3 \text{---} \frac{E_4 \text{---} R_4 \text{---} S}{S}$

in which Philip's writing to Mrs. Otter (E_4) takes place after the thinking of his future (E_3). This is not the way in which (3) is interpreted (or intended), though.

In order to arrive at a more plausible interpretation Reichenbach introduces the principle of the permanence of the reference point. It requires that the time of reference be the same for the whole fragment, and thus leads to analysis:

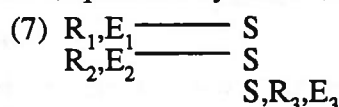


The events 1, 3 and 6 are claimed to hold at the same time: Philip's ceasing to think of her, his thinking of the future and his having an invitation to tea all hold at the permanent reference time R₁. The events 2, 4 and 5, which are described in clauses with a pluperfect, precede this time of reference, but there is no information about their mutual order ².

The principle of the permanence of the reference point is not without exceptions: if there is a time adverbial which explicitly introduces another reference point than the one which is contextually given, then the one given by the adverbial gets priority. In

(6) He was healthier when I saw him than he is now

the time of reference of the first two clauses is in the past, but the one of the third clause, specified by "now", is in the present:



Reichenbach calls this the positional use of the reference point.

1.2. Of the three functions which the time of reference has in Reichenbach's work, DRT concentrates on the third, i.e. on the role of the point of reference in the interpretation of texts.

Its role in the semantic analysis of the individual forms of tense and aspect is secondary. The distinction between the *passé simple* and the *imparfait*, for instance, is not described in terms of the contributions which they make to the truth conditions of individual sentences, but rather in terms of the different contributions which they make to the interpretation of texts: "The choice of the tense form depends on the function that the sentence in which it occurs has in a text. In other words, the factors which determine the use of *imparfait* and *passé simple* can only be explained at the level of discourse representation." [Kamp & Rohrer 1983, 253]

Indeed, whereas it seems rather hard to pin down the semantic difference between (8) and (9)

(8) Marie téléphonait

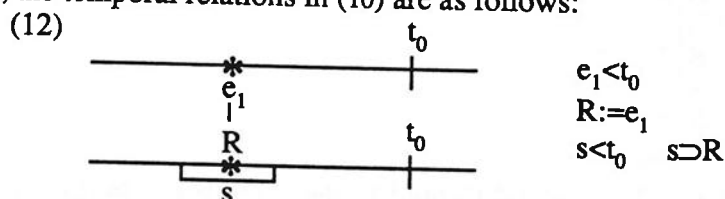
(9) Marie téléphona

it is easy to describe the difference between (10) and (11)

(10) Quand Pierre entra, Marie téléphonait

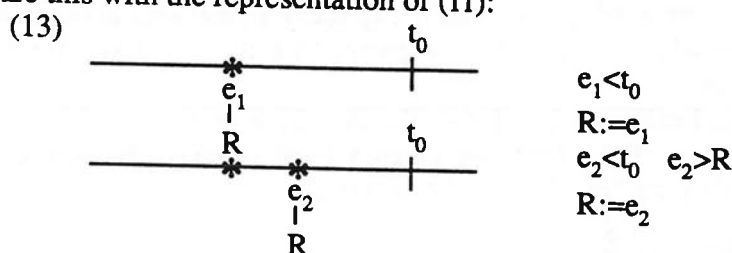
(11) Quand Pierre entra, Marie téléphona

In (10) Marie was already at the phone when Pierre entered, whereas in (11) she started the phone call after Pierre entered. With t_0 as the time of utterance, e_1 as the event of Pierre's entering and S as the state of Marie's being at the phone, the temporal relations in (10) are as follows:



The point of reference is provided by the event of the first clause ($R := e_1$) and serves as a starting point for the interpretation of the second clause. Since the latter is in the imparfait it expresses a state (S) which precedes t_0 and which includes the point of reference, and hence also e_1 .

Compare this with the representation of (11):



In this case, the second clause is in the passé simple and therefore introduces a second event (e_2), which follows the point of reference (R) and hence also e_1 .

A second difference is that e_2 in (11) provides the point of reference for the interpretation of the next clause ($R := e_2$), whereas the state in (12) does not provide a new point of reference, so that the next clause will be interpreted with respect to e_1 .

Starting from this scheme the meaning of a tense -defined as its contribution to the interpretation of a text- consists of three parts: (1) it introduces a state or an event in the discourse representation structure (DRS); (2) it locates this state or event with respect to the time of speech; (3) it defines the relation between the event or state and the current time of reference.

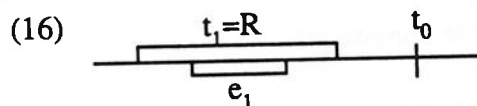
The imparfait, for instance, introduces a state (s_i), and locates it before the time of speech ($s_i < t_0$) and around the current time of reference ($s_i > R$). The passé simple introduces an event (e_i), and locates it before the time of speech ($e_i < t_0$). As for its relation with R , there are several possibilities: in (11) the event in the second clause follows R , but in

(14) Marie chanta et Pierre l'accompagna au piano
it coincides with R , since Pierre's performance on the piano coincides with Marie's singing, and in

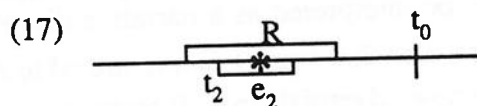
(15) L'année dernière Jean escalada le Cervin. Le premier jour il monta jusqu'à la cabane H.

it forms part of R: the climbing to hut H is part of the climbing of the Cervin. It follows that the constraint on the relation with R has to be rather weak; in Kamp & Rohrer 1983 it is proposed that the event introduced by the passé simple should not entirely precede R: $\sim e_1 < R$. Notice that this allows for a range of possible relations, including the one which was used for the imparfait ! The only difference between both forms, then, is that the passé simple introduces an event and the imparfait a state.

The role of the reference time in the analysis of the time adverbials is a complex matter in DRT. Time adverbials are treated as specifiers of temporal location times and the latter are introduced in DRS as intervals t_i , for $i > 0$. Whether these location times serve as reference times depends on many factors. For instance, the adverbial in the first clause of (15), "l'année dernière", specifies an interval which provides the time of reference for the interpretation of the first clause:



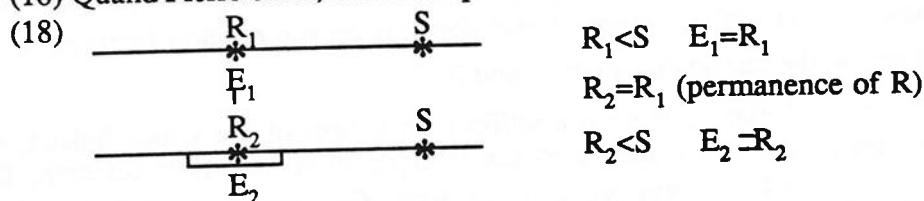
but the time of reference for the second clause is the event described in the first clause ($R := e_1$), and not the time adverbial, "le premier jour"; the latter specifies an interval t_2 without any special role in the discourse interpretation:



There is a lot more to be said about the treatment of time adverbials in DRT (cf. sections 2 and 6). At this point, however, it suffices to observe that time adverbials do not necessarily specify R.

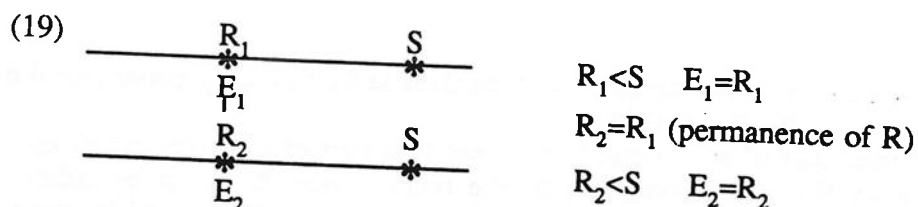
1.3. The role which Reichenbach attributes to the time of reference in the interpretation of texts is quite different from the role it has in DRT. To appreciate the differences compare the DRT analyses given in (12) and (13) with the ones which Reichenbach would provide for the same sentences:

(10) Quand Pierre entra, Marie téléphonait



The relation between E_1 and E_2 in (18) is the same as between e_1 and S in (12): in both cases Marie's phoning temporally includes Pierre's entering.

(11) Quand Pierre entra, Marie téléphona



Here, the relation between E_1 and E_2 in (19) is different from the one between e_1 and e_2 in (13): whereas the former predicts that Marie's phone call is simultaneous with Pierre's entering ($E_2 = E_1$), the latter predicts that Marie called after Pierre had entered ($e_2 > e_1$).

In order to get the same result in Reichenbach's framework one could loosen the constraint on the relation between E and R.

Instead of requiring it to be identical, one could allow E to follow R, as in DRT. Notice, however, that the resulting analysis of the passé simple, $R < S$ and $E > R$, is the one which Reichenbach employs for the future-in-the-past, i.e. for the conditionnel I. Assigning this meaning to the passé simple would amount to claiming that (11) is synonymous with

(20) Quand Pierre entra, Marie téléphonerait

which it is not.

Another way out would be to loosen the constraint of the permanence of the reference point, allowing the reference point to move forward, even if there is no time adverbial which triggers this move. This is, in essence, the solution proposed in Dowty's temporal discourse interpretation principle: "Given a sequence of sentences $S_1, S_2, \dots, S_n$ to be interpreted as a narrative discourse, the reference time of each sentence S_i (for i such that $1 < i \leq n$) is interpreted to be:

- (a) a time consistent with the definite time adverbials in S_i , if there are any
- (b) otherwise, a time which immediately follows the reference time of the previous sentence S_{i-1} ." [Dowty 1986, 45]

Reichenbach's positional use of the reference point is subsumed under (a), and the permanence of the reference point is now replaced by a systematic forward movement (b), at least for narrative discourse³.

An attractive property of this proposal is that it allows for a uniform treatment of the passé simple: its meaning is constant ($R_i < S$ and $E_i = R_i$), and whether E_i follows E_{i-1} in a given piece of text depends on the relation between R_i and R_{i-1} , and not on the relation between E_i and R_i .

DRT does not allow for such a uniform treatment of the tense. Indeed, the contribution of the passé simple to the interpretation of (11) is different from the one in (14) and (15): $R > e_i$ vs. $R = e_i$ vs. $R > e_i$. One way to accommodate this variation is to postulate a very broad meaning for the passé simple, as Kamp & Rohrer do, but this turns out to be so broad that it includes the one of the imparfait.

Another way to accommodate the facts in DRT would be to claim that the time of reference is not identical to a previous event ($R := e_{i-1}$), but rather follows it by a short distance ($e_{i-1} < R$). This alternative is the one adopted in Hinrichs 1986 and Partee 1984: "We can generalize the idea that in the simple linear case an event-clause moves the narrative forward by bringing in a new reference time that is 'just after' the given event." [Partee 1984, 256] Applying this scheme to the passé simple we could assign it a rather specific meaning, such as $e_i \subseteq R$, and achieve the effect of forward movement by locating R after e_{i-1} .

As a result we now have four ways of dealing with fragments like (11). There is the original Reichenbach treatment, which does not get the facts right, and there is the original DRT treatment, which is less felicitous from a methodological point of view. In between, there are the treatments of Dowty and Partee. Both manage to get the facts right without giving up the assumption that the contribution of the tense is constant, but they achieve this in different ways. Whereas Dowty defines the time of reference with respect to the time of reference of a previous clause, Hinrichs and Partee define it with respect to a previous event, as in DRT.

It follows that the latter make the forward movement of the time of reference dependent on the aspectual class of the previous clause: if it is an event, the time of reference moves; if it is a state, the time of reference does not move.

This reliance on the state/event distinction has been criticised in Dowty 1986. Since it touches on the very core of DRT, it is worth examining in some detail.

One of the basic claims of DRT is that the interpretation of a text is more than the sum of the interpretations of its sentences. More specifically, it treats the meaning of the individual sentences as a function of the meaning of the text, rather than the other way round. Formalising this holistic perspective on meaning, DRT postpones the interpretation of each individual sentence till after the construction of a discourse representation for the text as a whole. The treatment of the distinction between passé simple and imparfait provides a good illustration of this strategy.

Now, if the rules for the construction of DRS refer to the aspectual class of the individual sentences, and if the latter is not a syntactic but rather a semantic property, as has been argued by most linguists who have investigated the matter (cf. Mourelatos 1978, Dowty 1979, a.o.), then it follows that the individual sentences should be assigned a model-theoretic interpretation before the rules of DRS construal are applied. But this contradicts the basic DRT assumption. In Dowty's words: "If the compositional model-theoretic interpretation of the sentences in a discourse is determined only after a discourse representation has been constructed (as Kamp proposes), and if it is only in the model-theoretic interpretation that the aspectual class of a sentence is fully apparent (as I am arguing), then how can aspectual class have an effect on how the temporal relationships between sentences are represented in the discourse representation?" [Dowty 1986, 40]

As an alternative to DRT's holistic view on sentence meaning, Dowty pleads for a treatment which is more in line with the principle of compositionality: "... it seems that the temporal relationships among sentences in a discourse depends on the prior determination of the semantics of the individual sentences, contrary to Kamp's proposal as I understand it." [ibid.]

Following Dowty's suggestion I will explore an alternative in which the contribution of the temporal expressions to the meaning of a text is described as a function of their contribution to the meanings of individual sentences.

For the grammaticalised forms of tense and aspect, this implies that their meanings should be properly differentiated at the level of the sentence, or - rather- the clause.

For the time adverbials, it implies that their contribution to the meaning of a clause should be constant. The DRT treatment, in which they are treated as specifying the time of reference in some cases and not in others, is less attractive from this point of view than the treatments in which they invariably specify R. Examples of the latter are not only the ones proposed by Reichenbach and Dowty, but also the more DRT oriented treatments of Hinrichs and Partee: "The adverb, whether clausal or phrasal, provides a descriptive characterisation of the new reference time; it may identify it completely ('at 3 o'clock on June 12') or simply put bounds on it, as with frame adverbials like 'in June'." [Partee 1984, 257]

In both cases, i.e. the analysis of the grammaticalised forms of tense and aspect, and the treatment of the time adverbials, the time of reference will play an important role, but in order to cope with a larger set of facts and phenomena than the ones discussed in Reichenbach 1947 his notion of the time of reference has to be modified.

One such modification concerns the punctual nature of the time of reference. In Reichenbach 1947 R is invariably a single moment of time, but examples like (15) make it clear that the time of reference may be a period of some length as well. Partee is explicit on this: "From the fact that there are conditions requiring reference times to include events and to be included within states, i.e. within some period for which a given state holds, it follows that reference times must be construed as protracted events or bounded intervals." [Partee 1984, 255] This has some important consequences for the expressive power of the representation formalism (cf. section 4).

Another modification concerns the role of the time of reference in the representation of individual clauses. Reichenbach treats it as an intermediary between the time of event and the time of speech, but taking into account some developments in DRT, I will rather treat it as an intermediary between the time of event and a contextually determined point of perspective. The motivation for this modification is given in the next section.

2. The point of perspective

In Kamp and Rohrer 1983 the point of perspective is introduced for differentiating between various kinds of indexical and anaphoric time adverbials, and in Kamp and Rohrer ms. it is used for describing the meaning of the plus-que-par-

fait in flashbacks. As a first introduction, though, the need for a point of perspective, as distinct from both the time of speech and the time of reference, can most easily be demonstrated in the treatment of indirect discourse. In

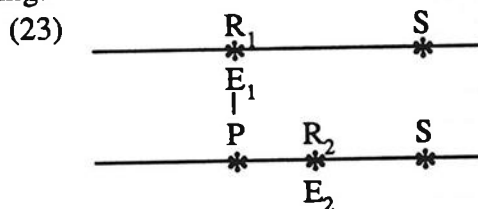
(21) John said: "I will come at 5"

the time of speech of the second clause is the time of John's utterance, which is in the past of the time of speech of the first clause. Shifting to indirect discourse we get

(22) John said that he would come at 5

in which there is only one time of speech. John's utterance is in the past of that time of speech, and provides a starting point for the interpretation of the subclause. The subclause has a time of reference, specified by "at 5", which is equal to its time of event, "John's coming", but not to the time of saying (E_1).

As a matter of fact, its time of reference is in the future with respect to the time of saying:



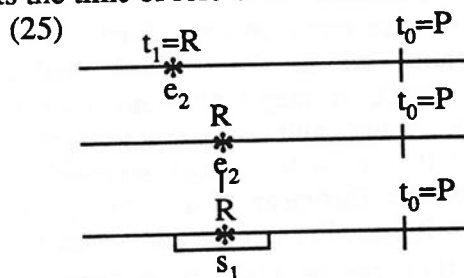
In the terminology of DRT, the time of saying (E_1) provides the point of perspective (P) for the interpretation of the subclause.

Notice that the transition from direct to indirect speech triggers a transposition of deictic into anaphoric expressions: this does not only affect the pronoun (I → he) but also the tense (will → would).

Given the distinction between point of perspective and time of reference one can express the difference between anaphoric and indexical time adverbials. An example of the former is "deux jours après" in the second clause of

(24) Kissinger arriva au Caire le sixième juillet.
Deux jours après il partit pour Jérusalem.
Il était très fatigué.

It shifts the time of reference from the sixth of July to the eighth:



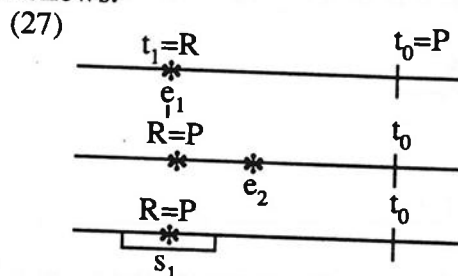
Since the state in the third clause is located with respect to the current time of reference, it is interpreted to hold at the time of leaving for Jerusalem, i.e. July 8th.

When "deux jours après" is replaced by the indexical time adverbial, "dans deux jours", we get another picture.

- (26) Kissinger arriva au Caire le sixième juillet.
 Dans deux jours il partait/*partit pour Jérusalem.
 Il était très fatigué.

The leaving for Jerusalem takes place two days after July the 6th, as in (24), but in contrast to the anaphoric adverbial, the indexical adverbial does not shift the time of reference. Instead, it triggers a shift of the time of perspective into the past and describes the future leaving from that point of view.

It follows that the state in the third clause is interpreted with respect to July the 6th, and not with respect to July the 8th. Kamp and Rohrer 1983 analyses (26) as follows:



The point of perspective is dissociated from the point of speech and identified with the time of reference which is provided by the time of event of the first clause. This shift in the past triggers transposition and the imparfait in the second and third clause is therefore interpreted as if it were a présent simple (cf. the simultaneity of time of reference and point of perspective). This also explains why the passé simple is ungrammatical in (26): it is the imparfait and not the passé simple which results from a transposition of the présent simple.

In section 5 I will propose an alternative analysis of (26), but the intuition that one should distinguish between the time of speech and the point of perspective is one which I will adopt.

3. A parametric analysis of the forms of tense and aspect

One of the main functions of the discourse representation format to be developed in this paper is to serve as a framework for the semantic analysis of the grammaticalised forms of tense and aspect. A major problem with terms like "tense" and "aspect", though, is that no two authors seem to agree on what they actually mean. For that reason I will start with a short overview of the relevant forms in the nine languages of the European Community, i.e. Danish, Dutch, English, German, Greek, French, Italian, Portuguese and Spanish.

As demonstrated in Van Eynde 1991, they can be analysed in terms of five binary distinctions: $[\pm\text{past}]$, $[\pm\text{future}]$, $[\pm\text{perfect}]$, $[\pm\text{imperfective}]$, and $[\pm\text{progressive}]$ ⁴. The following table gives a survey of the distinctions which are relevant for each of the nine languages; the last column mentions the number of relevant distinctions:

	$\pm$ past	$\pm$ imperf	$\pm$ fut	$\pm$ perf	$\pm$ progr	
Spanish	+	+	+	+	+	5
Portuguese	+	+	+	+	+	5
Italian	+	+	+	+	+	5
French	+	+	+	+	-	4
Greek	+	+	+	+	-	4
English	+	-	+	+	+	4
German	+	-	+	+	-	3
Danish	+	-	+	+	-	3
Dutch	+	-	+	+	-	3

Since each of the attributes can take two values, the maximum number of combinations for each language is 2^n , where n is the number of distinctions. For German, Dutch and Danish it is $2^3 = 8$:

past	fut	perf		past	fut	perf	
-	-	-	spricht	-	-	+	gesprochen hat
+	-	-	sprach	+	-	+	gesprochen hatte
-	+	-	sprechen wird	-	+	+	gesprochen haben wird
+	+	-	sprechen würde	+	+	+	gesprochen haben würde

In this case the maximum number of combinations equals the actual number of combinations. This is not the case for all languages, though. French, for instance, has four relevant distinctions, and hence $2^4 = 16$ possible combinations, but its number of actual combinations is 10^5 :

past	fut	imp	perf		past	fut	imp	perf	
-	-	0	-	parle	-	-	0	+	a parlé
+	-	-	-	parla	+	-	-	+	eut parlé
+	-	+	-	parlait	+	-	+	+	avait aprlé
-	+	0	-	parlera	-	+	0	+	aura parlé
+	+	0	-	parlerait	+	+	0	+	aurait parlé

The reduction is due to the partial neutralisation of the [$\pm$ imp] distinction.

For a similar analysis of the other Euro-languages, see Van Eynde 1991.

In the next sections I will use the term "tense forms" for the combinations of [$\pm$ past] and [$\pm$ future], and the term "aspect forms" for the combinations of [$\pm$ imperfective], [$\pm$ perfect] and [$\pm$ progressive].

4. The compositional core

The determination of temporal relations in a text results from the interaction between the compositionally derived interpretations of the individual clauses, on the one hand, and the dynamically applied principles of discourse organisation, on the other hand. In this section I will concentrate on the compositional core of the format. The dynamic part will be discussed in section 5.

In keeping with the conclusions reached in sections 1 and 2 I will assume that the interpretation of each clause i in a text involves three temporal entities: a point of perspective P_i , a time of reference R_i and the interval for which the tenseless proposition holds, E_i . The point of perspective may be equal to the time of speech, but this is not necessarily so.

Following a proposal in Johnson 1981 I will make a distinction between a treatment of tense (4.1) and a treatment of aspect (4.2).

4.1. Tense concerns the location of R and its relation to P; since R cannot only be a point but also a period, one gets the following possible relations with P:

$<(R,P)$ (anteriority)	$>(R,P)$ (posteriority)	$\supseteq(R,P)$ (simultaneity)

These correspond to the traditional notions of past, future and present; the reason for using terms like "anteriority" instead of the more familiar "past" is that the latter is already used for the tense form and that I do not want to suggest that the relation between tense forms and tense meanings is one-to-one.

There is just one small but significant departure from tradition: simultaneity is not defined as simple identity, but as inclusion. It follows that the time of reference in a clause with a simultaneous interpretation can cover the whole of the time line, and -conversely- that the only way to cover the whole of the time line is by means of a clause with a simultaneous interpretation. This goes a long way in explaining why it is the present that is used in so-called eternal truths and timeless clauses.

The three tense meanings are expressed by means of tense forms and indexical time adverbials:

- anteriority: [+past,-future], yesterday, five days ago
- posteriority: [-past,±future], tomorrow, dans deux heures
- simultaneity: [-past,-future], maintenant

In order to find out which meanings a tense form can express one can apply a grammaticality test: if the tense can be combined with an indexical adverbial of type X in one and the same clause and after the effects of transposition have been undone, then the tense can express X. This yields the following results for the French forms of tense:

- FR [-past,-future] → {simultaneity, posteriority}
- [+past,-future] → {anteriority}
- [-past,+future] → {posteriority}

The "conditionnel", [+past,+future], does not have a temporal meaning of its own; it acquires the meaning of the future in transposed contexts, but out of such contexts it expresses a modal meaning rather than a temporal one.

4.2. Aspect concerns the relation of E to R; since both E and R can be periods of a certain length, the number of their possible relations is higher than between R and P. The exact number can be determined in a deductive way:

Either $E \cap R \neq \emptyset$, and then					or $E \cap R = \emptyset$, and then	
$E \cap R = E$ $E \cap R = R$	$E \cap R = E$ $E \cap R \neq R$	$E \cap R \neq E$ $E \cap R = R$	$E \cap R \neq E$ $E \cap R \neq R$			
$=(E,R)$	$\subset(E,R)$	$\supset(E,R)$	$lo(E,R)$	$ro(E,R)$	$<(E,R)$	$>(E,R)$

It follows that there are seven possible binary relations between E and R ⁶. These relations provide rather straightforward formalisations of such well-known aspectual concepts as

- **perfectivity**: the event is seen as a single unanalysable whole from the point of view of the reference time; *subpart-of*: $\subseteq(E,R)$ or $\subset(E,R)$
- **durativity**: the event is described as going on from the point of view of R; *proper inclusion*: $\supset(E,R)$
- **retrospectivity**: the event is described as completed at R; this is the meaning usually associated with the perfect; *precedence*: $<(E,R)$
- **terminativity**: the event is described as leading up to the time of reference; there is a link here with the "current relevance" meaning of the perfect; *left overlap*: $lo(E,R)$
- **ingressivity/inchoativity**: the event is described as starting at the time of reference; *right overlap*: $ro(E,R)$
- **prospectivity**: the event is described as taking place in the future of the reference time; the futur proche is often taken to express this meaning; *succession*: $>(E,R)$

The six aspect meanings are expressed by the aspect forms and aspectual modifiers, such as duration and time span adverbials. Given the aim and the limited size of this paper I cannot go into the details of the relation between aspect forms and aspect meanings (cf. Van Eynde 1991 for discussion). For the purpose of illustration I only mention the correspondences for the aspect forms in French: FR

[-perf,-imperf]	→	{perfective}
[-perf,+imperf]	→	{durative,terminative}
[-perf,0imperf]	→	{perfective,durative,terminative}
[+perf]	→	{retrospective,terminative}

Prospectivity and inchoativity are expressed by the futur proche.

Taken together the treatments of tense and aspect cover the grammaticalised forms with a temporal meaning in a compositional way; for instance, the meaning of the imparfait is the result of combining the meaning of the past tense [+past,-future] with the meaning of the imperfective aspect [+imperf,-perfect]⁷.

5. The dynamic fringe

The compositional core concerns the temporal relations within individual clauses; for the specification of temporal relations across clause boundaries one needs principles for relating the times of perspective, reference and event of adjacent clauses.

The default case is that the time of perspective equals the time of speech and that both the time of perspective and the time of reference are kept constant: $P_1=S$, $P_i=P_{i-1}$ and $R_i=R_{i-1}$, for $i>1$. Things are not always that simple, though, and for the more complex cases we need rules for identifying the location of P_i , R_i and E_i . In this paper I will not spell out those rules in any detail; I only want to sketch a general format and notation for their formulation. The format has three modules, corresponding to the determination of P_i (5.1), R_i (5.2) and E_i (5.3).

5.1. The time of perspective can be shifted in the future or the past of the speech time, esp. in complement clauses; in that case the point of perspective is usually provided by the time of event of the dominating clause, as in

(22) John said that he would come at 5

TEMPORAL PERSPECTIVE	TENSE	ASPECT
$P_1=S$	$R_1 < P_1$	$E_1 \subseteq R_1$
$P_2=E_1$	$R_2 > P_2$	$E_2 \subseteq R_2$
	$R_2=at\ 5$	

Since E_1 precedes S , P_2 also precedes S , and this triggers transposition of "will" to "would" in the subclause. The tense and aspect meanings are determined by the regular rules for interpreting the simple past and the simple future where the latter results from undoing the transposition in the subclause.

Shifts of the time of perspective are not always marked by verbs of saying, as in (22); they are just as often left implicit and difficult to notice ⁸.

5.2. The time of reference can shift as well; it is the forward movement of the reference time which gets most attention in the literature (cf. Kamp, Rohrer, Dowty, Partee, Hinrichs), and for most authors there seem to be just two possibilities: either there is a forward movement, or there is no movement at all. I think that one should at least foresee the possibility of backward movement as well.

The most explicit way to signal the movement of the reference time is by means of anaphoric time adverbials, such as "two days later" and "five weeks before". They determine the location of the time of reference of the clause in which they appear with respect to the time of reference of some preceding or dominating clause. There are at least three possibilities:

- identity: at that moment, then
- precedence: five weeks before, two minutes earlier
- succession: two days later, deux jours après

Another way of signalling the movement is by means of the grammaticalised forms of tense and aspect. Indeed, following the observations made in DRT, one can make a distinction between the forms which can not trigger any movement, such as the imparfait, and those which can, such as the passé simple.

As applied to

- (24) Kissinger arriva au Caire le sixième juillet.
Deux jours après il partit pour Jérusalem.
Il était très fatigué.

we get the following result

TEMPORAL PERSPECTIVE	TENSE	TEMPORAL ANAPHORA	ASPECT
$P_1=S$	$R_1 < P_1$ $R_1=July\ 6$	—	$E_1 \subseteq R_1$
$P_2=P_1$	$R_2 < P_2$	$R_2 > R_1$ $R_2-R_1=2\ days$	$E_2 \subseteq R_2$
$P_3=P_2$	$R_3 < P_3$	$R_3=R_2$	$E_3 \supset R_3$

The time of perspective is equal to the time of speech and does not move. The tense and aspect meanings are straightforwardly derived from the corresponding forms: $\langle(R,P) \ \& \ \subseteq(E,R)$ for the passé simple and $\langle(R,P) \ \& \ \supset(E,R)$ for the imparfait. The time of reference is first identified as July 6th and then shifted to July 8th by the anaphoric adverbial, "deux jours après". In the third clause there is no shift: $R_3=R_2$ ⁹. The information that E_3 includes R_3 , and therefore also R_2 and E_2 , derives from the aspectual meaning of the imparfait: $E_3 \supset R_3$.

It is interesting to compare this analysis with the one of

- (26) Kissinger arriva au Caire le sixième juillet.
 Dans deux jours il partait/*partit pour Jérusalem.
 Il était très fatigué.

TEMPORAL PERSPECTIVE	TENSE	TEMPORAL ANAPHORA	ASPECT
$P_1=S$	$R_1 \langle P_1$ $R_1=$ July 6	—	$E_1 \subseteq R_1$
$P_2=E_1$	$R_2 \supset P_2$ $R_2-P_2=2$ days	—	$E_2 \subseteq R_2$
$P_3=E_1$	$R_3 \supset P_3$	—	$E_3 \subseteq R_3$

In this case there is no anaphoric information: the anaphoric adverbial is replaced by an indexical one, and the imparfait is not an anaphoric tense here (cf. $R_i=R_{i-1}$), but rather a transposed present. As a consequence it does not have its normal anterior meaning, but rather the meaning(s) which the présent simple has, i.e. a posterior meaning as in the second clause or a simultaneous meaning as in the third clause.

5.3. As far as I can see, natural languages do not provide any explicit means for relating the event times of adjacent clauses; it is, however, possible to compute these relations on the basis of the relations between the respective reference and perspective times. For (24) this computation would go as follows:

TEMPORAL PERSPECTIVE	TENSE	TEMPORAL ANAPHORA	ASPECT	TEMPORAL INFERENCING
$P_1=S$	$R_1 \langle P_1$ $R_1=$ July 6	—	$E_1 \subseteq R_1$	$E_1 \langle S$
$P_2=P_1$	$R_2 \langle P_2$	$R_2 \supset R_1$ $R_2-R_1=2$ days	$E_2 \subseteq R_2$	$E_2 \supset E_1$
$P_3=P_2$	$R_3 \langle P_3$	$R_3=R_2$	$E_3 \supset R_3$	$E_3 \supset E_2$

Clause 1: if $E_1 \subseteq R_1$ and $R_1 \langle P_1$, then $E_1 \langle P_1$, and if $E_1 \langle P_1$ and $P_1=S$, then $E_1 \langle S$.

Clause 2: if $E_2 \subseteq R_2$ and $R_2 \supset R_1$, then $E_2 \supset R_1$, and if $E_2 \supset R_1$ and $E_1 \subseteq R_1$, then $E_2 \supset E_1$.

Clause 3: if $E_3 \supset R_3$ and $R_3=R_2$, then $E_3 \supset R_2$, and if $E_3 \supset R_2$ and $E_2 \subseteq R_2$, then $E_3 \supset E_2$.

These results conform to the intuitions that the arrival in Cairo (E_1) is before the time of speech, that the departure for Jerusalem (E_2) follows E_1 , and that the protagonist feels tired at a time (E_3) which includes the time of the departure.

In a comprehensive treatment the rules for inferencing should be explicitly defined and formalised. This I intend to do in future work.

6. A comparison with the current DRT-approach

The format for the representation of temporal expressions which was developed in the two previous sections is -in a number of ways- similar to the one proposed in Rohrer (1986) and Kamp & Rohrer (ms.). It therefore makes sense to compare them in some detail.

6.1. The DRT format looks as follows:

TEMPORAL PERSPECTIVE	TENSE	PROGRESSIVE	PERFECT
$[\pm\text{PAST}]$	$\{\text{pr,pa,fu}\}$	$\{+,-\}$	$\{+,-\}$

Temporal perspective has two values: +PAST and -PAST. The former indicates that the point of perspective precedes the time of speech ($P_i < S$); in that case the tenses are usually transposed.

The tense values are the semantic present, past and future.

"The feature PROGRESSIVE with the values + or - is intended to express aspectual information. A verb form which contains the feature +PROG denotes a state, a verb form with the feature -PROG denotes an event. Furthermore a verb form with -PROG "brings the action forward", or in more technical jargon, the denoted event lies after the reference point. Verb forms with +PROG don't 'bring the action forward'." [Rohrer 1986, 80]

"The feature PERFECT also has the two values + and -. It is used to capture the notion of a "perfectum praesens". We want it to denote a state which is the result of a preceding event." [ibid.]

The first two features -Temporal Perspective and Tense- are very much like their equivalents in my format; there are only some minor differences.

As for temporal perspective, its value in DRT only says whether or not the point of perspective precedes the point of speech; in my system it is explicitly identified with some specific time introduced before, and whether that time precedes S is a matter of inferencing. Since the $[\pm\text{PAST}]$ information is probably all one needs to know for the treatment of the forms of tense and aspect, DRT's treatment may be justified on this account. However, if the objective is to make the temporal structure of a text explicit, one probably needs to be more specific about the location of P .

As for tense, the main difference is that I define simultaneity (semantic present) as inclusion, rather than as coincidence, thus offering an explanation for the use of the present in so-called eternal truths (cf. 4.1).

The more important differences between both treatments concern the features Progressive and Perfect.

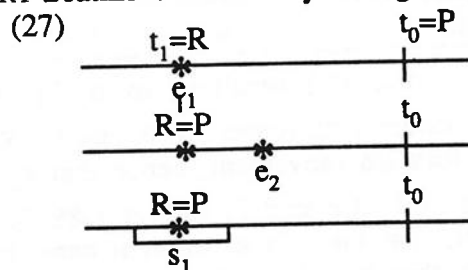
The feature PROGRESSIVE concerns the "forward movement of time", like TEMPORAL ANAPHORA in my treatment, but there are three major differences. First, I distinguish more values than the two which DRT provides; apart from forward movement [-PROG] and lack of forward movement [+PROG] I also foresee the possibility of backward movement, thus providing a way for accomodating such anaphoric adverbials as "two days earlier" and "five weeks before". Second, the value of PROG depends to a large extent on the aspectual type of the clause (event vs. state), whereas the ANAPHOR value is determined only by the forms of tense and aspect and the anaphoric time adverbials. Third, whereas PROG concerns the relation between R and e_i or s_i (where R is provided by e_j for $j < i$), ANAPHOR concerns the relation between R_i and R_{i-1} . The determination of the relation between adjacent E's in my alternative is left to the INFERENCING module.

As for the feature PERFECT, it is subsumed under my treatment of ASPECT. The latter is more comprehensive than the former, since it also includes a treatment of the [$\pm$ imperfective] and the [$\pm$ progressive] distinctions. In contrast to DRT, I claim that these distinctions are not only relevant at the level of discourse (cf. their different values for PROG) but also at the level of the individual clause.

6.2. To illustrate the differences between both treatments I will compare their analysis of

- (26) Kissinger arriva au Caire le sixième juillet.
 Dans deux jours il partait/*partit pour Jérusalem.
 Il était très fatigué.

The DRT treatment has already been given in section 2:



In terms of the four features it can be summed up as follows:

TEMPORAL PERSPECTIVE	TENSE	PROGRESSIVE	PERFECT
[-PAST]	past	-	-
[+PAST]	pres	-	-
[+PAST]	pres	+	-

In terms of my treatment it would be represented thus:

TEMPORAL PERSPECTIVE	TENSE	TEMP. ANAPH.	ASPECT	TEMPORAL INFERRING
$P_1=S$	$R_1 < P_1$ $R_1 = \text{July 6}$	—	$E_1 \subseteq R_1$	$E_1 < S$
$P_2=E_1$	$R_2 > P_2$ $R_2 - P_2 = 2 \text{ days}$	—	$E_2 \subseteq R_2$	$E_2 > E_1$
$P_3=P_2$	$R_3 \supset P_3$	—	$E_3 \subseteq R_3$	$E_3 < E_2$

The main differences concern the second clause, more specifically the location of R . In (27) it is provided by e_1 and equal to P ; the fact that e_2 follows e_1 is due to the relation between e_2 and R . In my treatment R includes E_2 , and the fact that E_2 follows E_1 is attributed to the relation between R and P . It follows that I have to stipulate two meanings for the present (simultaneity and posteriority), but there is plenty of independent motivation for that: many linguists and logicians, including Kamp and Rohrer, have postulated a posterior meaning for the present, next to its simultaneous one (cf. 4.1).

The difference between the two treatments may seem insubstantial, at this point: both $P=R < e_2$ and $P < R_2 E_2$ predict that $P < E_2$, and since $P=E_1$, that $E_1 < E_2$.

Let us now take a slightly more complex example, though:

(29) Kissinger arriva au Caire le 6 juillet.

Dans deux jours il aurait visité 5 pays.

Applying the rules for the representation of the passé simple and the conditional II, DRT would derive the following values:

TEMPORAL PERSPECTIVE	TENSE	PROGRESSIVE	PERFECT
[−PAST]	past	−	−
[+PAST]	fut	+	+

The TENSE value specifies that the state (s_2) resulting from the preceding event of visiting 5 countries is in the future with respect to P , and the PROGRESSIVE value predicts that there is no forward movement, hence that s_2 does not follow R , but rather includes it (it is explicitly said in Rohrer 1986 that [+Perf] tenses are [+Prog]). Notice, however, that there is a problem here. Indeed, if s_2 is in the future of P and includes R , then it follows that R has to follow P , but if R is determined by e_1 , as normal DRT practice dictates, then it has to be equal to P . One could, of course, claim that normal DRT practice in this case is overruled and that R has to be put in the future of P (and e_1), but this would amount to a rather cavalier treatment of the indexical adverbial: in (26) it would specify the distance between R and e_2 , and in (29) it would specify the distance between P and R .

In my treatment, on the other hand, the indexical adverbial can consistently be interpreted as specifying the distance between P_i and R_i :

TEMPORAL PERSPECTIVE	TENSE	TEMP. ANAPH.	ASPECT	TEMPORAL INFERENCING
$P_1=S$	$R_1<P_1$ $R_1=July\ 6$	—	$E_1\subseteq R_1$	$E_1<S$
$P_2=E_1$	$R_2>P_2$ $R_2-P_2=2\ days$	—	$E_2<R_2$	$E_2?E_1$ ¹⁰

The conditionnel II in (29) is a transposed future perfect, which gets its regular interpretation ($R>P$ and $E<R$) and which can be unified in a straightforward way with the indexical time adverbial.

7. Conclusion

The resulting treatment has a number of attractive properties:

- it provides a format for the representation of temporal relations in texts, in terms of which the contributions of the individual temporal expressions can be described;
- the text representations are derived and interpreted in a compositional way; this holds both for the processing of texts (cf. the distinction between a compositional core and a dynamic fringe) and for the analysis of the grammaticalised forms of tense and aspect (cf. the distinction between a treatment of tense and a treatment of aspect);
- it provides a conceptual framework for the comparison of the tense and aspect systems of different languages.

Notes

1. "Event" is used here as a cover term for state, process and event. Throughout the paper I will follow the convention that E or E_i stands for the interval at which a tenseless proposition holds, regardless of whether this proposition is a state, a process or an event. If the distinction matters, I will use lower case letters, such as e for event, and s for state.
2. Pragmatic considerations lead one to assume that E_5 precedes E_4 , since the introduction will presumably have occurred before the writing of the letter, and that E_4 precedes E_2 , since the writing will no doubt have occurred before the start of the journey; this information, though, is not so much based on linguistic knowledge (the meaning of the pluperfect) as on common sense entailment.
3. Since this principle does not mention tense, it will not only be applied when the second clause is in the passé simple, but also when it is in the imparfait. In that case there is a problem, since the analysis might well predict that E_2 follows, rather than includes E_1 .

Indeed, if $R_1<S$ $E_1=R_1$ (passé simple)
 $R_2>R_1$ (forward movement)
 $R_2<S$ $E_2\supset R_2$ (imparfait)

then it is not impossible that $E_2>E_1$. For that reason Dowty requires that R_1 immediately follow R_{i-1} .

4. A more comprehensive survey should also include the futur proche (aller+infin, to be going to+infin) and -perhaps- its past counterpart (venir de+infin). See also note 5.

5. Taking into account the futur proche would lead to the addition of two more forms: the présent "va parler" and the imparfait "allait parler". Similarly, one could add two forms for the "immediate past": "vient de parler" and "venait de parler".
6. Allen 1984 distinguishes no less than 13 binary relations between intervals. They result from differentiating between various types of sub-intervals (initial, final, middle) and precedence (immediate vs. non-immediate). Since these further distinctions are not grammaticalised in the Euro-languages I will not adopt them here.
7. There are a few exceptions to this scheme; the present perfect in German, Dutch and French, for instance, does not only express retrospective simultaneity, as predicted by the rules for the present and the perfect, but also perfective anteriority, which is one of the regular meanings of the simple past.
8. Rohrer 1986 discusses several examples in which the point of perspective is not provided by the saying event but by some other time of reference or event in the preceding text. There will, no doubt, be some constraints on the choice of an anchor for the point of perspective, but I prefer to leave the investigation of this topic for further research.
9. In Vet and Molendijk 1986 it is argued that $R_i = R_{i-1}$ is a constant feature of the French imparfait.
10. The relation between E_2 and E_1 cannot be determined unequivocally on the basis of the tense and aspect forms; pragmatic knowledge tells us that E_2 properly includes E_1 , but this is a matter of common sense entailment, rather than of inferencing from linguistic information.

References

- Allen, J.: 1984, Towards a general theory of action and time, *Artificial Intelligence* 23, 123-154.
- Dowty, D.: 1979, *Word meaning and Montague grammar*, Reidel, Dordrecht.
- Dowty, D.: 1986, The effects of aspectual class on the temporal structure of discourse: semantics or pragmatics?, *Linguistics and Philosophy* 9 (1), 37-61.
- Hinrichs, E.: 1986, Temporal anaphora in discourses of English. *Linguistics and Philosophy* 9 (1), 63-82.
- Johnson, M.: 1981, A unified temporal theory of tense and aspect, in Tedeschi and A. Zaenen (eds.), *Tense and aspect*, Syntax and Semantics XIV, Academic Press, New York.
- Kamp, H. and Rohrer, C.: 1983, Tense in texts, in R. Bäuerle et al. (eds.), *Meaning, use and interpretation of language*, Walter de Gruyter, Berlin.
- Kamp, H. and Rohrer, C., *Temporal reference in French*, Unpublished manuscript, Universität Stuttgart.
- Mourelatos, A.: 1978, Events, processes and states, *Linguistics and Philosophy* 2, 415-434.
- Partee, B.: 1984, Nominal and temporal anaphora, *Linguistics and Philosophy* 7 (3), 243-286.
- Reichenbach, H.: 1947, *Elements of symbolic logic*. New York.
- Rohrer, C.: 1986, Indirect discourse and "Consecutio temporum", in V. Lo Cascio and C. Vet (eds.), *Temporal structure in sentence and discourse*, Grass 5, Foris, Dordrecht.
- Van Eynde, F.: 1991, The semantics of tense and aspect, in L. Damas et al. (eds.), *Lecture Notes in Artificial Intelligence* 476, Springer Verlag, Berlin.
- Vet, C. and Molendijk, A.: 1986, The discourse functions of the past tenses of French, in: V. Lo Cascio and C. Vet (eds.), *Temporal structure in sentence and discourse*, Grass 5, Foris, Dordrecht.

Natural-Language Content: A Proof-Theoretic Perspective A Preliminary Report

Dov M. Gabbay and Ruth M. Kempson
Imperial College and School of Oriental & African Studies

1 Preliminaries

In this paper, we report on work in progress towards the goal of building a formal model of the general process of utterance interpretation. We introduce the background logical framework on which the account depends, we sketch out the account itself as we have currently developed it, we apply it to anaphora and wh-binding; and we show how it affects underlying assumptions about natural language. The heart of our claim is that natural language needs to be seen in a procedural perspective, not merely in reconstructing the process of utterance interpretation, not merely in the concept of information flow as presented by natural language sequences, but in reconstructing the very concept of the human faculty for language, itself embedded in a procedural logic framework (LDS).

First, we lay out the analysis informally. We see utterance interpretation as involving a pair of reasoning tasks, one embedded in the other (cf. Sperber and Wilson 1986). From the uttered string, a hearer has to reason to some overall conclusion in a predicate logic using some additional "context" set of premises (this the implicature phenomenon). As part of this task she has to decide what proposition the speaker has directly expressed by his utterance, using as input the information given by the individual words in the sequence uttered. In arriving at this decision, she is licensed to make additional assumptions as indicated, eg by anaphoric expressions. We argue that this process involves reasoning in a natural deduction system, a parsing logic which maps input sequences of data recovered from the words onto corresponding logic structures. In this paper, we restrict our attention to this parsing task, though we grant that decisions made at this level may be affected by requirements imposed by the need to succeed in the overall goal of establishing implicatures of one sort or another. Though the parsing process involves decisions to be made, and assumptions to be added, it is nonetheless, we argue, tightly directed by the nature of the input information provided by the lexical content of the words used. These give precise instructions on how to build the structure the speaker intends the hearer to recover as a result of such choices. The process of establishing an interpretation is then a step by step process of parsing as driven by these directions.

The mechanics of our approach works as follows. We start with a logic discipline (in this case LDS, described below) in which the notions of databases and proof theory based on such databases are clearly defined. This logic is taken to be an instrument for building a parser for natural language input, which uses the logic in an intrinsic way. The linguistic input is taken word by word to build a database in our logic. The construction process is our parser. The process can fail or succeed. If it fails, then the linguistic input is unacceptable linguistically. If it succeeds, then the linguistic input is accepted and the database which was constructed reflects the logical content of the linguistic input.

The process is driven by the linguistic input, each word having two associated specifications, a declarative component and a procedural component. If the input is given to the construction process, a partial database is already in existence (from previous inputs or the empty database right at the beginning), and the declarative component of the words tells the process what

additional data to put in and the procedural component affects the construction process itself. Some English words may not have declarative content at all but only procedural meaning for the construction process. The construction process can use the logic in an essential way in the sense that reasoning with the partially constructed database is part of the construction process and can be referred to by the procedural component of the linguistic input. The components of the linguistic input are considered as part of the lexicon.

The construction process builds a database which reflects the logical meaning of the input. If the construction is not unique but has several options, this will correspond to several readings of the input. If a certain reading is not acceptable, then we expect, if the construction process is sound, that a database corresponding to that reading cannot be constructed. The construction process is, then, a parser. Furthermore it uses logic to construct such a database, resolving traditional problems (anaphora) as it proceeds.

As things turn out, we shall see that this construction process has all the hallmarks traditionally associated with a grammar. It licenses strings of lexical items as wellformed; it pairs each such string with a representation in a logical system said to constitute its interpretation; and it defines all possible mappings from string to interpretation step by step. As a parser it is by no means unique. Like other parsers (Pereira and Warren 1983), it uses logical proof as part of the process of parsing and interpreting the input. Like other parsers that seek to model the human process of parsing, it is incremental (Pereira 1985, Marcus 1980, etc) We do not have the full input and then combinatorially examine it, but get the input one by one with possibly one look ahead. Nevertheless it is unique in two ways, the one a symptom of the other. Though some words contribute premises with descriptive content to the process of the parse in familiar ways, other words have procedural meaning and no declarative content, this leading to a new concept of information content for natural languages. As an explanation of the language faculty, it is we believe unique in defining a natural language as a parsing device, the concepts of wellformedness and interpretation defined procedurally in terms of the way natural-language sequences enable information to be built up on line. Furthermore, it is essentially a logic, given an enrichment of that notion to include a procedural element as in the natural deduction framework of the Labelled Deductive Systems (LDS) of Gabbay 1989.

2 LDS and Linked Databases

The LDS conception of a logic is a discipline with the following properties:

- (i) The notion of a formula is replaced by the notion of a declarative unit, which is a pair; a formula and a label. The label is taken from an algebra of labels. All rules of inference make critical reference to the labels, and manipulate both formulas and labels. The most useful rule for our purposes is Modus Ponens with functional application on the labels, namely:

$$\alpha : P \rightarrow Q$$

$$\beta : P$$

$$\alpha(\beta) : Q$$

- (ii) The static concept of a language is replaced by the procedural concept of a logic framework. The logic framework models the linking between one individual local database and another, specifying control features which dictate the steps licensed at any stage of reasoning, either within the local database, or in moving from one database to another. As part of this discipline, we define a concept of linking between a pair of databases. Informally, a pair of databases can form a database with subparts linked by sharing a variable as the result of some substitution process. The

notion of declarative units generalises the traditional notion of a formula, and the notion of linked databases generalises the traditional notion of a classical logical theory. The linked databases represent logical information in the manner analogous to classical theory.

We start by giving a formal definition of labels, the notions of declarative units and linked databases. We assume the logical language contains a set of pure, atomic labels and formulas built up from e, t , and $\rightarrow$. Both labels and wffs can contain variables, x, y, z .. There are several types of variables in the labels (details later). We adopt the following mutually recursive clauses.

1. A declarative unit is a pair

$$t : A$$

t a pure label, A a wff, called the body of the unit. We say t appears in $t:A$.

2. A database with label s and data f is either a declarative unit

$$t_1 : A_1,$$

with s identified as t_1 and f identified as the function

$$f: \{1\} \rightarrow \{A_1\}$$

or is the pair (s, f) , with s a label and f a function with domain the numbers $\{1, \dots, k\}$, for some k , such that:

$f(i) = (t_i, g_i)$, where $f(i)$ are databases with labels t_i and functions g_i

We say a label t appears in (s, f) , if either $t = s$, or t appears in s , or $t = t_i$, or t appears in t_i , or t appears in $f(i)$ for some i .

3. Let $\Delta_1 = (s, f)$ and $\Delta_2 = (t, g)$ be databases with labels $\alpha(x)$ and $\beta(y)$ appearing in them respectively, containing the variables x and y .

Then the pair $(r, \text{LINK}(\Delta_1, \Delta_2, \alpha, \beta, z))$ is a declarative unit with label r and the body is in the form of the pair $(\Delta_1(\alpha/z), \Delta_2(\beta/z))$ where z is the most general unifier of $\alpha(x), \beta(y)$.

We say t appears in the new database if either t appears in r , or $t = r$, or $t \neq \alpha$ and $t \neq \beta$ and t appears in Δ_1 or in Δ_2 , or in z .

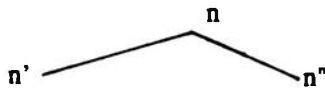
4. A label is either a pure label or a database.

5. We sometimes refer to databases informally as (meta)boxes.

2.1 Implementation of the construction process

We now describe a process for constructing simple linked databases which is sufficient to analyse natural language input. The linking structure in these databases is a tree. The nodes of the tree are databases:

Thus:



describes three databases. With n' and n'' linked into n . Each database has a label and a sequence of data in it. Thus at n

we have a database:

$$s: t_1:A_1, \dots, t_k:A_k$$

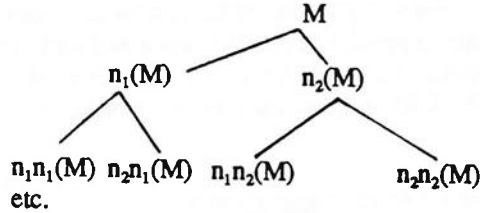
also written as $s:f$, $f(i) = t_i:A_i$.

s is the database label and $t_i:A_i$ the declarative units in it. The labels s and t_i may be composite labels, i.e.

$$s = (s', s''), s' = \text{time}, s'' = \text{other information}$$

Similarly $t_i = (t_i', t_i'')$ with t_i' being annotation, eg. "use me last" etc. (When we need to distinguish them we shall do so as CLabel(X), the Content label of X, and RLabel(X), the Resource label of X.) Thus the nodes n_1, n_2 , are not labels but vehicles for locating the databases.

It is convenient to implement the tree via successor functions. If M is the root of the tree (i.e. $n = M$), we allow successor functions n_1, n_2 , and this forming the tree by iterating these functions

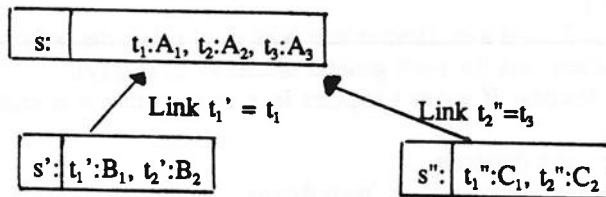


The database at each node x has the form, $s_x: f_x$. f_x is a function $f_x(1), \dots, f_x(j)$.

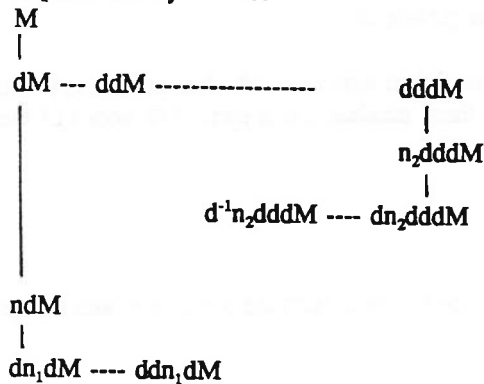
We want to incorporate the data into the tree. We do this via a successor function $d, d^2, \dots, d^j$. Thus at node x there sits the database

s:	$t_1:A_1, \dots, t_n:A_n$
----	---------------------------

The successor nodes $d(x), \dots, d^j(x)$ will carry the data $t_i:A_i$. There will also be nodes $d^{-1}(x)$ where linkage comes in the middle of the database. Special functions Label(x), Formula(x) will give the label and the data. Thus



is presented by the tree



Label $M = s$

Database at M are $t_1:A_1, t_2:A_2, t_3:A_3$

These are coded via $dM, ddM, dddM$, to be itemised as d^1M, d^2M, d^3M . Thus:

Label $(d^1M) = t_1$

The n_1 successor gives the database of s' . Since this is linked to t_1 , whose place is dM , we put n_1, dM in the tree.

The database at $x=n_1d^1M$ is put at d^1x and d^2x ,

Label $(d^1n_1d^1M) = t_1'$.

Label (d^2n, d^1M) = t_2'

The link is known because it is done through $n_1 d^1 M$. The equation $t_1' = t_2$ can be written as

In general we want $\text{Label}(d_n, y_x) = \text{Label}(x)$. Restrictions on the nesting of the linked item will emerge through the construction process.

 $t_3: A_3$
$$t_1^n: C_1 \text{ ————— } t_2^n: C_2$$

To express that we attach $t_2":C_2$ to $d_n d^m M$ and go "backwards" for $t_1":C_1$ i.e. put it at $d^1 n_1 d^3 M$. Thus

$$\text{Label} \quad \text{dn}_2 \text{d}^3\text{M} = t_2^n$$

Formula $dn_2d^3M = C_2$

Label $d^{-1}n_2d^3M = t_1''$

Label	$d^{-1}n_2d^3M$	C_1
Formula	$d^{-1}n_2d^3M = C_1$	

The linkage requires $\text{Label}(\text{dn}_2\text{d}^3\text{M}) = \text{Label } \text{d}^3\text{M}$
still satisfying the equation $\text{Label}(\text{d}^i\text{n}_j\text{y}\text{x}) = \text{Label}(\text{x})$

We have defined how one node may succeed another. We may also have tree configurations in the labels themselves. The successor functions n- and d- apply not only to nodes but to labels, giving rise to databases such as:¹

$$s_i: \quad t_1:A_1 \quad t_2:A_2 \quad \boxed{t_3: x_1:B_1 \ x_2:B_2} : A_3$$

Suppose we have some database M containing:

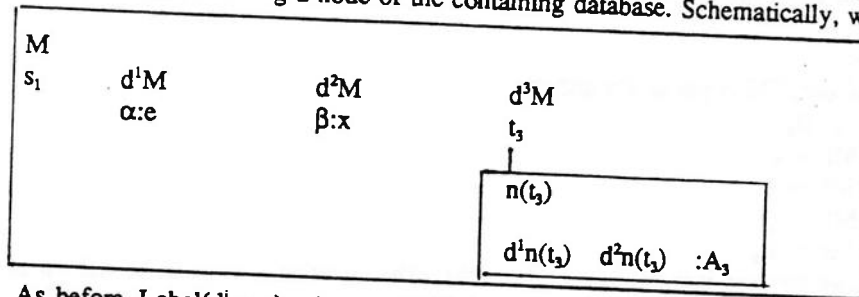
$$\begin{array}{ccc} d^1M & d^2M & d^3M \\ t_1:A_1 & t_2:A_2 & \text{Formula}(d^3M) = e \\ & & \text{Label}(d^3M) = t_1 \end{array}$$

We may apply the n -function to t_1 , yielding $n(t_1)$, and add declarative units to that created node, each with a label and a formula:

 $n(t_3)$

d^1nt_3	d^2nt_3
Label (d^1nt_3)= x_1	Label (d^2nt_3)= x_2
Formula(d^1nt_3)= B_1	Formula(d^2nt_3)= B_2

Viewing the result from the perspective of inside outwards, the result is a label for a database, that complex itself labelling a node of the containing database. Schematically, we have



As before, $\text{Label}(d^i n_i y x) = \text{Label}(x)$. Here $i = 0$, $y = 0$.

Definition:

A partial database is a tree T with successors n_i , d , d^1 , and labelling and formula functions, Label , Formula satisfying the following:

- (1) if $t \in T$ and $s < t$, then $s \in T$ (i.e. there are no gaps)
- (2) For any node of the form $n_i x$, there exists an α such that $\text{Label}(\alpha n_i x) = \text{Label}(x)$ where α is a sequence of successors.
- (3) The Labelling function can give as a label either a composite atomic label of the form $t = (s, s')$, or a label which is in itself a database of the form $t : \Delta$, in which case Δ is generated by a tree with root nt , where n is the successor function specially reserved for this.

3. The Dynamics of Utterance Interpretation

What we are proposing is that the two formal notions of labelling and linking, combined with the goal-directed construction process of building labelled databases using input from the lexicon, enable us to reconstruct the way in which information is built up step by step through the process of utterance interpretation, using only information specified in the lexicon. We assume, as do others, that parsing over the types (taken as propositions) can lead us from information in the lexicon to a conclusion of the form $\alpha : t$ by using Modus Ponens as an integral part of the labelling discipline. Premises all take the form $\gamma : x$, γ an expression in some predicate calculus, x a formula made up of some combination of e , t , $\rightarrow$. With the types as propositions labelled by the corresponding construct of predicate logic, the task of utterance interpretation becomes a goal-directed exercise of building a database which will establish a conclusion of type t with a structured label α , where α represents the intended content of the speakers utterance. As required by the proof discipline, α reflects the mode of combination as given by the premises provided by the lexicon.

Each word in the sentence uttered has to be used; and the result has to be some database based on the information given by those words, that database should be able to prove $\alpha : t$. Schematically:

α John : e
 smiled : $e \rightarrow t$
 smiled(John) : t

In the spirit of the LDS enterprise, there is, however, more to the proof enterprise than just a set of premises. There are at least three types of lexical specification:

- (i) There are the cases which provide additions to the database as premises
 John : e
 touch: $e \rightarrow (e \rightarrow t)$

(ii) There are cases which provide additions to the database plus some control specification on the way the proof structure is built up. Some of these side conditions are simple, eg dictating identification of an introduced variable (anaphoric expressions). Others determine properties of other entities in the database (eg the selectional specifications associated with verbs). Yet others are complex, dictating the building of additional databases within the overall proof structure, eg determiners.

(iii) There are cases which do not themselves introduce any premise into the database but solely provide control specification on the way the database is built up - eg who, which imposes a goal specification on the containing database (namely that it contain an open variable to be linked), without itself providing any addition to the database directly.

The overall goal is that from a certain input sequence of words, $\langle w_1, \dots, w_n \rangle$, the hearer must construct a (possibly linked) database with internal proof structure. Only some of those words provide premises. Those premises as database, together with control features provided by the other words, dictate the building of a proof structure capable of proving a conclusion of $\alpha:t$, α labelling the type t . This database is then available for the hearer to reason on direct from this database. A further mapping can derive a predicate logic database representing its logical content, if so desired.

There are two further things we shall need. We assume we have a notion of one database proving another, although we are not giving the rules now. The definition of construction will use the notion of proof. Intuitively, think of it as $\Delta \vdash \Gamma$ iff the "meaning" of Δ entails the "meaning" of Γ . Any formal definition of proof of a database must agree with our intuitive meaning. We also need to define a construction function that builds the requisite databases. For this, we define an Input function which takes input from the lexicon and builds a tree configuration using the two successor functions. We assume the construction proceeds as follows. It starts from the first word in the sequence opening a database M , building a sequence of nodes to be filled each with a formula and a label. This input function takes a set of nodes (T), a set of labels ($Label$), a set of formulae ($Form$), and the specification given in the lexicon for some word w , and maps that configuration onto some new labelled configuration ($T', Label', Form'$):

Input($T, Label, Form, w$) = ($T', Label', Form'$)

All labels of T' are bipartite, containing Resource labels, $RLabel$, which control the derivation and Content labels, $CLabel$, which are the lexical items themselves. It is the $CLabels$ which combine in tandem with steps of Modus Ponens over the types. The $RLabels$ merely provide extra control on that process.

Each word in the lexicon W , has a two-part specification. There is a Declarative specification which contains the $CLabel$ and a Formula, to be added respectively to T' by the function $Label(X)$, $Formula(X)$ for some node X . And there is a Procedural specification which is a function updating (T, L, F) into T', L', F' in varying ways.

Where the Declarative specification of a word w is nonempty, as with all lexical "content" words, the input function can be seen as carrying out the update process in three steps.

- (i) It will modify the tree at some node $d^i x$ at its rightmost edge (i.e. " d " is a right successor and any " n_i " are left), by adding a node $d(d^i x)$, mapping T into a new configuration T' ,
- (ii) It will fill that node with a declarative unit formed with the declarative specification of the leftmost word w_k not yet processed out of the sequence $\langle w_1, \dots, w_n \rangle$:

$C\text{-Label}(d(d^i x)) = C\text{-Label of } w_k$

$Formula(d(d^i x)) = Formula of w_k$

thereby modifying L, F , in the configuration (T, L, F) to yield the configuration (T', L', F') .

Label(d^1M) = John

Form (d^1M) = e

With touches as the next word, the d-successor function builds the second node d^2M , to be filled with the label touch, the formula $e \rightarrow (e \rightarrow t)$

Label(d^2M) = touch

Form (d^2M) = $e \rightarrow (e \rightarrow t)$

The procedural specification of touches provides, in addition, a rich amount of information with which to further update the partial tree so far established. It assigns information about the labelling of its two adjacent nodes - that the immediately preceding node of the form $\alpha:e$ be used last in the chain of reasoning to lead to a formula t (ie is its logical subject), that the immediately following node of the form $\alpha:e$ be used first in such a chain of reasoning (ie is its logical object), that the last node be labelled with an individual variable (singular), and that the mother node labelling the database be identified as a temporal variable denoting a period of time to include the time of utterance (tense). (We list such resource labelling for the premises in a right-hand column of the metabox.) The input function adds this information to the tree, building new nodes if necessary, labelling existing nodes as directed. The result:

M CLabel(M) = s_i , $\Theta s_i \supset s_{ut}$

d^1M

CLabel(d^1M) = John

Form (d^1M) = e

RLabel(d^1M) = {use me last}

d^2M

CLabel(d^2M) = touch

Form (d^2M) = $e \rightarrow (e \rightarrow t)$

d^3M

RLabel(d^3M) = {use me first}

We now have a node d^3M which needs to be filled. The remaining word may provide the required information:

CLabel(d^3M) = Mary

Form (d^3M) = e

The proof rules themselves then create the further nodes:

d^4M , CLabel(d^4M) = touch(Mary), Form(d^4M) = $e \rightarrow t$

d^5M , CLabel(d^5M) = touch(Mary)(John), Form(d^5M) = t

The task of the construction is complete. From the node M , using here the d-successor function alone, the input function and the logic together have constructed a database yielding a conclusion s_i , $\alpha:t$, α a function on labels provided by the words used.

Note that the lexical entry for the verb in essence does all the work. We give a lexical specification for touches (assuming that the entry from the lexicon is to some node d^1x):

touches touch $e \rightarrow (e \rightarrow t)$ RLabel(d^1x) = {use me last}

RLabel($d^{i+1}x$) = {use me first}

RLabel($d^{i+1}x$) = {CLabel($d^{i+1}x$) = $\{y, y \in \{y_1, \dots\}\}$ }

RLabel(x) = {CLabel(x) = $\{s_i, \Theta s_i \supset s_{utterance}\}$ }

We have chosen to capture the concept of subject and object directly on the proof-structure. "Use me last" is an annotation in the label such that if we have an assumption (use me last, t): e it can combine in modus ponens only with $\beta:e \rightarrow t$. Thus if $\beta':e \rightarrow (e \rightarrow \dots (e \rightarrow t) \dots)$ is looking for $\alpha:e$ to "combine" with, α cannot be used if it has a 'use me last' component.²

Consider now a sequence of words for which we need the n-successor function, adding one database to another:

(2) John touched Mary, who smiled.

As before we give the metabox display and work towards it:

M	{CLabel(d ¹ M):Form(d ¹ M),...} ⊢ s _i , α:t		
	Θ s _i < s _{utt}		
d ¹ M	John :	e	use me last
d ² M	touch :	e → (e → t)	
d ³ M	Mary :	e	use me first
d ⁴ M	touch(Mary) :	e → t	
d ⁵ M	touch(Mary)(John) :	t	

α = nd ³ M
{CLabel(d ¹ nd ¹ α):Formula(d ¹ nd ¹ α),..., (CLabel(d ⁿ yα)=u:e),...} ⊢ β(u):t, human(u)
LINK(CLabel(d ⁿ yα) = CLabel(d ³ M))
u = Mary
d ¹ α Mary : e
d ¹ α smile : e → t
d ² α smile(Mary) : t

We take John touched Mary exactly as before.

M		
d ¹ M	d ² M	d ³ M
		Label(d ³ M) = Mary

d¹M, d²M and d³M all labelled and filled. Subsequently, we have the option of constructing a new node:

n(d³M)

The decision to construct such a node is driven, as usual, by the leftmost word unused in the sequence - who. Who has no declarative specification of its own. Who as a relative indicates subordination of a particular sort: the clause to be built must be open with respect to some variable, to be linked online.³ In our terms here, it has the Procedural specification that the n-successor function add a node to the current point, as always the rightmost edge. Moreover, this node must have added to it the specification that it label a database capable of proving a conclusion β(x):t, x some metavariable, its value to be selected. Accordingly, to d³M, a new node n(d³M) is added, =α, and the requisite control specification listed with it. Since α is a node which results from the n-successor function applying to some mother node, it must be linked, and we choose:

LINK(CLabel(dⁿyα) = CLabel(d³M))
viz: x = Mary

With x = Mary, we proceed. We fill the database via the d-successor function. Initially we build d¹α, and fill it with the declarative specification of the leftmost word available:

CLabel_w = smile
Form_w = e → t

These specifications are entered into the node d¹α by the Label and Formula functions respectively. The procedural specification of this word smiles leads to a two-fold modification of the database α:

- (i) the label of α is a temporal variable to be chosen, its value to precede the time of utterance
- (ii) there is a node d¹α preceding d¹α in the database, a node to be marked with the Resource specification that it be last to combine with the chain of premises e^a → t to yield t. Accordingly the

construction function adds the node $d^1\alpha$. This node has to be filled, and by assumption we use the information of the target specification, viz. that the database contain a node labelled Mary, of formula e , thereby providing the necessary premise to complete the target. The target complete, the node α is complete. The node α complete, the mother node d^3M is complete. And, with no more words presented, the database M is thereby complete. And so the target can be successfully reached within the specification provided by the input sequence.

Under what conditions can the interpretation go beyond such lexical specification? The answer is - "In the presence of special operators, viz. anaphoric expressions". We give the derivation of:

(3) John touched Mary. She smiled.

We propose that all anaphoric expressions (pronouns, anaphors, definite NP's alike) introduce metavariables which - instead of giving a data item as input - give special instructions on how to identify the input which it represents. Such devices are wellknown from many areas. We say anaphoric expressions induce a premise of type e , labelled by a meta-variable

$u: e$

which by definition must be resolved by identifying some other label online during the construction task. The process is displayed in (3') below, where we assume the two strings are parsed as a concatenation of two formulas of type t , leading to a single goal,

viz: $t+t$. Formally a sequence $A_1...A_n \vdash C+D$ iff for some k , $A_1...A_k \vdash C$ and $A_{k+1}...A_n \vdash D$.

(3')

s_0	$f_{INPUT}\langle w_1,...,w_n \rangle = \{Label(w_1):Form(w_1),...,Label(w_n):Form(w_n)\} \vdash \alpha(Label(w_1),...,Label(w_n)):t+t$		
	$f_{INPUT}\langle w_1,...,w_k \rangle = \{Label(w_1):Form(w_1),...,Label(w_k):Form(w_k)\} \vdash \alpha(Label(w_1),...,Label(w_k)):t$		
s_1			
d^1s_1	john	:	e use me last
d^2s_1	touch	:	$e \rightarrow (e \rightarrow t)$
d^3s_1	Mary	:	e use me first
d^4s_1	touch(Mary)	:	$e \rightarrow t$
d^5s_1	touch(Mary)(John)	:	t
	$f_{INPUT}\langle w_{k+1},...,w_n \rangle = \{Label(w_{k+1}):Form(w_{k+1}),...,Label(w_n):Form(w_n)\} \vdash \alpha(Label(w_{k+1}),...,Label(w_n)):t$		
s_2			
d^1s_2	u_{pron}	:	e female(Θu_{pron}) $\Theta u_{pron} \in s_2$
	CHOOSE $u_{(she)} = Mary$		
d^2s_2	Mary	:	e
d^3s_2	smile	:	$e \rightarrow t$
d^4s_2	smile(Mary)	:	t

The relevant additions are in the second database, s_2 . In s_2 , with proof goal as before, the pronoun she is assigned a metavariable as an initiating assumption, an assumption which has to be resolved in the light of the locality restriction stated in the side condition. This locality condition is listed as:

$\Theta u_{pron} \in s_2$

This is to be read as shorthand for the locality restriction:

In a set containing one major premise $\beta:e^n \rightarrow t$, and a sequence of minor premises, $\langle \alpha_1:e_1, \dots, \alpha_n:e_n \rangle$, a premise of the form $\alpha_i:e_i$, α_i a pronominal metavariable, may not be instantiated as:

$$\Theta \alpha_i = x, x:e_{i+k}, 1 \leq k \leq n-i.$$

In the case here, the assumed variable is resolved by choosing as its instantiation the label from the previous database (satisfying its stated side condition). The label 'Mary' is substituted as an instantiation of u_{pron} and the interpretation of box s_2 as a whole is 'Mary smiled'. Note that this process is compatible with computer science practice of putting data in a database or memory location through a pointer of where to find it. Notice how the difference between (2) and (3) lies solely in the way the information is built up. In (2) the n-successor function, and linking through who, provide the means of adding one database to another. In (3) the instantiation function chooses a value for a metavariable from some earlier database, and so provides as output the same interpretation. We have two cases with identical truth-theoretic content but a different structured database encapsulating that content, the result of a different construction procedure.

We have so far displayed the n-successor and d-successor functions adding nodes to the tree node by node, database by database. We now turn to the addition of structure internally to the labels. We exemplify with a metabox display of (4) (the nodes created by Modus Ponens omitted for simplicity):

(4) Mary worries that John fancies Jo.

M	$s_1, \Theta s_1 \models s_{\text{utterance}}$	
d ¹ M	Mary : e	use me last
d ² M	worry: $e \rightarrow (e \rightarrow t)$	
d ³ M	$s_j : e$	use me first
ns _j $s_j, \Theta s_j \models s_{\text{utterance}}$		
d ¹ ns _j	John : e	use me last
d ² ns _j	fancy : $e \rightarrow (e \rightarrow t)$	
d ³ ns _j	Jo : e	use me first

The critical part in this derivation is in the processing of worries. Worry is a verb which takes only clausal complements, and these must be finite. There is no form of embedding with an expletive pronoun:

(5) *Mary worried it that John fancies Jo

And there is no the fact that construction:

(6) *Mary worried the fact that John fancies Jo.

Furthermore, worry cannot take wh-questions as its complement. Accordingly the procedural specification of worry has to narrowly control the construction of the clausal structure. It induces the building of a node to house its complement, this node itself filled as ' $s_j:e$ ' to be the first argument to combine with the premise 'worry: $e \rightarrow (e \rightarrow t)$ '. This node must be labelled by a database. Furthermore, that database must be independently capable of proving a conclusion of the form $\alpha:t$, α containing not merely a record of how the labels in that database have combined, but also a variable of the temporal system, fixing a unit of time labelling that declarative unit $\alpha:t$. On the other hand, a database open with respect to some variable u is precluded: worry cannot take embedded wh-question as complement. Worries also, as other verbs, imposes the restriction that the formula preceding it in the data base of type e be annotated as the last minor premise in the chain of premises leading to $\alpha:t$, that premise to be labelled with some expression denoting an

individual. And, given its tense specification, it constrains the construal of the database M itself as some s_i, s_j containing the time of utterance. Its full lexical specification accordingly is:

worries worry $e \rightarrow (e \rightarrow t)$ $RLabel(d^{i+1}x) = \{ \text{use me first},$
 $CLabel(d^{i+1}x) = \{s_i, \Theta s_i = s_{utterance}\}, n(s_i)\}$
 $RLabel(d^{i+1}x) = \{ \text{use me last},$
 $CLabel(d^{i+1}x) = \{y, y \in \{y_1, \dots\}\},$
 $CLabel(ns) \neq (\alpha:t, \alpha=f(u))\}$

Examples such as these display graphically how the verb is not merely the initiating major premise, but controls every essential step of the deductive process establishing $\alpha:t$ for the database in which it is entered.

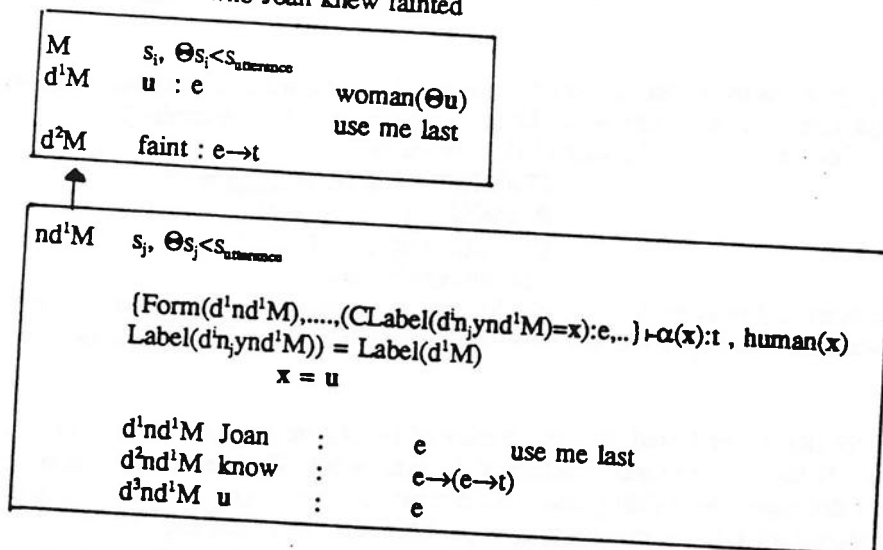
Despite the simplicity of our linked database display of (1)-(4), lacking any quantification, they nonetheless already lay out the overall structure of the explanation. The exercise is a combination of goal-directed deduction and building structured database configurations. Deduction is over the types, with structured information incrementing in the labels for every local step of Modus Ponens taken. All such steps are subject to the local (conditional) resource logic. Furthermore, extra assumptions are made and resolved in the process of achieving the construction. However links between such databases are determined by a global discipline, for linked databases. We thus have a combination of two disciplines, the local discipline, and the global one in which all local reasoning tasks are embedded. In all cases it is the lexical specifications which dictate not merely what goal is reached but the dynamic process of how that goal is reached.

3.2. Tense and quantification

We now extend the analysis in two ways. First we incorporate temporal specifications explicitly. Tense we assume to project information construing the containing metabox labels within a temporal or modal system along lines set out in Gabbay 1989, Gabbay and Hodkinson 1990, Finger and Gabbay 1991. Such labels may, like other labels, contain variables and these have to be identified on-line in an exactly analogous manner to anaphora, subject to whatever side conditions the individual tense or aspectual specification imposes. We thus assume a two-sorted system, a full set of variables ranging over individuals in the domain, and a discrete set of variables ranging over units of time (the units to be chosen). These two sets of variables may interact, individual variables able to be identified as dependent on some containing metabox label. The details of this to come later.

Quantifying determiners, we assume with others, are semantically binary operators. They induce a pair of declarative units, sharing a variable, one presenting a restriction on the range of the variable, the other presenting the main body of the assertion. We seek to express this within the natural deduction format that we are adopting, while retaining sensitivity to the structure presented by the string of words. All determiners are mapped from the lexicon onto variables as an initiating step of elimination, more strictly onto a premise of type e labelled by a variable; and each variable is subject to idiosyncratic side conditions which determine its particular role in the inference process. The restriction on the variable (given by the associated nominal content) is also stated as a side condition. The choice of variable is made on line, even in the relatively trivial case of free (universal) variables, where the only element of choice lies in guaranteeing that the variable selected does not collapse with some other variable illicitly. Determiners are accordingly listed in the lexicon as a premise of the form ' $x: e$ ', x a metavariable subject to an array of side conditions on the identification and use of its instantiation, Θx (specified as a Resource label on that premise). Universal determiners project free variables; existential determiners project existential variables.⁴

We now combine such determiner specifications with the linking account of relative clauses (again omitting the steps of Modus Ponens):
 (7) A woman who Joan knew fainted



The structure is built exactly as before.³ The only new features are the provision of an existential variable at node d¹M labelling the formula e, with a side condition restricting its range using the following noun (this being a definitional property of nouns). Technically, u must also satisfy:

u : e	woman : $e \rightarrow t$	$\vdash \alpha : t$
-------	---------------------------	---------------------

3.3 A semantics for linked databases

From this resulting structure, we can straightforwardly define a classical theory giving its content in familiar predicate logic terms. We illustrate here how it might be done - in two steps.

- (i) We define a mapping for the local databases in our parsing logic (LDS) onto a classical logic.
- (ii) We define a classical theory for a linked database by induction over the subparts of the linked database. We need two proof-theoretic concepts.

Let $\Delta = \{t_1 : A_1, \dots, t_n : A_n\}$

We say $\Delta \vdash_0 \alpha : B$ can be obtained by modus ponens from some elements of Δ . We say $\Delta \vdash \alpha : B$ if $\Delta \vdash_0 \alpha : B$ and all elements of Δ are used exactly once.

- (i) Let Δ be a labelled database. Let t be an atomic formula, and let e be another atomic formula. Consider:

$\Delta \vdash_0 x : e$
 $\Delta \vdash_0 y : t$

For some x_i, y_i this holds. We construct a classical theory corresponding to Δ as follows:

Terms: $\{x \mid \Delta \vdash_0 x : e\}$
 Unary predicate symbols $\{x \mid \Delta \vdash_0 x : e \rightarrow t\}$
 Binary predicate symbols $\{x \mid \Delta \vdash_0 x : e \rightarrow (e \rightarrow t)\}$

Let $\psi_\Delta = (y \mid \Delta \vdash_0 y : t)$ where y is read as a wff

The semantic principle underlying such lexically driven translations into L_1 is now straightforwardly stateable. In predicate logic we read John and Mary as terms and love as a predicate. Let a_{John} , a_{Mary} , R_{love} be the expressions denoting the elements and predicate involved. Then John loves Mary becomes:

$$R_{love}(a_{Mary})(a_{John})$$

The LDS database Δ_0 can produce a corresponding classical logic database Δ_1 comprising all labels α such that

$$\Delta_0 \vdash_0 \alpha : t$$

α is read as a predicate logic sentence according to the recipe:

labels with lexical e as objects (terms)

labels with lexical $e \rightarrow (e \rightarrow t)$ are binary predicates

labels with lexical $e \rightarrow t$ are unary predicates

and $\Delta_1 \vdash R(b)(a)$ iff $\Delta_0 \vdash_0 R(b)(a) : t$

(ii) We now address the problem of defining a classical theory for linked boxes to determine the predicate logic formula that is the result. We want (7) to be assigned a truth-value corresponding to:

$$(7') \exists x (woman(x) \wedge knew(x)(joan) \wedge fainted(x))$$

We must make sure that we get (7) out of the links. On the basis of a linked database, we define a classical theory ψ^* associated with it. Assume the links form a tree. This means that we can define ψ^* by induction:

(1) $\psi_{\Delta^*} = \psi_{\Delta}$ for one database

(2) $\psi_{\Delta^*} = \gamma(\psi_{\Delta_i^*}, \psi_{\Delta_j^*})$ for data base Δ_i, Δ_j where γ is some (Boolean) operation on $\psi_{\Delta_i^*}$ and $\psi_{\Delta_j^*}$. For example, for wh links, γ gives conjunctions.

Such a mapping will yield as output predicate logic formulae, for which we can define a semantics in familiar ways.

4. Consequences: The crossover phenomena

With an outline of the framework to hand, we are now in a position to see how phenomena familiarly thought to require syntax-internal stipulation turn out to be consequences of the process whereby linked databases are built up as interpretations. We exemplify with crossover phenomena. Crossover phenomena are the data displaying interaction between wh-gap dependencies and pronominal construal, phenomena notable for not being reducible to semantic considerations of an orthodox sort:

(8)* A woman_i who_i John believes she_i thinks Sue fancies fainted.

She in (8) cannot be construed as dependent on the woman, despite the fact that neither a woman nor who is local to the occurrence of she. If we analyse wh as a variable-binding operator, this is perplexing since we seem to have an operator idiosyncratically not automatically licensing the construal of all pronominals which it c-commands as dependent on it. But this is by no means the end of the puzzle. If the wh binds a gap in argument position and the pronominal is also an argument, any configuration with the wh to its left and the associated gap to its right debar dependence of the pronominal with the wh. The dependence of the pronominal on the wh is only licensed if both wh and gap precede it:

(9) Who_i is Mary_j certain she_{i,j} knows that Sue is anxious about e_i?

(10) Who_i is Mary_j certain e_i knows that Sue is anxious about her_{i,j}?

(11) Joan_i, who Mary_j is certain she_{i,j} knows that Sue is anxious about e_i, is drunk again.

(12) Joan_i, who Mary_j is certain e_i knows Sue is anxious about her_{i,j}, is drunk again.

But if either the pronominal or the wh is in a determiner position, the restriction is sensitive to the larger configuration in which it is embedded. In particular, though the crossover effect is real enough in wh-questions, and in restrictive relative clauses too, it vanishes if the wh construction is serving as some (nonrestrictive) relative clause to an independently identified head:⁶

(13)*Who_i is her_i mother anxious about all the time?

(14)*Every woman_i who_i her_i mother gets anxious about gets sick.

(15) Joan_i, who her_i mother is anxious about all the time, is sick again.

(16)*Whose_i dying mother is she_i anxious about all the time?

(17)*A woman_i whose_i dying mother she_i is anxious about has had to go up to the hospital.

(18) Joan_i, whose_i dying mother she_i is anxious about all the time, has had to go up to the hospital. The heart of the problem is that an apparently configurational restriction between wh, pronominal and gap is sometimes sensitive to context, and sometimes not. Why should this be so? The data remain recalcitrant defined solely over the local configurations in which they are contained. They appear to need at least two, if not three stipulations in addition to the characterisation of wh and pronominals themselves (for weak, strong and extended crossover respectively)⁷ -possibly even an additional type of empty category (Lasnik & Stowell 1992), or, within Categorical Grammar, syntactic invocation of an additional operator which is not lexically reducible (Jacobson this volume). Yet they can be explained direct from the lexical definition of wh and pronominal expressions, if we shift into looking at the problem from the procedural perspective of the steps that have to be taken, the choices made, in getting from the given input specification to some licensed goal. We take wh questions first where there is no lifting of this "crossover" effect.

Wh, recall, is not a premise in its own right - merely a specification of a goal to fulfil, a specification entered in the database on parsing the wh expression as an additional restriction on the goal to be achieved, that it be open with respect to some variable. Even the choice of variable remains to be made online, so all we know in virtue of the wh itself is the requirement of building an open formula, this specification entered as a metavariable specification in the goal box. Any subsequent pronoun entered into a database with this weakly specified goal has no information online in the database itself with respect to this goal specification, and can only be identified independently. We predict correctly:

(19) *Who_i does his_i mother like e_i?

(20)*Whose_i mother does he_i like e_i?

(21)*Who_i does he_i like e_i?

The depth of embedding between the wh and the gap makes no difference. The target specification remains but a shadow, a goal to be realised and not part of the database itself. In the case of an intervening pronoun, then, the metavariable of the target specification does not constitute a possible source from which to pick up a value. (22) is no more acceptable, construed as indicated than (19)-(21):

(22) Who_i does Bill think she_i worries e_i is sick?

The process works as follows. The wh sets up a goal specification, $\alpha(x):t$, requiring a declarative unit of the form $x:e$ to be used at some point in the projection of a chain of modus ponens to this conclusion. In the building of such proof-theoretic structures, the procedural specification of the verb is the major initiating force, annotating the nodes in its local domain to determine how the various premises filling these nodes combine to yield the requisite conclusion. These two properties combine to determine the phenomenon known as wh-gap binding. In order for the label and formula in the goal specification induced by some wh expression to be used in attaining the appropriately labelled conclusion, there must be one node required in the projection of proof structure for this conclusion which lacks both label and formula. If there is no such empty node, neither label nor formula will get used. So if at some juncture in the proof-building process a verb

induces an array of nodes to be appropriately annotated, and one such is not filled by the specification of the words in sequence, then the label and the formula of the goal specification will duly be used, by assumption, to fill it and so fulfil that goal specification. Once having been so entered in a database, together with an annotation as to how it is to be used, that premise, albeit labelled only by a metavariable, is able to function as a properly labelled premise in a proof-theoretic structure; and is accordingly identified as a premise of that database, and available for the purposes of establishing anaphoric dependence. Hence we allow:

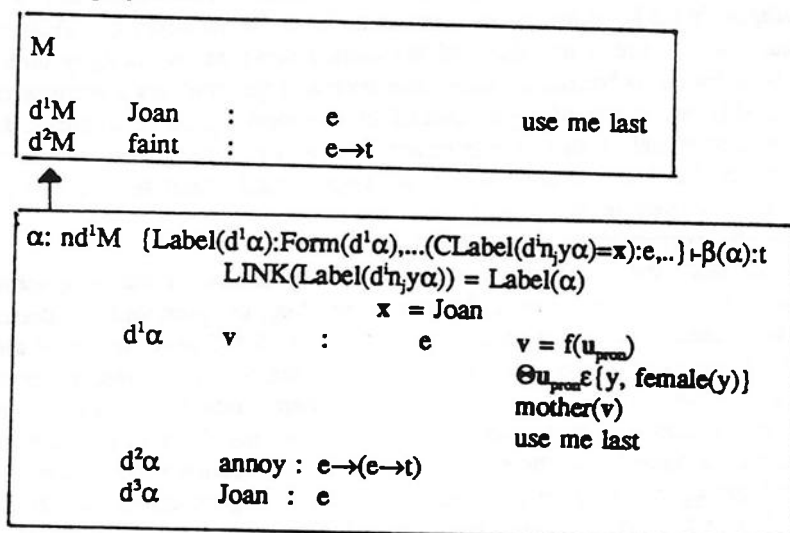
(23) Who_i does Joan think worries that she_i's sick?

(24) Who_i is frightened she_i's going to fail?

Whether the information presented by the wh is useable as a premise has so far turned on whether the requisite premise has been independently constructed in the database. Suppose however the wh is introduced in an environment in which it is to be linked to some database through a variable independently identified as in (25), containing a nonrestrictive relative clause:

(25) Joan_i, who_i her_i mother annoys all the time, fainted.

Here we shall have a different story, for with prior identification, more information is available as input to the process of establishing online dependencies. We know now the form the premise to be added is going to take: it is the premise 'Joan' : e'. The only information we lack is what role it is going to play in the database. This extra information makes all the difference. The procedure is displayed in:



The building of nd¹M is driven by who. But, in order to satisfy the linkage requirement on all nodes of the form nx, x a node, the new database open with respect to some variable is immediately replaced by the specification that the database α must contain some predication on Joan. This information about the required use of the premise Joan:e is available when the hearer processes the metavariable induced by the presence of her. The pronoun can therefore be identified as 'Joan' as long as the configuration licenses her as dependent on it. The locality restriction on the metavariable associated with her is indeed satisfied because the metavariable, u, is not itself a minor premise in the local database: it is merely some function on u that is. So, just as dependency between Joan and her is licensed in Joan loves her mother, so here, dependency between her and the specification Joan given in the initiating goal specification is not precluded.

Suppose however we turn to the core strong crossover cases with both wh and pronoun in argument position. Here we find no asymmetry between wh-questions and relative clauses of any sort. We take first the simple case:

(26) *Joan_i, who_i she_i thought Bill liked, fainted.

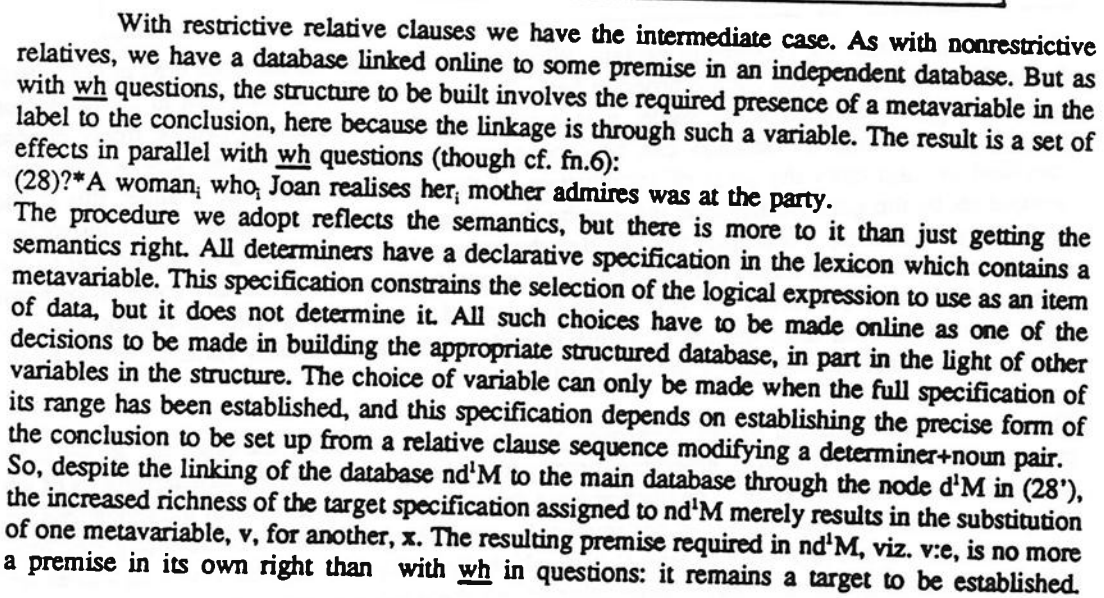
As before, we have an initiating constant to which the database built up through the relative clause is to be linked. Here, as with (25), the target for this second database becomes much more specific. It is that a database be built containing some combination involving the premise 'Joan' : e'. Here however, unlike (25), we come up against the locality condition intrinsic to the metavariable associated with the pronominal. This requires that its metavariable not be instantiated by any label which is a minor premise in its own database. But this time, the pronoun is itself in the database; so too is the premise 'Joan:e'. Accordingly, dependence on the premise induced by the wh as linked to the initiating database s_0 is precluded.

It might seem as though this restriction on identification of the pronominal would only apply as stated in cases such as (26) in which the wh and pronoun are demonstrably local to each other in the surface string. But this is not so. The identified wh-premise is part of the goal specification of the database for the clause that the wh introduces. The discipline of goal specifications requires that any overall goal be carried down to all subgoals until such specification be satisfied. The specification "use some premise of the form $\alpha:e$ " (α in these examples some fully identified name) must be satisfied in one of these subgoals. The building of the database constructed out of a complement clause is a subgoal in the building of some containing label. So the target specification of identifying which argument role $\alpha:e$ is to fulfil will get carried down as an ancillary target specification for the database for the subordinate clause. But such a target contains a premise of the form to which the locality restriction of a pronominal of the form $u_{\text{pron}}:e$ is sensitive. In consequence, if there is a pronominal in such a subordinate clause, its locality restriction will preclude its dependence on the label α contained within the target specification as long as this assigned highly specific target remains unmet. (27) displays the process:

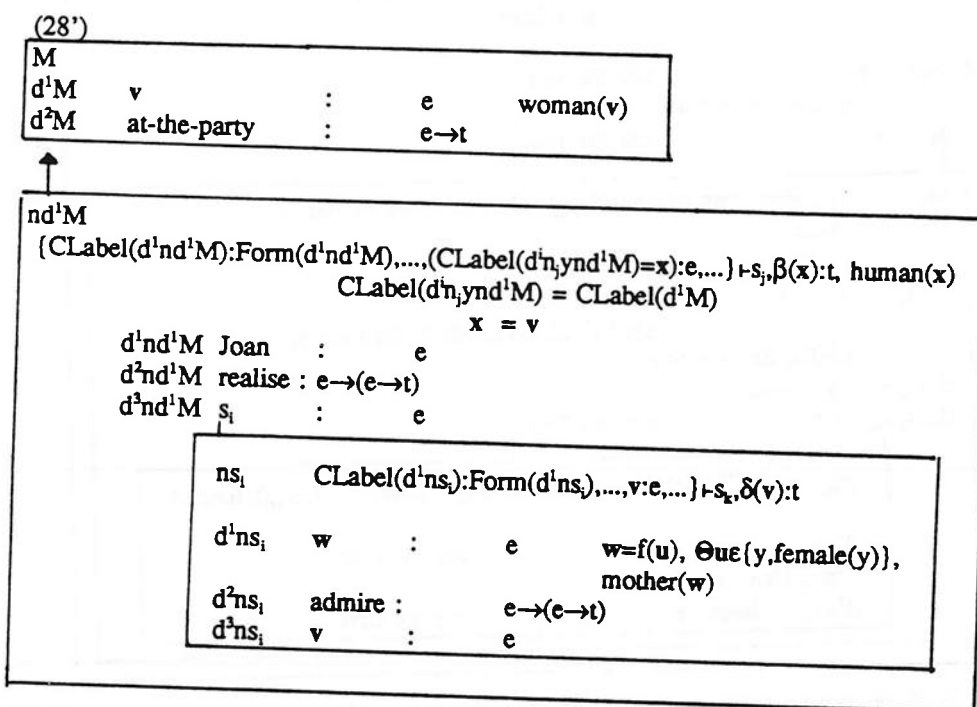
(27) Joan, who Sue knows she worries Mary likes, fainted.

In (27') below, the node ns_k bears the Resource Label that it form a database containing some minor premise 'Joan:e' in deriving its conclusion $s_k, \delta(\text{Joan}):t$. At $d'ns_k$, the pronominal induces the premise 'u:e', Θu to be chosen. This choice may not include 'Joan' for 'Joan:e' is one of the premises, according to the target specification, to be used in that database. Here, instead, u is identified as 'Sue'. Once of course such a premise has been entered into the database, the requirement of the goal specification will be met and will need no carrying down to any further subgoals. Hence once the gap is passed, and the necessary premise duly constructed, anaphoric dependency on the gap and the wh will duly be licensed. We get the strong crossover facts from the proof discipline direct. Notice too that we absorb the extended strong crossover cases directly, for it is not the status of the gap that is critical but its position in the database. It is the locality condition on the pronoun that does all the work, not some condition on the gap.

M	s, $\Theta s_i < s_{ux}$
d ¹ M	Joan : e
d ² M	faint: e → t



However, again just as with wh questions, this metavariable labelling a premise of type e , must be used as a premise in the projection of proof structure, and accordingly there must be one node induced by a verb which can only be identified by assuming it bears that label and that formula. Once so used at a node whose role is identified by an associated major premise, the premise, with its metavariable as label, has a fully identified role in the proof structure and accordingly is available as a label for a subsequent pronominal in that proof structure to use as antecedent, exactly as in wh questions.



We have one remaining problem to explain. Why cannot the label v to the node d¹M in (28') itself be used to provide a value for the pronoun her? The reason is that, at the point in the derivation at which her is assigned some value, v is a metavariable. Pronominals cannot freely choose variables as antecedent: variables can be chosen as value for a pronominal only within some specified domain from the point of introduction of that variable. For metavariables, this is the domain set by the goal specification induced by the wh expression (i.e. internal to the construction of the relative clause). So despite the fact that the metavariable v at node d¹M is identical to the goal specification for its linked database nd¹M, it cannot itself be used to identify some pronominal inside nd¹M. And the value for that pronominal must be chosen online, so cannot have its identification delayed until the metavariable is replaced by some chosen variable which might license pronominal dependency from that position.⁸

Though the data presented by the crossover phenomena are complex, the analysis, as we have presented it, is relatively simple, emerging directly from the proof-theoretic characterisations of wh and pronominal anaphora alone. The phenomenon demonstrates how the array of interpretations we assign to natural-language strings is not invariably explicable in terms of the resulting structure. Anaphora and wh are two types of lexical item which direct the way in which such structures are built up, and their interaction may depend critically on the way information on-line is set up, here by linking. For both, what choices are available depend on what choices have already been made.

The procedural account is not just an adjunct to what is at heart a configurational restriction: to the contrary it is the heart of the story. The information that is built up in utterance interpretation depends not just on the set of premises, not even just on the input sequence of words, but on those words and how they are used to make choices en route to achieving some assigned goal.

5. The Language Faculty: The new procedural look

What novel assumptions have we committed ourselves to? We have a new concept of information content, and a new concept of syntax, hence of a grammar. We address the semantic issue first. What we are proposing is that natural-language expressions project information of two sorts - information about the world around us, and information about how to build configurations from which such information is derivable. The structural concept of information is meta to the more classical concept, and it is the structural concept in terms of which lexical specifications of content are projected. This twofold approach to information arises from the nature of utterance interpretation and the role that natural language plays in that process. We use natural language to convey information about the world around us. But this is only the end goal of language use. In order to convey this information, we have also to convey information about how such information is to be retrieved from the signal we provide. The former may be reconstructible in model-theoretic or situation-theoretic terms, depending on our choice. But the latter is essentially proof-theoretic.

The heart of this shift of perspective into the meta enterprise of how to build requisite inferential structures lies in the richer conception of logic itself, as embodied in the LDS framework. We can now, in closing, see how the shift from a static configurational perspective into a dynamic procedural one explicit in the natural deduction format of the LDS formalism demands a comparable shift in our conception of a grammar for natural languages. The point of departure for our research was to assume some encapsulated concept of grammar, albeit reduced and underdetermining descriptive content, feeding the reasoning processes which serve to enrich this input specification into some full specification of (utterance-specific) content. As we now present the model however, the encapsulated body of information is embodied in the lexicon. It is the lexicon, with associated statements as to order of expressions to which the construction function may be sensitive, which presents this under-determining input. What then of the grammar? There is now only one possible answer. The grammar is the formal model of the knowledge that a speaker/hearer has which enables him/her to use the language as he/she does. The realisation of any such model has to license all and only the wellformed strings of a language; it has to assign each such string with a wellformed representation in some logical system, a representation said to constitute the interpretation of that string; and it has to define a formal mapping between a string and its interpretation which determines at each stage what are the possible steps in this mapping process. It does not itself make choices as to which out of a set of interpretations should be chosen; it merely sets out the full array of interpretations available. This our model does. Indeed it defines wellformedness of a string as the successful outcome to a conclusion α :t using all information lexically presented by the given sequence of words. It was set out as a formal parser. But it is in substance a grammar. It does not make actual choices; it makes no wrong choices en route as would a human parser; it follows no heuristics for attachment choices as would a human parser; and it does not have the flexibility to assign interpretations to strings it doesn't license, as humans, charitably, do. It merely lays down a general characterisation of all possible moves.⁹

Notice the outcome. We have abandoned all trace of a dichotomy between a separate, and encapsulated, mental faculty specific to language and some generalised reasoning faculty. Our faculty for language and our faculty of reasoning are two sides of one and the same coin. There

is, of course, language-specific information intrinsic to our knowledge of language as expressed in the lexicon; but it is information which, as the input to language processing, plays a critical part in controlling the reasoning process, and so must be stated in terms of that reasoning process - not in terms particular to or unique to any encapsulated black box device. Our reasoning process may take input other than that provided by the natural language lexicon, we may reason over linked databases direct, our languages may vary dramatically in terms both of the premises they induce and the control features they impose; but none of these is incompatible with the intuitive observation that at the heart both of language and of our capacity for reasoning is the ability to take elements of information, combine them together to form complex, structured concepts, and then reason with them as wholes. And it is this which our LDS model directly reflects. Our separate languages may constitute different logics according as the vocabulary of the labels and the control specifications on those labels varies. But they all fall within the same family of logics defined by the LDS framework.¹⁰

Notes

1. The d-successor function applied to a label gives rise to more than one label to a node, as in John, the idiot, forgot his keys. As with the n-successor function applied to a label, such successive labels must be linked. For some label t_i , $\text{Label}(d^i t_i) = t^i$
2. We assign predicates such as seem the specification that the agreement feature they bear is not restricted to applying to such a final premise. Their co-occurrence with infinitives which are incomplete databases, lacking only their final argument and a metabox label, then guarantees that the surface subject of seem combines with this predicate.
3. As a question, wh will lack the subordinating function, and has no instantiating value available with which to be linked.
4. We make this assumption here for simplicity. In a later paper we argue that indefinites project dependent variables, a dependency which can be established as a function from both expressions in the temporal system and expressions in the regular individual system.
5. We assume that the proof theory for linked databases will dictate that given a pair of databases each of the form $\text{label}:t$ the result of linking the two together will also have the form $\text{label}:t$, reflecting the general form of inference rules:

$$\frac{\Delta_j: t \quad \Delta_k: t}{\Delta_m: t}$$
6. The effects are clearest with examples in which the determiner disallows nonrestrictive interpretations altogether, as does every. In examples which allow both nonrestrictive and restrictive interpretations, judgments are weakened.
7. We are grateful to Polly Jacobson for reminding us of the Postal lecture to the 1991 LSA Summer Institute in which he drew attention to the "extended strong crossover phenomena", i.e. the examples which illfit the mainstream analysis.
8. This part of the analysis turns on a locality theory of variables. All details later. That variables have an associated locality restriction turns out to be further evidence that indefinites do not

introduce variables of a discrete existential sort, but functionally dependent elements, whose accessibility as antecedents turns on the nature of the element on which they depend. The label introduced by an indefinite article can act as antecedent for a subsequent pronominal which is not in its associated relative clause, as long as either the element on which the indefinite is dependent is itself a constant, or if that element is a variable then the pronominal is within the domain licensed by the variable in question.

9. Despite its articulation as a procedural device, the relationship between string and assigned interpretation can straightforwardly be viewed in reverse. The relation is that between proving forward to some conclusion, and planning backwards in the face of some fixed conclusion how best to reach it. Perception is the forwards, weakly goal directed task of proving to some conclusion $\alpha:t$. Production is the strongly directed activity of having some particular conclusion $\alpha:t$ in mind and planning how to provide a string of symbols which will lead inexorably to that conclusion.

10. The research for this paper was supported in part by ESRC grant R000232069. We are grateful to our collaborators, Jeremy Pitt, Marcello d'Agostino, Jim Cunningham for day by day feedback; and to Annabel Cormack, Ruy de Quieroz, Rita Manzini, Deirdre Wilson, Franz Guenther, Asli Goksel, Julia Capritsa, Wynn Chao, Hans Jürgen Ohlbach, Jan Yiang, Wilfried Meyer Viol, for many comments.

References:

- Finger, M. & Gabbay D. (1991) 'Adding a temporal dimension to a logical system'. ms.
 Gabbay, D. (1989) Labelled Deductive Systems. to be published OUP.
 Gabbay, D. and Hodkinson, I. (1990) 'Temporal logic in context'
 Lasnik, H. and Stowell, T. (1991) 'Weakest crossover' LJ 22, 687-720.
 Marcus, M. (1980) A Theory of Syntactic Recognition for Natural Language. MIT.
 Pereira, F. (1985) 'A structure-sharing representation for unification-based grammar formalism'. ACL, 137-44.
 Pereira, F. & Warren, D. (1983) 'Parsing as deduction'. Proceedings of 21st Annual Meeting of ACL.
 Sperber, D. and Wilson, D. (1986) Relevance: Communication & Cognition. Blackwell.

A Quasi-Naive Semantics for Interrogatives and its Implications

Jonathan Ginzburg¹

Stanford University

1 Introduction

There is a simple and somewhat naive conception of what the content of (a query use of) a *wh*-interrogative sentence is (see e.g. Jespersen 1965.), namely that it is a proposition containing one or more blank slots or free variables:

- (1) a. Who does Jill like.
b. LIKE(jill, □)
- (2) a. Who likes whom.
b. LIKE(□, □)

Blank-slots or free variables always take widest scope, and propositions with slots/free variables in them do not have descriptive power. So this conception would seem to be defeated in the face of phenomena involving

- Embedded interrogatives:
 - (3) Max knows who Jill likes. (The embedded sentence denotes a propositional entity that has descriptive power, in this case an object of knowledge.)
- Scopal interactions with quantifiers:
 - (4) Which person choose each book. ('Each book' seems to be able to scope over 'which person')

This has lead to a fairly general consensus that the real story on both interrogative sentences and interrogative phrases must be quite different. The dominant line of research into the meanings of interrogative sentences is one that can be dubbed *supra-propositional*, since it views interrogative sentences as denoting on any use a higher-order object in propositions. The basic intuition driving this approach is that interrogatives specify either *instantiated answerhood conditions* (apparently Hamblin's original intuition as expressed in Hamblin 1973), where an *instantiated answer* is an instantiation of the open sentence underlying the interrogative, or *resolving answer conditions* (Karttunen 1977, Groenendijk and Stokhof 1984), where a *resolving answer* is essentially the maximal among true instantiations. For the most part, it has been assumed that *wh*-phrases are quantificational expressions that interact scopally with other quantificational expressions such as nominal quantifiers or for that matter other *wh*-phrases.²

In Ginzburg 1992, I argue extensively against both the supra-propositional approach to interrogative sentence meaning and the quantificational approach to interrogative phrase meaning and offer an alternative which starts out with the naive conception as its intuitive basis. By refining the naive, using various insights that have developed over the years, it is possible to confront the various data that have lead to its implicit rejection head on, and provide solutions that provide new insights into these and other issues. In the current work, for obvious reasons of space, I limit myself to discussion of a small subset of the issues and provide a sketch of my proposal.

I start by arguing for the need to reconsider the standard characterisation of the response space to a query use of a unary wh-interrogative sentence. I then discuss what this can tell us about the nature of questions, by which term I refer to the descriptive contents of queries. I subsequently move to consider some factors pertaining to the quantificational status of wh-phrases: evidence from multiple wh-sentences is provided that wh-phrases manifest scopal behaviour that is intrinsically different from that of nominal quantifiers. This leads on to a sketch of the proposed non-quantificational account of wh-phrase meaning.

2 Answerhood

What does the use of a wh-interrogative sentence to query involve?³ A common conceptualisation assumes that a query provides either for a choice between various instantiators, or, alternatively, for some kind of specification of a true and exhaustive answer. I would like to offer an alternative conception, namely that a query specifies an informational void by means of its descriptive content, which is a propositional entity I will refer to as an *unresolved* proposition. Roughly speaking, this is a singular proposition one or more of whose argument roles is parametrised.

Of course, any choice situation can be represented as a informational void conjoined with a disjunction over the members of the choice sequence. Conversely, one can conceive of voids as providing for choices in some sense, since any parameterised role carries information concerning the potential fillers of that role, and the query provides implicit indication of the domain of candidate fillers (the *queried domain*). The crucial difference, however, is that an agent can specify an informational void by providing an unresolved proposition together with a queried domain, *without* having any information whatever about candidate fillers, indeed about the existence thereof, within the queried domain. Such a state of ignorance is common for queriers.

I consider two types of data from query/response discourses that militate in favour of the current conception and against either the choice or the true and exhaustive answer conceptions. The first comes from considering which answers can be conveyed by felicitous responses to a wh-query. The second concerns the possibilities for reporting responses to queries using embedded interrogatives.

2.1 The answer space of questions

I first examine the status of non-exhaustive instantiated and quantified answers, claiming it to be on a par, semantically, with that of true and exhaustive instantiated answers. In other words, responses that convey answers that are intrinsically quantified or instantiated but explicitly inexhaustive or even false are an acceptable manifestation of rule-governed linguistic behaviour. In fact, even negative answers, answers that indicate that no true instantiation exists can be felicitously conveyed. Taken together, these facts belie the conception of a choice between instantiations or the specification of a true and maximal answer.

Making this point requires a little care: I do not wish to imply that in all contexts the *desires* of the querier would be equally met by conveying, say, an intrinsically quantified answer rather than an instantiated answer. It is easy to refute

this claim. Frequently, it is indeed the case that a (wh)-query is posed because of a desire to find out the actual identity of the set of instantiators of the void described by its descriptive content. Responders identify this desire, and adhere to the norms of maintaining communicative cooperativeness. Hence, answers are implicated, perhaps even as generalized or default implicatures, to be true and exhaustive. This means that *ceteris paribus* (cooperative) speakers are under an obligation to provide implicit or explicit indications when exhaustiveness is not intended to be conveyed:

- (5) a. Who has worked in the past on this problem?
- b. I know Hughes and McClair have.

In this example, the responder cancels any pretense of providing an exhaustive answer by relativising the force of the assertion to his own knowledge.

Our claim is, therefore, that the considerations that often bias a question to prefer exhaustive answers can and *should* be explained on a pragmatic level, by reference to basic principles of communication. For one, false and inexhaustive answers are *semantically* acceptable, even when the latter are often *unuseful*: someone violating these requirements will usually be thought of as a jerk, rather than an incompetent speaker. But more crucially, there are many contexts where inexhaustive, in particular intrinsically quantified answers are felicitous, as we shall see.

Consider the question in (6) uttered during an election night special:

- (6) Well, I'm sure we're all keen to understand the following: who supported the candidate?

Plausibly, the interviewer is not interested in either a full or explicit characterisation of the set of supporters of the candidate. What he is interested in are general trends, describing both the make up of the set of supporters and of its complement, and references to select members of either of these sets. Responses such as the following, the first of which is non-exhaustive instantiated, the other two of which are quantified, would be typical:

- (7) a. Senator Johnson and Representative Smith definitely did.
- b. Surprisingly enough most naturalised Javanese did.
- c. Well, few people from the West did.

Consider now the question in (8) made as a request for a plan of action for two people standing at a crossroads, waiting for a traffic light to change:

- (8) Which way shall we go now?

The addressee can pick one specific way, and point at it, as in (9a), or she can choose a no less optimal strategy, implicit in the quantificational answer indicated in (9b). Both responses are equally felicitous, it seems:

- (9) a. *That* way.
- b. Any way that's not blocked at present.

Responses conveying negative answers can be felicitous. Once again, it is important to emphasise pragmatic/semantic distinctions: using a wh-sentence to query often carries with it an implicature that the set of instantiators is not empty. But this implicature can be cancelled explicitly, does not always arise and is calculable:

- (10) What, if anything, should I buy at the store?

(10) can be uttered felicitously and does not obviously suggest that the querier believes the set of 'things he should buy at the store' is non-empty.

(11) Who is in favour of amending the Bill of Rights?

The speaker of the House can felicitously utter the sentence in (11) before a vote without a necessity that he *believe* anyone supports amending the Bill.

The existential implicature is calculable: roughly, a hearer can reason: 'Why use (11), for instance, when the neutral 'Is anyone in favour of amending the Bill of Rights?' could have been used? The speaker must have had grounds for using a form in which the *role* of 'person in favour of the amendment' is focussed. Hence, he must suspect that, indeed, that role is instantiated, etc.'

The basic point that emerges so far is that queries can, in principle be used to specify both lower and higher order information, with pragmatic factors deciding which option is favoured. In particular, true and exhaustive instantiated answers should not be assigned any particular priority as answers that concern the question. Rather, we have argued that the exhaustive and veridical force of answers is a defeasible, pragmatic effect, which does not persist in all contexts.

2.2 Reported responses and resolving answers

Evidence for the assumption that answers are conveyed to carry a true and exhaustive force has been precisely the veridicality and exhaustiveness manifested by interrogative complements embedded by such predicates as *tell* and *guess* (see e.g. Karttunen 1977, Groenendijk and Stokhof 1984, Engdahl 1986). This same meaning, possibly intensionalised, was assumed to be embedded by the illocutionary query operator, or by such predicates as the query reportive predicate *ask*, the corresponding mental puzzlement predicate *wonder*, or the preposition *about*.

As I show in Ginzburg 1992, however, there is strong evidence that supports *distinguishing* the class of interrogative meanings embedded by propositional complement predicates such as *tell* (resolving answer interrogatives [RI]) from the class of interrogative meanings embedded by predicates such as the illocutionary query operator or *ask* (question interrogatives [QI]), and hence against thinking that veridicality and exhaustiveness are conveyed as part of the *semantic* force of the query. Various arguments are provided against *reducing* one type of meaning to the other, *pace* Groenendijk and Stokhof's proposed intensional/extensional interrogative complement distinction.

Furthermore, it is demonstrated that the veridicality inherent in RI meaning cannot be captured by assuming the interrogative meaning specifies or provides for a true proposition (Karttunen 1977, Groenendijk and Stokhof 1984) or derives from considerations of factivity of the embedding predicates (Berman 1990). A detailed proposal for capturing the veridicality is provided, based on a distinction between two notions of propositional entity and corresponding notions of truth. This enables one to account for systematic and cross-linguistically stable patterns of applicability to RI meaning (i.e. the ability to take an RI meaning as an argument.) that I show propositional complement predicates to exhibit.

Establishing these general points, then, lies at the heart of my claim of adequacy for the *naive* intuition about the contents of queries. We restrict ourselves

here to raising one central point, the need to distinguish the notion of a *resolving answer* of a given question from information that concerns or is pertinent to that question.

Felicitous responses, even indirect or extra-linguistic, can be reported using embedded interrogative complements:

- (12) a. (It is known that only George Bush and Dan Quayle could possibly have committed the crime.) Q: Who committed the crime?
Jill: Put it this way, Dan Quayle was out of town.
b. Jill informed me, albeit somewhat indirectly, who committed the crime.
- (13) a. Q: Who is your best friend?
Jill: (points at Mike)
b. Jill indicated to me who her best friend was.

Matters are more complex, however. In fact, the acceptability of the reportings above is dependent on the veridicality and exhaustiveness of the answers conveyed. We would not assent to (13b) as a report of the dialogue if we were aware that Mike were not Jill's best friend. Rather, we would be forced to a report such as

- (14) Jill purported to indicate who her best friend was.

The conclusion this suggests is not that responses that convey false answers are infelicitous from a semantic point of view, though of course they violate basic principles of cooperative conversation. Rather it suggests that the RI interrogative meaning cannot be used to paraphrase just any felicitous response. Similar remarks *mutatis mutandis* apply to reports of responses that do not convey exhaustive answers. It is not the case that false or inexhaustive responses cannot be reported as *concerning* or *being about* the query they are directed at. Consider, for instance, (15b): it can be used to report the response provided to the query regardless of the veridicality or exhaustiveness of the answer conveyed. This is clearly not the case with (15c).

- (15) a. Q: Who do you think offered me a job today?
Jill: Dunno, Dan Quayle perhaps.
b. Jill made a guess about/tried to guess who offered me a job.
c. Jill guessed who offered me a job.

The point the examples above bring out, then, is the need to distinguish the notion of an *answer*, any information that is conveyed directly or via pragmatic inference by felicitous responses, from a *resolving answer*, information that is factual and fully determines a resolution of the issue described by the question. The contents of all felicitous responses to a given query can be characterised using the former notion, whereas the latter is a property that is quite tangential to the *felicity* of a response. In other words, whether a response can be reported by appropriate disquotations of an interrogative sentence under a predicate such as *tell* or *guess* cannot be used as a diagnostic for the felicity of a response.

2.3 Interim conclusions about answerhood

In the previous two sections, then, we have seen evidence from query/response pairs and from the reporting of felicitous responses to queries suggesting the need for a richer notion of answerhood than that recognized previously. The basic idea is that

answerhood is a relation that can be used to characterise the relation *Response – concerning*(*r*,*q*), that holds between a question *q* that is the descriptive content of a query (or the content of an interrogative embedded by a predicate such as *ask* or *about*) and the content conveyed directly or via pragmatic inference by an information-providing act *r*, linguistic or otherwise, whose content is semantically appropriate relative to *q* ('provides an answer to *q*').

Restricting ourselves for the moment to unary wh-questions, this is captured simply by postulating that questions are *unresolved* propositions, schematically, a propositional entity one of whose argument roles is parametrised, *q*(parametrised-role → *x*):

- (16) a. Response-Concerning (*p*,*q*) iff
 Instantiated-Answer(*p*,*q*) ∨ Quantificational-Answer(*p*,*q*)
 b. Instantiated-answer: *q*(*x*)(*x* → referent). (Referent supplied by referential act using a referential NP, deixis etc)
 c. Quantificational-answer: Quant-force *x*($\lambda x q(x)$)

3 Questions as unresolved states-of-affairs

I provide here an overview of a generalised theory of propositions which *inter alia* provides for a notion of an unresolved propositional entity whose need was motivated above.

The theory is framed within a class of 'Russellian' proposition or state-of-affairs (SOA) theories. This approach originates with Russell and has recently gained renewed interest particularly for purposes of cognitive representation starting with the work of Kaplan, as well as work in situation theory originating with Barwise and Perry 1983.⁴ The current conception and exposition derives from the situation theoretic perspective, and draws in particular on an idea from Crimmins' recent work on a parameter-less theory of SOA's (Crimmins 1991).

SOA's are structured objects individuated by a main relation and a (possibly partial) assignment of entities to the argument roles of the main relation. It should be emphasised that SOA's are taken to be non-linguistic (abstractions), individuated in terms of real-world (or fictional, if the need arises.) objects. They are not sentences in a formal language, though some of them can be profitably be thought of as contents of uses of sentences. Bold-face type is used when the non-linguistic nature of entities is emphasised.

- (17) a. < **LIKE**, liker:jill, likee:bill >
 b. < **HOT**, location: cordura-hall, time: 3:45 pdt >
 c. < **ANNOYED-AT**, annoyed-person:zhang,
 object-of-annoyance:- >
 d. < **GIVE**, giver: ming, recipient:-,object-exchanged:- >

The first two SOA's in (17) are *resolved* SOA's, ones in which every argument role has been assigned an entity. A resolved SOA can be used to describe the world as being a certain way: take (17a). It is an informational unit pertaining to jill's liking bill, for two persons, jill and bill. This unit can be used to represent the

claim that the world is such that **jill likes bill**. One which is verified, and will be called true if indeed it is the case that **jill likes bill**. In order for a SOA to be usable as a description of the world or for representing a claim it must be *resolved*: every argument role has to be assigned an entity.

In the latter two examples, the (depicted) SOA's are *unresolved*, one or more argument roles has not been assigned an entity, which case we depict with a '-': '-' has no status as an object. It indicates merely that the assignment function associating entities with argument roles is undefined on that argument role. Perhaps the most obvious property of an unresolved SOA is that the issue of its factuality cannot arise. This is a direct definitional consequence of what the set of argument roles of a relation are. A direct consequence of this latter fact is an explanation why questions cannot be objects of assertion, claiming etc.

(18) # I claim the following: who does Bill like.

Nonetheless, even in the case where an argument role does not get assigned an entity, that argument role is in the domain of the assignment function, and so is a constituent of the SOA. What the unfilled role contributes are 'presuppositions' on what can count as a unit of information that subsumes the unresolved SOA:

(19) (**GIVE**, **giver**: ming, **recipient**:-, **object-exchanged**:-)

Thus, even though the SOA depicted in (19) has no entities assigned to the **recipient** and **object-exchanged** roles, the SOA carries with it the information provided by the selectional restrictions on each role, for instance that a recipient can only be human. In other words, this unresolved SOA carries with it information that could be used to infer that (20), is a SOA with information that subsumes it, or as we shall say, is a *resolution* of it:

(20) (**GIVE**, **giver**: ming, **recipient**: mikki,
object-exchanged: 5 shekels)

With the notion of resolution we can define the relation characterising instantiated answerhood. Intuitively, these are the resolved SOA's obtained from a question by assigning objects to the unfilled roles of the question.⁵

(21) $\text{INST-ANS}(\sigma, q)$ if and only if $\text{RESOLUTION}(\sigma, q)$.

RESOLUTION relates a resolved SOA σ to any unresolved SOA q whose argument role to entity assignment function is extended by σ 's assignment function. This reduces to the identity on its left argument when this latter is *resolved*.

The class of unresolved SOA's can be put in a 1-1 correspondence with the class of relations: each unresolved SOA's corresponds to the relation obtained from it by abstraction over all its unfilled roles. But there are some crucial differences: a SOA resolved or unresolved is a saturated object that cannot be used to predicate. Boolean operations can be naturally defined, as we see below, to apply to all SOA's. These properties do not apply to relations: they are, *par excellence*, instruments of predication, resolution (or some other substitution-like notion) does not apply to them, and they are not conjoinable or disjoinable unless they share the same type (r-ity and argument-type).

Conditionals can be used to form compound interrogative sentences, and provide one piece of evidence for the position that interrogative sentences must be of an appropriate semantic type to form compounds with declarative sentences.⁶ Further-

more, the fact that conditionalised interrogatives can be embedded by QI predicates serves as evidence that the conditional interrogative is a syntactic unit, rather than coordination across discourse or type shifting:

- (22) a. Who will be the new Prime Minister if the little man resigns?
 b. Jill asked me whether Huebner can withstand the strains of the tournament, if he is required to play many adjournments.
 c. Jill investigated who would be hardest hit if a nuclear winter were to hit Nebraska.

The semantic objects we associate with declaratives and interrogatives allow for a simple account of these cases of mixed coordination.⁷

A compound SOA will be unresolved, unless *both* its juncts are resolved, in which case the issue of its factuality can arise. The unfilled roles are, naturally enough, the union of the unfilled roles from each junct. Given the notion of SOA that we have, and the fact that we wish the operations to reduce to 'ordinary' Boolean operations for resolved SOA's, we do not have much choice in the way we define resolutions of compound SOA's. A resolution of a SOA whose main relation is a binary connective will be a SOA whose main relation is that same binary connective and whose juncts are resolutions of each of the juncts of the unresolved SOA:

- (23) RESOLUTION($\langle \text{BINARY-CONNECTIVE}_{\sigma_0, \sigma_1, \tau_0} \rangle$,
 $\langle \text{BINARY-CONNECTIVE}_{\sigma_1, \tau_1} \rangle$) iff RESOLUTION(σ_0, σ_1) and
 RESOLUTION(τ_0, τ_1)

For instance, (24b) is a resolution of the unresolved compound SOA in (24a):

- (24) a. $\langle \wedge, \langle \text{LIKE, liker:bill, likee:-} \rangle, \langle \text{HATE, hater:-, hatee:jill} \rangle \rangle$
 b. $\langle \wedge, \langle \text{LIKE, liker:bill, likee:minnie} \rangle, \langle \text{HATE, hater:daryl, hatee:jill} \rangle \rangle$

Consider the following example of a mixed coordination:

- (25) a. Who will be the new Prime Minister if the little man resigns.
 b. $\langle \rightarrow, \langle \text{RESIGN, resigner: the little man} \rangle, \langle \text{IS-A, predicated-property: PRIME-MINISTER, predicated-role:-} \rangle \rangle$

Hence,

- (26) INST-ANS($\sigma, \langle \rightarrow, \langle \text{RESIGN, resigner: the little man} \rangle, \langle \text{IS-A, predicated-property: PRIME-MINISTER, predicated-role:-} \rangle \rangle$) if and only if $\sigma = \langle \rightarrow, \tau, \phi \rangle$, and INST-ANS($\tau, \langle \text{RESIGN, resigner: the little man} \rangle$) and INST-ANS($\phi, \langle \text{IS-A, predicated-property: PRIME-MINISTER, predicated-role:-} \rangle$)

Notice that since $\langle \text{RESIGN, resigner: the little man} \rangle$ is a resolved SOA, it has only itself as a SOA that stands in the relation INST-ANS. Hence, we obtain the correct responsehood and answerhood predictions: responses to a conditional interrogative are, for the most part, identical to those elicited simply by the non-conditionalised interrogative. Nonetheless, all responses convey answers whose force is conditionalised by the (content of the) antecedent declarative.

4 Problems for a quantificational view of wh-phrase meaning

Having sketched my proposal concerning the semantic nature of wh-interrogative contents, I move now to consider how meanings of such sentences get constructed

compositionally. I start by considering one of the central motivations for the quantificational approach to wh-phrase meaning that derives from an assumption about the meaning of multiple wh-sentences. It is assumed that any given multiple wh-sentence is synonymous with a corresponding mixed nominal quantifier/interrogative sentence on a reading where the quantifier (putatively) scopes over the wh-phrase. I provide some data that refutes this assumption, and that is, moreover, hard to square with a quantificational view of wh-phrase meaning.

4.1 Disjunction resolution contexts

The first case we consider are 'disjunction resolution' contexts, contexts in which disjunctive information is salient. Multiple wh-sentences used in such contexts do not elicit pair-list responses. That this is not simply a pragmatic fact is indicated by the fact that in the same contexts unary wh-questions containing quantifiers are preferably responded to with pair-list (or functional) responses.

- (27) a. (Context: It is known that Cal phoned Ann or Ann phoned Cal.) Who phoned whom?
 b. (Same context:) Who did each person phone?
- (28) a. Well, it's Cal who phoned Ann.
 b. Cal phoned Ann, while Ann phoned Zhang.

In the context provided in (27), the response provided in (28a) would be the natural one. On the other hand, in the same context, the preferred response to (27b) is still (28b). Indirectly, this last utterance could be taken to resolve the disjunction, by implicating, for instance, that 'Fu is *not* chasing Li.' But that is fundamentally different information than the sparser information that (28a) conveys as a felicitous response to (27a).

4.2 Uniqueness relativisation and Bach—Peters sentences

'which phrases' where the common noun phrase is singular carry a uniqueness presupposition:⁸

- (29) a. Can you remind me please which Canadian athlete was suspended in 1988?
 b. Ben Johnson.
 c. Let's see, # Ben Johnson, Trevor Smith,...

However, in certain multiple 'which- sentences', as noted by Higginbotham and May 1981, the uniqueness can become relativised to yield a bijective pair-list:

- (30) a. Which Canadian athlete won which competition in 1988 ?
 b. Ben Johnson won the 100 meters, Debbie Brill won the high jump,...

The interesting point is that uniqueness does not disappear in all multiple wh-sentences. 'Crossing coreference' readings are available but do not elicit pair-list responses:

- (31) a. Which director who produced it backed out of which highly publicised movie that was causing him losses?
 b. Coppola was the director, the movie was Hammett.
 c. # Coppola (was the director who) backed out of Hammett, Ponti (was the director who) backed out of The Bicycle Thief,...

Quantifier/interrogative Bach-Peters sentences show markedly different behaviour:⁹ a crossed reading should correlate with a use where a dependency is established between the argument role associated with the interrogative and the argument role associated with the quantifier. In fact, the only available reading seems to be one where the interrogative role is *independent* of the quantifier role, that is, the interrogated argument role has widest scope. The pronoun in the restriction on the interrogative has to be interpreted deictically if the pronoun in the restriction on the quantifier is bound by the interrogative:¹⁰

- (32) a. Which director who produced it backed out of each highly publicized movie that was causing him losses?
 b. Coppola (was the director who backed out of all his loss-producing movies).
 c. # Coppola was the director, the movie was Hammett.
 d. # Coppola (was the director who) backed out of Hammett, Ponti (was the director who) backed out of The Bicycle Thief,...

This data is problematic for a view that *wh*-phrases are quantifiers. Quantificational approaches cannot generate a crossed scope reading for such a sentence using their 'quantifying into *Wh*' rule if it is assumed that both anaphora are bound variable: the variable corresponding to the narrowly scoped quantifier must remain unbound in the restrictive term of the wider scoping quantifier. Even if one assumes that interrogatives or nominal quantifiers do or need not scope each other strictly at a level of logical form (see e.g. Higginbotham and May 1981, May 1985.) or alternatively that one of the anaphora is generated by E-type anaphora, the essential asymmetry would appear to require differentiating the scoping mechanisms of the two expression types.

5 A non quantificational approach to interrogative meaning

5.1 Basics

The basic idea we propose is that interrogative phrases serve to specify focussed, unresolved argument roles: *Asking* a question has a clear effect on the context, an assertion of a declarative sentence made as a response to a query carries a different force than when uttered independently. This widely acknowledged property of responses is often expressed by saying that 'answers are foci'.

- (33) a. Who does Bill like?
 b. As response 'Bill likes Jill.' carries force of 'Bill likes JILL.' uttered independently.
 c. As response 'Bill likes no one.' carries force of 'Bill likes NO ONE' uttered independently.

Given that a query can be posed in a neutral context where no existential presupposition exists, this suggests that it must be something about the descriptive content of the query that gives rise to this effect. This focussing effect of queries, as well as evidence from anaphora and embedded interrogatives, which I will not enter into here, provides motivation for my assumption that the contents of uses of

interrogatives are 'lifted' variants of the contents we argued for in the first part of the paper:

- (34) a. $\langle \text{LIKES, liker:bill, likee:-HUMAN} \rangle$
 b. $\langle \lambda x \langle \text{LIKER, annoyed-person:bill, likee:xHUMAN} \rangle, x:- \rangle$

Notice though that the new and old unresolved interrogative contents are equivalent in terms of answerhood. For any SOA τ_0 that is a resolution of the lifted unresolved SOA_0 , there exists a distinct SOA τ_1 , that is a resolution of the unlifted unresolved SOA_1 , such that τ_1 describes information necessarily compatible with that of τ_0 , and vice versa.¹¹ For instance:

- (35) a. $\langle \text{LIKES, liker:bill likee: jill} \rangle$
 b. $\langle \lambda x \langle \text{LIKES, liker:bill likee:xHUMAN} \rangle, x: \text{jill} \rangle$

This fact, that argument roles associated with independent uses of interrogative phrases are focussed and hence take a widest sentential scope, is, I believe, one of the key facts about interrogative phrase meaning in general, and which, suitably modified, holds for all classes of interrogative phrase use. There are various cases where the argument role associated with the interrogative-phrase clearly does not take widest scope. Our basic claim, however, is that this has precisely two general causes: one is the effect of *embedding* an interrogative content as an argument of a predicate, in which case the role associated with the interrogative phrase gets widest *embedded* scope. Scoping effects that arise from 'echo' uses of interrogatives are also subsumed under this account.¹²

The other cause concerns uses in which the interrogative phrase is utilised to specify a functional dependency between argument roles, a type of use discussed extensively in Engdahl 1986. In such a case, it is the specification for the dependency that is focussed and receives widest sentential scope:

- (36) Which table does each person like.

In the current approach this gets the following schematic meaning:

- (37) $\langle \lambda h \langle \text{EACH, PERSON,} \rangle$
 $\langle \lambda x \langle \lambda z \langle \text{LIKE, subj-role:z, obj-role: h(z)} \rangle, z:x \rangle, h:- \rangle$

Multiple-wh-sentences can involve either contents where all the wh-phrases are used independently and don't strictly scope each other, or they can involve one of the wh-phrases being used dependently. Schematically:

- (38) a. Which person liked which table.
 b. $\langle \lambda x, y \langle \lambda z \langle \text{LIKE, subj-role:z,} \rangle$
 $\text{obj-role: y} \rangle, z:x \rangle, x:-, y:- \rangle$
 c. $\langle \lambda h, x \langle \lambda z \langle \text{LIKE, subj-role:z, obj-role: h(z)} \rangle, z:x \rangle, x:-, h:- \rangle$

The first type of content corresponds to the uses whose existence was brought out in the previous section in the disjunction resolution contexts and the crossing co-reference multiple-wh-questions. The second type of content allows us to account for the similarity in meaning that multiple-wh-sentences and quantifier/interrogative sentences can exhibit, in which the question specifies a dependency, which a response supplies either by providing either a pair-list or a functional response.^{13, 14}

5.2 A sketch of the compositional account

I now provide a brief and simplified sketch of the way the meaning of interrogative expressions is handled in the fragment provided in Ginzburg 1992, which assumes the input to meaning is a surface syntactic level, combined with a storage mechanism for treating scopal ambiguities. The semantic framework used there is the situation semantics framework of Gawron and Peters 1990a, while the syntactic framework is the HPSG framework of Pollard and Sag (forthcoming).

For current illustrative purposes, the only feature of the Gawron and Peters system we need mention concerns the notion of restrictions on variables. Where a variable is used to stand for an argument role *filler*, that variable is a *parameter of the meaning description*, that is, it signals that a value must be assigned to it in order that the content be determined. When a variable is introduced which is not a parameter of meaning, this means that it is being used to describe restrictions on what entities can fill the argument role with which it is associated. In such cases, the variable eventually gets bound by a binding operator in whose scope it lies.

Interrogative meaning is assumed to derive from two components: the one component is a variable together with associated restrictions contributed by the interrogative phrase. This acts as a place holder in the argument role associated with the interrogative phrase that adds restrictions to that argument role.

The other component is an indication¹⁵ at which sentential level the variable is input to a closure operation we call α -closure. This is a closure operation at a sentential level, which is argued to scope just wider than quantificational closure (the operator used to retrieve quantificational expressions from storage.) In the case of *independent* uses of interrogative phrases, the effect of α -closure is to create an unresolved SOA, whose main relation is obtained by abstracting over any variables filling argument roles associated with (independent) interrogative uses, and whose argument role(s) are unresolved. The operation defined is, thus, applicable to more than one variable simultaneously in which case a multiply unresolved SOA will result. A simple example is the following:

- (39) $\alpha x \langle \text{LIKES, liker:x, likee: jill} \rangle =_{def}$
 $\langle \lambda x \langle \text{LIKES, liker:x likee: jill} \rangle x:- \rangle$ (Rough meaning for 'Who likes Jill')

Here the SOA in which the variable x , associated with the **liker** argument role, was previously free, is mapped by α -abstraction to the unresolved SOA, whose main relation is the property of 'being a 'liker of jill'', and whose argument role is unresolved.

In general:

- (40) $\alpha x_1, \dots, x_n \langle Q, \dots r_1 : x_1, \dots, r_n : x_n \rangle =_{def}$
 $\langle \lambda x_1, \dots, x_n \langle Q, \dots r_1 : x_1, \dots, r_n : x_n \rangle, x_1:-, \dots, x_n:- \rangle$

Thus, we can provide the following (schematic) derivation of a unary wh-sentence meaning:

- (41) a. ['likes Bill'] ($dis - sit_0$) = $\lambda x \langle \text{LIKE, subj-role:x, obj-role: bill} \rangle$
 b. ['who']($dis - sit_0, domain - sit_0$) = t ;
 RESTRICTIONS: $t: domain - sit_0 \models \langle \text{PERSON, } t \rangle$
 c. ['who likes Bill']($dis - sit_0$) = $\alpha t \langle \lambda x \langle \text{LIKE, subj-role:x,}$

obj-role: bill $\rangle x:t$ $\rangle$;

RESTRICTIONS: $domain - sit_0 \models \langle PERSON, t \rangle$

With the same resources, we can provide for a (simplified) crossing coreference reading of a Bach-Peters sentence.¹⁶

- (42) a. ['Which movie that was causing him losses']($dis - sit_0, uniqueness - sit_0$) = s ;
 RESTRICTIONS: $s: uniqueness - sit_0 \models \langle \lambda z(MOVIE-CAUSING-LOSSES-TO, t, z), s \rangle$
- b. ['backed out of which movie that was causing him losses'] ($dis - sit_0$) = $\lambda x \langle \text{BACKED-OUT}, subj-role: x, obj-role: s \rangle$
- c. ['Which director who produced it']($dis - sit_0, uniqueness - sit_1$) = t ;
 RESTRICTIONS: $t: uniqueness - sit_1 \models \langle \lambda y(DIRECTOR-WHO-PRODUCED, s, y), t \rangle$
- d. ['Which director who produced it backed out of which movie that was causing him losses']($dis - sit_0$) = $\alpha s, t \langle \lambda x \langle \text{BACKED-OUT}, subj-role: x, obj-role: s \rangle x:t \rangle$ RESTRICTIONS: $s: uniqueness - sit_0 \models \langle \lambda z(MOVIE-CAUSING-LOSSES-TO, t, z), s \rangle$, $t: uniqueness - sit_1 \models \langle \lambda y(DIRECTOR-WHO-PRODUCED, s, y), t \rangle$

5.3 Dependent uses

The same basic principle of interrogative independent uses applies for dependent uses: a variable fills the argument role associated with the interrogative phrase. However, the idea is not that that argument role remain parametrised and create a new complex argument role. Rather, the idea is to create a SOA with an unresolved role for a dependency, constrained to relate the argument role associated with the interrogative and some other argument role(s). The issue, then, is this: is there any natural way of modifying independent use meaning in a way that will eliminate this scopal property of an interrogated argument role? The account proposed involves subsuming such uses under principles suggested by Gawron and Peters 1990a to deal with similar issues concerning bound-variable anaphora.¹⁷

A variable x that is a constituent of a meaning description will be said to be dependent on another variable y if y occurs in a condition restricting x . This is assumed to be a transitive relation: so if x is dependent on y and y is dependent on z , then x is dependent on z .

With the notion of dependence at hand, we can state a logical principle, postulated by Gawron and Peters to hold of Situation Theory, which they call the **Absorption principle**. We state this principle as follows:

If y depends on x and x has been absorbed (i.e. abstracted over or existentially quantified away), then y must be existentially quantified away.

What this is supposed to rule out are relations of the following schematic form:

- (43) $\lambda x \langle T \dots y_{\langle R, y, x \rangle} \dots \rangle$

A relation satisfying the absorption principle and differing minimally would be:

- (44) $\lambda x \exists y \langle T \dots y_{\langle R, y, x \rangle} \dots \rangle$

Apart from offering logical arguments for its need, Gawron and Peters also offer *linguistic* motivation for the principle from considerations pertaining to ellipsis and anaphora. In particular, they exploit it in their account of bound variable anaphora, which incorporates an insight from Heim 1982 concerning the anaphoric use that definites can have. Consider, for instance:

(45) Every dog wagged its tail.

This one sentence has two meanings, depending on the use to which the expression 'its tail' is put. Any use of the sentence according to the deictic use of 'its tail' provides for reference to the unique possessor of the tail, as indeed to the tail. Thus, s , the variable corresponding to the tail, and $uniqueness - sit_0$, the domain of uniqueness, are parameters of the meaning in such a case:

(46) ['Every dog wagged its tail.'](dis - sit_0) = $\langle$ EVERY, DOG, $\lambda x \langle \lambda z \langle$ WAG, wagger: z , wagged: s $\rangle$, $z:x$ $\rangle$ $\rangle$;
 RESTRICTIONS: $s: \langle$ TAIL, s $\rangle$, $\langle$ POSSESSION, poss-er: t , poss-ee: s $\rangle$; $t:$
 $uniqueness - sit_0 \models \langle$ INANIMATE, t $\rangle$ $uniqueness - sit_0: \langle$ UNIQUE,
 INANIMATE, $uniqueness - sit_0$ $\rangle$

On a bound-variable use of 'its tail' the following additional restriction is imposed:

(47) $\langle \models, uniqueness - sit_0 \langle =, t, z \rangle \rangle$

Imposing the dependency has a cascading effect: causing both $uniqueness - sit_0$ and s to be dependent on z . By the Absorption principle, this forces these two variables to be existentially quantified away, providing the following (schematic) meaning:

(48) $\langle$ EVERY, DOG, $\lambda x \langle \lambda z \exists s \exists uniqueness - sit_0 \langle$ WAG, wagger: z , wagged: s $\rangle$ $z:x$ $\rangle$ $\rangle$

The point is that $uniqueness - sit_0$ and s are no longer free in the meaning description, but depend on the subject argument. Hence the uniqueness that applies to the entity s that is a tail in $uniqueness - sit_0$ is not absolute, but is relative to every dog, the desired result.

The interrogative case works analogously. First the independent use:

(49) ['Which tail did every dog wag'](dis - sit_0) = $\alpha s \langle$ EVERY, DOG, $\lambda x \langle \lambda z \langle$ WAGGED, wagger: z , wagged: s $\rangle$, $z:x$ $\rangle$ $\rangle$
 RESTRICTIONS: $s: uniqueness - sit_0 \models \langle$ TAIL, s $\rangle$ $uniqueness - sit_0: \langle$ UNIQUE, TAIL, $uniqueness - sit_0$ $\rangle$

Notice, then, that in this case, $uniqueness - sit_0$ is a parameter of the meaning description, ensuring that whether or not some tail was actually wagged in $uniqueness - sit_0$, there was a unique tail involved.

When 'which tail' is used dependently on the subject role associated with 'each dog' a dependency condition is imposed on the interrogated role, just as an identity condition is imposed on roles associated with bound-variable uses:

(50) $\langle \models, uniqueness - sit_0 \langle =, h(z), s \rangle \rangle$

This causes both $uniqueness - sit_0$ and s to be dependent on z . By the Absorption principle, this forces these two variables to be existentially quantified away, providing the following schematic meaning description:

(51) ['Which tail did every dog wag'](dis - sit_0) = $\alpha h \langle$ EVERY, DOG, $\lambda x \langle \lambda z$

$\exists s \exists \text{uniqueness} - \text{sit}_0 \langle \text{WAGGED}, \text{wagger:}z, \text{wagged:s} \rangle, z:x \rangle$
 RESTRICTIONS: $s: \text{uniqueness} - \text{sit}_0 \models \langle \text{TAIL}, s \rangle \text{uniqueness} - \text{sit}_0: \langle$
 $\models, \text{uniqueness} - \text{sit}_0 \langle =, h(z), s \rangle \rangle$

So, the effect of the dependence condition has been this: it has forced the individual interrogative variable to be absorbed and leaves the functional dependence as the interrogative variable free to be α -closed, and hence create an unresolved role for a dependency.

6 Notes

¹I would like to thank Stanley Peters for many useful discussions and encouragement, and for comments or advice: Mark Crimmins, Polly Jacobson, Ivan Sag and Anna Szabolsci. Research supported by the Information Technology Promotion Agency, Japan, as part of the R&D of Basic Technology for Future Industries "New Models for Software Architecture" sponsored by NEDO (New Energy and Industrial Technology Development Organization).

²Though see e.g. Berman 1990 for a treatment of wh-phrases as variables within DRT, which for matrix questions build up Hamblin question objects. In Ginzburg 1992, I argue against Berman's account that in certain embedded interrogatives, adverbs do bind wh-phrases.

³In Ginzburg 1992 this discussion is extended to yes/no and alternative questions, for which analogous points to the one made here for wh-interrogatives can be made.

⁴See e.g. Devlin 1991 for a recent exposition.

⁵The basic idea for defining QUANT-ANS characterising quantified-answerhood is the following: it is intended to relate an unresolved SOA to each of the quantifications over the SOA's unfilled roles. The quantificational answers are SOA's that entail either a positive (i.e. instantiated) resolution of the issue raised by the unresolved SOA, or a negative resolution of that issue, or in the case of monotone decreasing quantificational forces, compatible with both positive and negative resolutions.

⁶See Ginzburg 1992 for further evidence in this regard and for discussion of restrictions that exist on such compounding.

⁷See Groenendijk and Stokhof 1989 and Ginzburg 1992 for discussion of the type-shifting operations that supra-propositional approaches to interrogatives require in order to get coordination to apply adequately even just among interrogatives.

⁸This places roughly the following obligation on the speaker: even if she does not know that an entity satisfying the queried property exists, this means that she knows of a domain in which an entity satisfying the restrictive condition exists uniquely, and she *strongly suspects* that it, whatever or whoever it is, does satisfy the queried property.

⁹See Ginzburg 1992 for additional data concerning the asymmetry.

- ¹⁰And this is not due to any subject/object asymmetry as indicated by parallel behaviour of
- (52) a. Which movie that was causing him losses did each director that produced it back out of?
 b. The movie that caused losses for all its directors was Hammett.
 c. # Coppola was the director, the movie was Hammett.
 d. # Coppola (was the director who) backed out of Hammett, Ponti (was the director who) backed out of The Bicycle Thief,...

¹¹The notion of informational compatability assumed here is captured situation theoretically by the notion of *logical involvement*, related to the traditional notion of entailment. See for instance Devlin 1991 for discussion.

¹²See Ginzburg 1992 for arguments that such uses should be accommodated by the grammar, and for a detailed semantics for such uses.

¹³See Chierchia 1991 and Ginzburg 1992 for arguments against assuming that the difference between such responses derives from an ambiguity.

¹⁴This latter example illustrates why uniqueness can be relativised, rather than absolute in multiple 'which' sentences. The property any filler of the role specified by the subject interrogative depends on the dependency specified by the role for dependency. Different resolutions of that dependency will yield potentially different resolutions of the filler of the unresolved subject slot. So, responses to such questions provide pairs of entities and dependencies uniquely associated with them. The prediction is that for any dependency provided, there will be a unique filler.

¹⁵Intuitively, this can be thought of as an instruction when to retrieve from storage. Formally this is captured by means of facts that serve to describe restrictions on the context of utterance.

¹⁶For simplicity, we project away here from the uniqueness presuppositions introduced by the determiner 'which'.

¹⁷In Ginzburg 1992 it is shown how this makes for predictions concerning the subject/object asymmetries characteristic of this use and the Bach-Peters asymmetries discussed above.

7 References

- Barwise, J. and Perry, J.: 1983, *Situations and Attitudes*, MIT Press, Cambridge, Mass.
- Berman, S.: 1990, Towards the Semantics of Open Sentences: Wh-phrases and Indefinites. in M. Stokhof and L. Torenvliet (eds.), *Proceedings of 7th Amsterdam Colloquium, December 19-22, 1989*, ITLI, Amsterdam
- Chierchia, G.: 1991, Functional Wh and Weak Crossover. in *Proceedings of WCCFL 10*
- Crimmins, M.: 1991, *States of Affairs without parameters*. Unpublished manuscript, Cornell University.
- Devlin, K.: 1991, *Logic and Information*, CUP, Cambridge.
- Engdahl, E.: 1986, *Constituent Questions*, Kluwer, Dordrecht.
- Gawron, M. and Peters, S.: 1990a, *Anaphora and Quantification in Situation Semantics*, CSLI publications, CSLI, Stanford, Ca.
- Ginzburg, J.: 1992, *Questions, Queries and Facts: a Semantics and Pragmatics for Interrogatives*, PhD. thesis, Stanford University.
- Groenendijk, J. and M. Stokhof. 1984. *Studies on the Semantics of Questions and the Pragmatics of Answers*, Amsterdam.
- Groenendijk, J. and M. Stokhof.: 1989, Type Shifting and the Semantics of Interrogatives. In Chierchia, G et al. (eds.) *Properties, Types, and Meaning, Vol 2*, Kluwer, Dordrecht.
- Hamblin, C.L.: 1973, Questions in Montague English, *Foundations of Language* 10(1), 41-53.
- Heim, I.: 1982, *The Semantics of Definite and Indefinite Noun Phrases*, Ph.D thesis, University of Massachusetts, Amherst, Published in 1989 by Garland, New York
- Higginbotham J. and May R.: 1981, Questions, quantifiers and crossing, *The Linguistic Review* 1(1), 41-80
- Jespersen, O.: 1965, *The Philosophy of Grammar*, Norton, New York
- Karttunen, L.: 1977, The Syntax and Semantics of Questions, *Linguistics and Philosophy* 1(1), 1-44
- May, R.: 1985, *Logical Form*, MIT Press, Cambridge, Mass.
- Pollard, C. and Sag, I.A.: (forthcoming), *Head Driven Phrase Structure Grammar*, Jointly published by: U. of Chicago Press, Chicago and CSLI, Stanford.

Lambek Semantics

Normalization, Spurious Ambiguity, Partial Deduction and Proof Nets

Herman Hendriks
ILLC, Department of Philosophy
University of Amsterdam

When Lambek (1958) introduced his syntactic calculus, he showed that it is equivalent to a sequent axiomatization, the Lambek-Gentzen sequent calculus (henceforth: **L**). The calculus **L** defines a general notion of derivability in the following sense: an expression consisting of the lexical items $e_1, \dots, e_n$ with respective categories $c_1, \dots, c_n$ is parsed as belonging to a certain category c if and only if the statement ' $c_1, \dots, c_n$ is a c ' (written as a so-called *sequent* $c_1, \dots, c_n \vdash c$) can be derived as a theorem of the system. Thus, grammatical derivations are reduced to logical deductions, giving rise to the slogan 'parsing as deduction'. The semantic interpretation procedure of Moortgat (1988) for product-free **L** is a sequent implementation of van Benthem's term construction algorithm for the natural deduction formulation of **L** (1986, chapter 7; 1991). In this semantics, the interpretations of a grammatical expression (derivable sequent) are directly determined by the proofs of its validity in the syntactic calculus. The calculus **L** contains, among other rules, the so-called *Cut* rule. By proving that the set of theorems of **L** is not increased by adding *Cut*, Lambek established the decidability of **L**: for any sequent, the proof procedure is guaranteed to answer the question whether the sequent is valid after a finite number of steps. The proof of this fact is constructive: Lambek defined an algorithm which eliminates all occurrences of *Cut* in proofs.

The present paper provides a modified formulation of the van Benthem/Moortgat semantics for **L** (which is introduced in Section 1) and uses it to prove the following results. Section 2 contains a straightforward proof of a semantic version of Lambek's *Cut* elimination theorem which entails that **L** is semantically decidable as well: the result of applying Lambek's *Cut* elimination algorithm is a derivation which is semantically equivalent to the original derivation. Section 3 is devoted to further normalization of **L**, giving rise to a sequent calculus **L*** which is put to use in the subsequent sections. First, we show in Section 4 that **L*** provides a solution to the so-called 'spurious ambiguity' problem – the problem that different proofs of a given sequent may yield one and the same semantic interpretation. In **L***, exactly one proof corresponds to each interpretation of a sequent. In Section 5, we use **L*** to prove that an explicit elaboration of the suggestions in Moortgat's 'Unambiguous Proof Representations for the Lambek Calculus' (1990) leads to proof representations which 'summarize' **L*** derivations and are, hence, indeed 'unambiguous', i.e., devoid of spurious ambiguity. On the other hand, a comparison of **L*** with the proof nets proposed in Roorda (1991) shows that every **L*** proof corresponds to a *set* of proof nets.

1 Semantics

We will employ the following definitions of the notions of category and sequent:

- (1) Let **ATOM** be some finite set of atomic categories. Then **CAT**, the set of categories based on **ATOM**, is the smallest set such that (i) $\text{ATOM} \subseteq \text{CAT}$, and (ii) if $a \in \text{CAT}$ and $b \in \text{CAT}$, then $(a/b) \in \text{CAT}$ and $(b \backslash a) \in \text{CAT}$.
- (2) A *sequent* is an expression $T \vdash c$, where T is a finite non-empty sequence of categories and $c \in \text{CAT}$. (So $T = c_1, \dots, c_n$, $n > 0$, and for all i , $1 \leq i \leq n$: $c_i \in \text{CAT}$.)

Outermost brackets of categories will be omitted. The sequence of categories T will be called the *left-hand side* of the sequent $T \vdash c$, and the single category c its *right-hand side* or *goal*. The product-free Lambek-Gentzen sequent calculus (**L**) consists of a set of *axioms* plus five *inference rules*: $/L$, $\backslash L$, $/R$, $\backslash R$ and *Cut*.

The axioms and rules are listed in (3) and (4). In (4), a, b , and c denote arbitrary categories and T, U , and V arbitrary finite sequences of categories, of which T is non-empty.

- (3) AXIOM, the set of *axioms* of L , is the set $\{c \vdash c \mid c \in \text{CAT}\}$.
- (4)
$$\frac{T \vdash b \quad U, a, V \vdash c}{U, a/b, T, V \vdash c} [/L]$$

$$\frac{T \vdash b \quad U, a, V \vdash c}{U, T, b \backslash a, V \vdash c} [\backslash L]$$

$$\frac{T \vdash a \quad U, a, V \vdash c}{U, T, V \vdash c} [Cut]$$

$$\frac{T, b \vdash a}{T \vdash a/b} [/R]$$

$$\frac{b, T \vdash a}{T \vdash b \backslash a} [\backslash R]$$

In the semantics given by Moortgat (1988), every category occurrence c in an L derivation is assigned some typed lambda term γ . The type of γ depends on c . For categories $c \in \text{ATOM}$, we assume some type assignment $\text{TYPE}(c)$, and the function TYPE is extended to non-atomic categories by the stipulation that $\text{TYPE}(a/b) = \text{TYPE}(b \backslash a) = (\text{TYPE}(b), \text{TYPE}(a))$. In this semantics, which is given in (5) below, the axioms of L correspond to *identity*, the rules $/L$ and $\backslash L$ to *functional application*, the rules $/R$ and $\backslash R$ to *lambda abstraction*, and the *Cut* rule to *substitution*. The respective assignment of terms $\gamma_1, \dots, \gamma_n, \gamma$ to categories $c_1, \dots, c_n, c$ in a derivable sequent $c_1, \dots, c_n \vdash c$ is usually written as $c_1 : \gamma_1, \dots, c_n : \gamma_n \vdash c : \gamma$.

- (5) SYNTAX:
- $$c \vdash c \text{ [AXIOM]}$$
- $$\frac{T \vdash b \quad U, a, V \vdash c}{U, a/b, T, V \vdash c} [/L]$$
- $$\frac{T \vdash b \quad U, a, V \vdash c}{U, T, b \backslash a, V \vdash c} [\backslash L]$$
- $$\frac{T, b \vdash a}{T \vdash a/b} [/R]$$
- $$\frac{b, T \vdash a}{T \vdash b \backslash a} [\backslash R]$$
- $$\frac{T \vdash a \quad U, a, V \vdash c}{U, T, V \vdash c} [Cut]$$
- SEMANTICS:
- $$\gamma \vdash \gamma$$
- $$\frac{T' \vdash \beta \quad U', \phi(\beta), V' \vdash \gamma}{U', \phi, T', V' \vdash \gamma}$$
- $$\frac{T' \vdash \beta \quad U', \phi(\beta), V' \vdash \gamma}{U', T', \phi, V' \vdash \gamma}$$
- $$\frac{T', v \vdash \alpha}{T' \vdash \lambda v. \alpha}$$
- $$\frac{v, T' \vdash \alpha}{T' \vdash \lambda v. \alpha}$$
- $$\frac{T' \vdash \alpha \quad U', \alpha, V' \vdash \gamma}{U', T', V' \vdash \gamma}$$

In (5), only the terms of the 'active' categories are indicated separately. The expression v denotes a variable of type $\text{TYPE}(b)$, while α, β, γ and ϕ denote arbitrary terms of type $\text{TYPE}(a), \text{TYPE}(b), \text{TYPE}(c)$, and $(\text{TYPE}(b), \text{TYPE}(a))$, respectively. The expressions T', U' and V' refer to the sequences of terms assigned to the sequences of 'inactive' categories T, U , and V . Thus, the terms assigned to the categories in T, U, V are identical in premise and conclusion. The interpretation which a proof with conclusion $c_1 : \gamma_1, \dots, c_n : \gamma_n \vdash c : \gamma$ assigns to a sequence of lexical expressions $e_1, \dots, e_n$ of categories $c_1, \dots, c_n$ with interpretations $e'_1, \dots, e'_n$ is obtained by performing the simultaneous substitution $[\gamma_1 := e'_1, \dots, \gamma_n := e'_n]$ in the term γ assigned to the conclusion sequent goal: $\gamma[\gamma_1 := e'_1, \dots, \gamma_n := e'_n]$.

Note that this is not always well-defined for semantics (5), which allows complex terms and different occurrences of the same term in the left-hand side of sequents. For (i) it is unclear what is meant by a substitution of a term for a complex term (this requires at least some extension of the notion of 'free occurrence of' to terms other than variables), and (ii) the notion of simultaneous substitution does not make sense when the same term is to be substituted twice. The problem reappears at deeper levels, e.g., when category c_i is assigned a term γ_i which also occurs as a subterm of a term γ_j assigned to a different

category c_j . Moreover, this substitution problem causes a complication in Moortgat's proof of 'semantic *Cut* elimination'.¹

These observations suggest that it is more natural to have a semantics which always yields proofs with a conclusion sequent $c_1 : v_1, \dots, c_n : v_n \vdash c : \gamma$, where $v_1, \dots, v_n$ are different variables. Therefore, the alternative semantics (6) which will be used in the following sections is based on the same category-to-type assignment as semantics (5), but differs from (5) in the terms it assigns as interpretations to the categories in a derivable sequent $c_1, \dots, c_n \vdash c$. For the left-hand side categories c_i ($1 \leq i \leq n$), this interpretation is some variable v_i of type $\text{TYPE}(c_i)$, whereas for the right-hand side category c , it is a (possibly complex) term γ of type $\text{TYPE}(c)$: $c_1 : v_1, \dots, c_n : v_n \vdash c : \gamma$. In (6), u, v, x and w represent variables of type $\text{TYPE}(a)$, $\text{TYPE}(b)$, $\text{TYPE}(c)$ and $(\text{TYPE}(b), \text{TYPE}(a))$; α, β and γ denote terms of type $\text{TYPE}(a)$, $\text{TYPE}(b)$ and $\text{TYPE}(c)$; and T', U' and V' refer to the sequences of variables assigned to the sequences of categories T, U and V . The expression $\gamma[u := \alpha]$ denotes the result of substituting the term α for all free occurrences of u in γ .

(6) SYNTAX:

$$\begin{array}{l} c \vdash c \text{ [AXIOM]} \\ \frac{T \vdash b \quad U, a, V \vdash c}{U, a/b, T, V \vdash c} [/L] \\ \frac{T \vdash b \quad U, a, V \vdash c}{U, T, b \backslash a, V \vdash c} [\backslash L] \\ \frac{T, b \vdash a}{T \vdash a/b} [/R] \\ \frac{b, T \vdash a}{T \vdash b \backslash a} [\backslash R] \\ \frac{T \vdash a \quad U, a, V \vdash c}{U, T, V \vdash c} [Cut] \end{array}$$

SEMANTICS:

$$\begin{array}{l} x \vdash x \\ \frac{T' \vdash \beta \quad U', u, V' \vdash \gamma}{U', w, T', V' \vdash \gamma[u := w(\beta)]} \\ \frac{T' \vdash \beta \quad U', u, V' \vdash \gamma}{U', T', w, V' \vdash \gamma[u := w(\beta)]} \\ \frac{T', v \vdash \alpha}{T' \vdash \lambda v. \alpha} \\ \frac{v, T' \vdash \alpha}{T' \vdash \lambda v. \alpha} \\ \frac{T' \vdash \alpha \quad U', u, V' \vdash \gamma}{U', T', V' \vdash \gamma[u := \alpha]} \end{array}$$

In the context of a proof, we will assume that all variables x and w assigned to an axiom instance or introduced in the conclusion of a $/L$ or $\backslash L$ inference are different. Observe that, consequently, (i) the variables $v_1, \dots, v_n$ assigned to the left-hand side of a sequent $c_1, \dots, c_n \vdash c$ are all different; (ii) these variables $v_1, \dots, v_n$ make up the free variables of the term γ assigned to the goal c ; and (iii) each variable $v_1, \dots, v_n$ occurs exactly once in γ .

Apart from the fact that semantics (6) is more in keeping with its intended use than semantics (5), it is also more 'compositional', in that the complexity of the term γ assigned to the goal of a sequent $c_1, \dots, c_n \vdash c$ increases with the length of the proof of that sequent. This property renders it more perspicuous, which facilitates proving things about it. In semantics (5), on the other hand, we see that the rule $/L$ assigns a term ϕ to its conclusion category a/b which is less complex than the term $\phi(\beta)$ assigned to the category a in its right-hand side premise (the same holds for $\backslash L$). Finally, semantics (6) is handier to use in practice, since it entails less duplication of semantic information than (5). Still, to the extent that semantics (5) conforms to its intended use, it is the same as semantics (6), on account of the following relationship. Let $\pi[c_1, \dots, c_n \vdash c]$ denote a proof of the conclusion $c_1, \dots, c_n \vdash c$, and let $v_1, \dots, v_n$ denote a sequence of different variables. Then

$$(7) \quad \pi[c_1 : v_1, \dots, c_n : v_n \vdash_{(5)} c : \gamma] \text{ iff } \pi[c_1 : v_1, \dots, c_n : v_n \vdash_{(6)} c : \gamma].$$

This entails that if the left-hand side terms of the conclusion sequent constitute a sequence of different variables, then the term which semantics (5) assigns to the conclusion goal is

identical to the term assigned by (6). Correspondence (7) can be shown by induction on the length of the proof π , using the FACT of footnote 1 as a lemma.

2 Cut Elimination and Semantics

Van Benthem (1986) proved that the number of distinct readings for an L -derivable sequent is finite. However, in addition one would like to have an effective method for generating all readings. The availability of such a method could serve practical purposes by providing a solution to the 'parsing problem', which is: '... to find all readings for a given string. This is a necessary step if one wants to give a satisfactory parsing algorithm for L . It should be noticed, however, that it is not at all clear which members of the infinite set of derivations for a sentence S in L lead to different readings, and thus, finding all possible readings seems to be problematic to begin with, let alone finding them efficiently' (Bouma (1989)).

A possible way of solving this problem is to prove that the result of applying Lambek's *Cut* elimination algorithm is a derivation which is semantically equivalent to the original derivation. Then for every derivation of an L -derivable sequent there is a *Cut*-free derivation with the same interpretation, and we can safely restrict our attention to the (finite number of) *Cut*-free derivations. Each of the inference rules $/L$, $\backslash L$, $/R$ and $\backslash R$ derives its conclusion from one or more premises with a strictly smaller number of occurrences of $/$ and $\backslash$. Hence, establishing the derivability of the premise(s) is really simpler than establishing the derivability of the conclusion, and it follows that every sequent has only finitely many *Cut*-free derivations. Adapting the proof of Lambek (1958), we will show in the present section that sequents in the proof of which the *Cut* inference rule has been used can also be derived without using *Cut* while keeping their semantics the same. That is, *Cut* is a derived rule of inference which does not yield new theorems, nor new interpretations. This is done in a constructive way: it is possible to transform every proof which makes use of the *Cut* rule into a *Cut*-free proof which determines the same interpretation as the original one. The two base cases of this transformation procedure (case 1 and 2 below) involve *Cut* inferences where one of the premises is an instance of the axiom scheme $c \vdash c$. Here, the conclusion coincides with the other premise, and the *Cut* inference can be eliminated immediately. The recursive cases of the procedure (case 3, 4 and 5 below) work by reduction of what is called the 'degree' of the *Cut* inference. Degree is defined as follows:

- (8) (i) The degree $d(c)$ of a category c is defined inductively:
 $d(c) = 0$ for $c \in \text{ATOM}$; $d(a/b) = d(b/a) = d(a) + d(b) + 1$.
(ii) The degree $d(c_1, \dots, c_n)$ of a finite sequence of categories $c_1, \dots, c_n$ equals $d(c_1) + \dots + d(c_n)$.
(iii) The degree $d(T \vdash c)$ of a sequent $T \vdash c$ equals $d(T) + d(c)$.
(iv) The degree $d(\alpha)$ of a *Cut* inference $\alpha = \frac{T \vdash a \quad U, a, V \vdash c}{U, T, V \vdash c}$ equals $d(T) + d(U) + d(V) + d(a) + d(c)$.

Any *Cut* inference of which at least one premise has been proven without *Cut* can be turned into one or two *Cut* inferences of strictly smaller degree. As the minimal degree of an inference is zero, this will ultimately converge on the base case. We will now prove that

- (1) in any application of *Cut* of which at least one premise has been proven without *Cut*
(a) the conclusion is identical with one of the premises, or
(b) the application can be replaced by 1 or 2 applications of *Cut* of smaller degree; and
(2) execution of this *Cut* elimination procedure always leads to semantically equivalent proofs. There are five main cases:

1: $T \vdash a$ is an axiom; then $T = a$ and the conclusion coincides with the premise $U, a, V \vdash c$.

$$\frac{a \vdash a \quad U, a, V \vdash c}{U, a, V \vdash c} [Cut]$$

$$\frac{u' \vdash u' \quad U', u, V' \vdash \gamma}{U', u', V' \vdash \gamma[u := u']}$$

Cut can be eliminated, for note that the variable u' is different from the variables U', u, V' . Hence we also have the interpretation $U', u, V' \vdash \gamma[u := u']$ for (the subproof of) the premise $U, a, V \vdash c$, due to the following fact (proven by induction on the length of π):

If $\pi[U : U', a : u, V : V' \vdash c : \gamma]$ and u' is a variable of type $\text{TYPE}(a)$ different from U', u, V' , then $\pi[U : U', a : u', V : V' \vdash c : \gamma[u := u']]$.

2: $U, a, V \vdash c$ is an axiom; then U and V are empty, $a = c$, and the conclusion coincides with the premise $T \vdash a$. *Cut* can be eliminated, for the terms α and $u[u := \alpha]$ coincide too:

$$\frac{T \vdash a \quad a \vdash a}{T \vdash a} [\text{Cut}] \quad \frac{T' \vdash \alpha \quad u \vdash u}{T' \vdash u[u := \alpha]}$$

3: The last step in the proof of $T \vdash a$ uses a rule but does not introduce the main connective of a . Then $T \vdash a$ is derived by $/L$ (or $\backslash L$) from two sequents, one of which has the form $T' \vdash a$, with $d(T') < d(T)$. Hence, we can reverse the order of the rules $/L$ (or $\backslash L$) and *Cut*, resulting in a new *Cut* inference which has smaller degree than the old one. We only consider $/L$. ($\backslash L$ is analogous.) Then $T = P, b/d, Q, R$, and we can replace:

$$\begin{array}{c} \frac{Q \vdash d \quad P, b, R \vdash a}{P, b/d, Q, R \vdash a} [/L] \quad U, a, V \vdash c \\ \hline U, P, b/d, Q, R, V \vdash c \quad [\text{Cut}] \end{array}$$

by:

$$\begin{array}{c} \frac{P, b, R \vdash a \quad U, a, V \vdash c}{U, P, b, R, V \vdash c} [\text{Cut}] \quad \frac{Q \vdash d}{U, P, b/d, Q, R, V \vdash c} [/L] \end{array}$$

$$\begin{array}{c} \frac{Q' \vdash \delta \quad P', v, R' \vdash \alpha}{P', w, Q', R' \vdash \alpha[v := w(\delta)]} \quad U', u, V' \vdash \gamma \\ \hline U', P', w, Q', R', V \vdash \gamma[u := \alpha[v := w(\delta)]] \end{array}$$

$$\begin{array}{c} \frac{Q' \vdash \delta \quad P', v, R' \vdash \alpha}{U', P', v, R', V \vdash \gamma[u := \alpha]} \quad U', u, V' \vdash \gamma \\ \hline U', P', w, Q', R', V \vdash [\gamma[u := \alpha]][v := w(\delta)] \end{array}$$

As v does not occur in γ , the terms $\gamma[u := \alpha[v := w(\delta)]]$ and $[\gamma[u := \alpha]][v := w(\delta)]$ are identical.

4: The last step in the proof of $U, a, V \vdash c$ uses one of the rules $/L, \backslash L, /R, \backslash R$, but does not introduce the main connective of a . Then $U, a, V \vdash c$ is inferred from one or two sequents, one of which has the form $U', a, V' \vdash c'$ with $d(U') + d(V') + d(c') < d(U) + d(V) + d(c)$. Again, we can reverse the order of the rules; the new *Cut* inference has smaller degree than the given one. We only consider $\backslash L$ and $\backslash R$. ($/L$ and $/R$ are analogous.)

4.1: $U, a, V \vdash c$ is obtained by $\backslash L$. Then $U, a, V = P, Q, d \backslash b, R$. Note that a and $d \backslash b$ are distinct category occurrences (for $\backslash L$ does not introduce the main connective of a). Therefore, there are three possibilities: a is part of P , Q or R (The last subcase, which is analogous to the first one, will not be treated.)

4.1.a: The category a is part of P , so $P = P_1, a, P_2$. Replace:

$$\begin{array}{c} \frac{Q \vdash d \quad P_1, a, P_2, b, R \vdash c}{P_1, a, P_2, Q, d \backslash b, R \vdash c} [\backslash L] \\ \hline P_1, T, P_2, Q, d \backslash b, R \vdash c \quad [\text{Cut}] \end{array}$$

by:

$$\begin{array}{c} \frac{T \vdash a \quad P_1, a, P_2, b, R \vdash c}{P_1, T, P_2, b, R \vdash c} [\text{Cut}] \quad \frac{Q \vdash d}{P_1, T, P_2, Q, d \backslash b, R \vdash c} [\backslash L] \end{array}$$

$$\begin{array}{c} \frac{Q' \vdash \delta \quad P'_1, u, P'_2, v, R' \vdash \gamma}{P'_1, u, P'_2, Q', w, R' \vdash \gamma[v := w(\delta)]} \\ \hline P'_1, T', P'_2, Q', w, R' \vdash [\gamma[v := w(\delta)]][u := \alpha] \end{array}$$

$$\begin{array}{c} \frac{T' \vdash \alpha \quad P'_1, u, P'_2, v, R' \vdash \gamma}{P'_1, T', P'_2, v, R' \vdash \gamma[u := \alpha]} \\ \hline P'_1, T', P'_2, Q', w, R' \vdash [\gamma[u := \alpha]][v := w(\delta)] \end{array}$$

The terms $[\gamma[v := w(\delta)]][u := \alpha]$ and $[\gamma[u := \alpha]][v := w(\delta)]$ are identical, because u does not occur in $w(\delta)$ and v does not occur in α .

4.1.b: The category a is part of Q , so $Q = Q_1, a, Q_2$. Replace:

$$\begin{array}{c} \frac{Q_1, a, Q_2 \vdash d \quad P, b, R \vdash c}{P, Q_1, a, Q_2, d \backslash b, R \vdash c} [\backslash L] \\ \hline P, Q_1, T, Q_2, d \backslash b, R \vdash c \quad [\text{Cut}] \end{array}$$

by:

$$\begin{array}{c} \frac{T \vdash a \quad Q_1, a, Q_2 \vdash d}{Q_1, T, Q_2 \vdash d} [\text{Cut}] \quad \frac{P, b, R \vdash c}{P, Q_1, T, Q_2, d \backslash b, R \vdash c} [\backslash L] \end{array}$$

$$\begin{array}{c} \frac{Q'_1, u, Q'_2 \vdash \delta \quad P', v, R' \vdash \gamma}{P', Q'_1, u, Q'_2, w, R' \vdash \gamma[v := w(\delta)]} \\ \hline P', Q'_1, T', Q'_2, w, R' \vdash [\gamma[v := w(\delta)]][u := \alpha] \end{array}$$

$$\begin{array}{c} \frac{T' \vdash \alpha \quad Q'_1, u, Q'_2 \vdash \delta}{Q'_1, T', Q'_2 \vdash \delta[u := \alpha]} \quad P', v, R' \vdash \gamma \\ \hline P', Q'_1, T', Q'_2, w, R' \vdash \gamma[v := w(\delta[u := \alpha])] \end{array}$$

$[\gamma[v := w(\delta)]] [u := \alpha]$ and $\gamma[v := w(\delta[u := \alpha])]$ are identical: $u \neq w$ and u does not occur in γ .
4.2: $U, a, V \vdash c$ is obtained by $\backslash R$. Then $c = d \backslash b$, and we can replace:

$$\begin{array}{c} \frac{T \vdash a \quad \frac{d, U, a, V \vdash b}{U, a, V \vdash d \backslash b} [\backslash R]}{U, T, V \vdash d \backslash b} [Cut] \\ \text{by:} \\ \frac{T \vdash a \quad \frac{d, U, a, V \vdash b}{d, U, T, V \vdash b} [Cut]}{U, T, V \vdash d \backslash b} [\backslash R] \end{array} \quad \begin{array}{c} \frac{v, U', u, V' \vdash \beta}{U', u, V' \vdash \lambda v. \beta} \\ U', T', V' \vdash [\lambda v. \beta] [u := \alpha] \\ \\ \frac{T' \vdash \alpha \quad v, U', u, V' \vdash \beta}{v, U', T', V' \vdash \beta [u := \alpha]} \\ U', T', V' \vdash \lambda v. \beta [u := \alpha] \end{array}$$

Since $u \neq v$, the terms $[\lambda v. \beta] [u := \alpha]$ and $\lambda v. \beta [u := \alpha]$ are identical.

5: The last steps in the proofs of both $T \vdash a$ and $U, a, V \vdash c$ introduce the main connective, $/$ or $\backslash$, of a . So $a = b/d$ or $a = d \backslash b$. (Only the $/$ -case is treated here.) Replace:

$$\begin{array}{c} \frac{T, d \vdash b}{T \vdash b/d} [/L] \quad \frac{V_1 \vdash d \quad U, b, V_2 \vdash c}{U, b/d, V_1, V_2 \vdash c} [/L] \\ \frac{}{U, T, V_1, V_2 \vdash c} [Cut] \\ \text{by:} \\ \frac{V_1 \vdash d \quad \frac{T, d \vdash b \quad U, b, V_2 \vdash c}{U, T, d, V_2 \vdash c} [Cut]}{U, T, V_1, V_2 \vdash c} [Cut] \end{array} \quad \begin{array}{c} \frac{T', v \vdash \beta}{T' \vdash \lambda v. \beta} \quad \frac{V'_1 \vdash \delta \quad U', u, V'_2 \vdash \gamma}{U', w, V'_1, V'_2 \vdash \gamma [u := w(\delta)]} \\ U', T', V'_1, V'_2 \vdash [\gamma [u := w(\delta)]] [w := \lambda v. \beta] \\ \\ \frac{T', v \vdash \beta \quad U', u, V'_2 \vdash \gamma}{V'_1 \vdash \delta \quad U', T', v, V'_2 \vdash \gamma [u := \beta]} \\ U', T', V'_1, V'_2 \vdash [\gamma [u := \beta]] [v := \delta] \end{array}$$

As w does not occur in γ , the terms $[\gamma [u := w(\delta)]] [w := \lambda v. \beta]$ and $[\gamma [u := [w(\delta)]]] [w := \lambda v. \beta]$ are identical. They are also identical to $\gamma [u := [\lambda v. \beta] (\delta)]$, for w does not occur in δ . Since all free variables in δ are free for v in β , we can apply β -conversion, with the result $[\gamma [u := \beta [v := \delta]]]$.² As v does not occur in γ , this is the same as $[\gamma [u := \beta]] [v := \delta]$. $\square$

3 Further Normalization

We have shown that if $c_1 : v_1, \dots, c_n : v_n \vdash c : \gamma$ is derivable in L , then there is a *Cut*-free L derivation of $c_1 : v_1, \dots, c_n : v_n \vdash c : \gamma'$ such that γ and γ' are equivalent. Consequently, we only have to consider the *Cut*-free derivations of a sequent to obtain all its semantic interpretations. But in spite of the fact that a given sequent has only finitely many *Cut*-free derivations, *Cut*-less L still suffers from what has been called the 'spurious ambiguity problem' (König (1989)), viz., the problem that different proofs of a given sequent may yield one and the same semantic interpretation. When the calculus is actually used for parsing and interpreting expressions, this may entail superfluous work: interpretations will be computed n times instead of once. For example, the sequent $c/c, c/c, c \vdash c$ has two semantically equivalent *Cut*-free proofs: (9) and (10), in which the symbol $!$ is used to keep track of the active category in each step of the derivations.

$$\begin{array}{c} (9) \quad \frac{\frac{c \vdash c \quad c \vdash c}{c/c!, c \vdash c} [/L] \quad c \vdash c}{c/c!, c/c, c \vdash c} [/L] \quad \frac{\frac{x \vdash x \quad u \vdash u}{y, x \vdash y(x)} [/L] \quad v \vdash v}{z, y, x \vdash z(y(x))} [/L] \\ \\ (10) \quad \frac{c \vdash c \quad \frac{c \vdash c \quad c \vdash c}{c/c!, c \vdash c} [/L]}{c/c, c/c!, c \vdash c} [/L] \quad \frac{\frac{u \vdash u \quad v \vdash v}{z, u \vdash z(u)} [/L] \quad x \vdash x}{z, y, x \vdash z(y(x))} [/L] \end{array}$$

Of course, not all ambiguity in L is spurious. Some sequents are really ambiguous. The sequent $s/(n \backslash s), (n \backslash s)/n, (s/n) \backslash s \vdash s$, for example, has the following six *Cut*-free derivations

(each derivation is followed by the interpretation it assigns to its conclusion sequent):

$$\begin{array}{ll}
(11) \frac{\frac{n \vdash n \quad s \vdash s \quad [\wedge L]}{n \vdash n \quad n, n \backslash s^\dagger \vdash s \quad [\wedge L]} \quad \frac{n, (n \backslash s)/n^\dagger, n \vdash s \quad [\wedge R]}{n, (n \backslash s)/n \vdash s/n^\dagger \quad [\wedge R]} \quad \frac{s \vdash s \quad [\wedge L]}{n, (n \backslash s)/n, (s/n) \backslash s^\dagger \vdash s \quad [\wedge L]} \quad \frac{s \vdash s \quad [\wedge R]}{(n \backslash s)/n, (s/n) \backslash s \vdash n \backslash s^\dagger \quad [\wedge R]} \quad \frac{s \vdash s \quad [\wedge L]}{s/(n \backslash s)^\dagger, (n \backslash s)/n, (s/n) \backslash s \vdash s \quad [\wedge L]} \\
u, v, w \vdash u(\lambda u'. w(\lambda w'. v(w')(u'))))
\end{array}
\quad
\begin{array}{ll}
(12) \frac{\frac{n \vdash n \quad s \vdash s \quad [\wedge L]}{n, n \backslash s^\dagger \vdash s \quad [\wedge R]} \quad \frac{n \backslash s \vdash n \backslash s^\dagger \quad s \vdash s \quad [\wedge R]}{n \vdash n \quad s/(n \backslash s)^\dagger, n \backslash s \vdash s \quad [\wedge L]} \quad \frac{s/(n \backslash s), (n \backslash s)/n^\dagger, n \vdash s \quad [\wedge L]}{s/(n \backslash s), (n \backslash s)/n \vdash s/n^\dagger \quad [\wedge R]} \quad \frac{s \vdash s \quad [\wedge L]}{s/(n \backslash s), (n \backslash s)/n \vdash s/n^\dagger \quad [\wedge R]} \quad \frac{s \vdash s \quad [\wedge L]}{s/(n \backslash s), (n \backslash s)/n, (s/n) \backslash s^\dagger \vdash s \quad [\wedge L]} \\
u, v, w \vdash w(\lambda w'. u(\lambda u'. v(w')(u'))))
\end{array}$$

$$\begin{array}{ll}
(13) \frac{\frac{n \vdash n \quad s \vdash s \quad [\wedge L]}{n, n \backslash s^\dagger \vdash s \quad [\wedge R]} \quad \frac{n \vdash n \quad n \backslash s \vdash n \backslash s^\dagger \quad [\wedge L]}{(n \backslash s)/n^\dagger, n \vdash n \backslash s \quad [\wedge L]} \quad \frac{s/(n \backslash s)^\dagger, (n \backslash s)/n, n \vdash s \quad [\wedge R]}{s/(n \backslash s), (n \backslash s)/n \vdash s/n^\dagger \quad [\wedge R]} \quad \frac{s \vdash s \quad [\wedge L]}{s/(n \backslash s), (n \backslash s)/n, (s/n) \backslash s^\dagger \vdash s \quad [\wedge L]} \\
u, v, w \vdash w(\lambda w'. u(\lambda u'. v(w')(u'))))
\end{array}
\quad
\begin{array}{ll}
(14) \frac{\frac{n \vdash n \quad s \vdash s \quad [\wedge L]}{n \vdash n \quad n, n \backslash s^\dagger \vdash s \quad [\wedge L]} \quad \frac{n, (n \backslash s)/n^\dagger, n \vdash s \quad [\wedge R]}{(n \backslash s)/n, n \vdash n \backslash s^\dagger \quad [\wedge R]} \quad \frac{s \vdash s \quad [\wedge L]}{s/(n \backslash s)^\dagger, (n \backslash s)/n, n \vdash s \quad [\wedge R]} \quad \frac{s/(n \backslash s), (n \backslash s)/n \vdash s/n^\dagger \quad [\wedge R]}{s/(n \backslash s), (n \backslash s)/n, (s/n) \backslash s^\dagger \vdash s \quad [\wedge L]} \\
u, v, w \vdash w(\lambda w'. u(\lambda u'. v(w')(u'))))
\end{array}$$

$$\begin{array}{ll}
(15) \frac{\frac{n \backslash s \vdash n \backslash s \quad s \vdash s \quad [\wedge L]}{n \vdash n \quad s/(n \backslash s)^\dagger, n \backslash s \vdash s \quad [\wedge L]} \quad \frac{s/(n \backslash s), (n \backslash s)/n^\dagger, n \vdash s \quad [\wedge R]}{s/(n \backslash s), (n \backslash s)/n \vdash s/n^\dagger \quad [\wedge R]} \quad \frac{s \vdash s \quad [\wedge L]}{s/(n \backslash s), (n \backslash s)/n, (s/n) \backslash s^\dagger \vdash s \quad [\wedge L]} \\
u, v, w \vdash w(\lambda w'. u(v(w'))))
\end{array}
\quad
\begin{array}{ll}
(16) \frac{\frac{n \vdash n \quad n \backslash s \vdash n \backslash s \quad [\wedge L]}{(n \backslash s)/s^\dagger, n \vdash n \backslash s \quad [\wedge L]} \quad \frac{s/(n \backslash s)^\dagger, (n \backslash s)/n, n \vdash s \quad [\wedge R]}{s/(n \backslash s), (n \backslash s)/n \vdash s/n^\dagger \quad [\wedge R]} \quad \frac{s \vdash s \quad [\wedge L]}{s/(n \backslash s), (n \backslash s)/n, (s/n) \backslash s^\dagger \vdash s \quad [\wedge L]} \\
u, v, w \vdash w(\lambda w'. u(v(w'))))
\end{array}$$

The terms assigned to the conclusions of the derivations (12), (13) and (14) are identical. Moreover, since the subterm $\lambda u'. v(w')(u')$ of the term $w(\lambda w'. u(\lambda u'. v(w')(u')))$ assigned to the conclusion goal of these derivations is equivalent to $v(w')$ by η -conversion,³ the interpretation of these conclusions is equivalent to $u, v, w \vdash w(\lambda w'. u(v(w')))$, the interpretation of the conclusion goal of (15) and (16). So, the sequent $s/(n \backslash s), (n \backslash s)/n, (s/n) \backslash \vdash s$ has five equivalent *Cut*-free derivations. However, interpretation $u(\lambda u'. w(\lambda w'. v(w')(u')))$ of the conclusion of derivation (11) is *not* equivalent to the interpretation of the other ones. This can be seen as follows. If the expressions *someone*, *pleases* and *everyone* are assigned categories and translations as in (17), then performing the simultaneous substitution (18) in the term of the conclusion goal in (11) supplies the sentence *Someone pleases everyone* with interpretation (i) $\exists x \forall y \text{ PLEASE}(y)(x)$, whereas performing this substitution in the terms of the conclusion goal in (12) through (16) yields interpretation (ii) $\forall y \exists x \text{ PLEASE}(y)(x)$.

(17)	EXPRESSION	CATEGORY	INTERPRETATION
	<i>someone</i>	$s/(n \backslash s)$	$\lambda P. \exists x P(x)$
	<i>pleases</i>	$(n \backslash s)/n$	PLEASE
	<i>everyone</i>	$(s/n) \backslash s$	$\lambda P. \forall y P(y)$

$$(18) \quad [u := \lambda P. \exists x P(x), v := \text{PLEASE}, w := \lambda P. \forall y P(y)]$$

This example gives a nice illustration of what spurious ambiguity comes down to: interpretation (i) is derived only once, but interpretation (ii) is associated with five derivations.

Clearly, if we want to put the calculus **L** to use in parsing and interpreting expressions, this is going to be a problem: (ii) will be computed 5 times instead of once.

König (1989) is the first attempt to solve the problem, but it did not succeed completely. For instance, König's system allows the two derivations (9) and (10) for the unambiguous sequent $c/c, c/c, c \vdash c$. In the sequel we will show how this problem can be solved for **L** by further restricting the *Cut*-free calculus. The resulting system **L***, which is equivalent to the system which was independently proposed by Hepple (1990), is a solution to the spurious ambiguity problem: it provides only one proof per interpretation. The 'non-spuriously ambiguous' calculus **L***, which will be presented below, is based on three observations:

A: First, each non-atomic axiom instance $b \backslash a \vdash b \backslash a$ (and, analogously, $a/b \vdash a/b$) with interpretation $w \vdash w$ can be decomposed into a proof with two less complex axiom premises:

$$\frac{\frac{b \vdash b \quad a \vdash a}{b, b \backslash a \vdash a} [\backslash L]}{b \backslash a \vdash b \backslash a} [\backslash R] \qquad \frac{\frac{u \vdash u \quad v \vdash v}{u, w \vdash w(u)} [\backslash L]}{w \vdash \lambda u. w(u)} [\backslash R]$$

Due to η -conversion, the term $\lambda u. w(u)$ is equivalent to w .

B: Second, if a $\backslash R$ or $/R$ inference yields the right premise of a $/L$ or $\backslash L$ inference, we can always reverse the order of the rules, whereas the semantics remains the same. We only consider the $(/R, /L)$ -case, where the rules involved are $/R$ and $/L$, respectively. (The cases $(/R, \backslash L)$, $(\backslash R, /L)$ and $(\backslash R, \backslash L)$ are analogous.) We can replace:

$$(19) \quad \frac{\frac{U, c, V, b \vdash a}{T \vdash d \quad U, c, V \vdash a/b} [/R]}{U, c/d, T, V \vdash a/b} [/L] \qquad \frac{\frac{U', z, V', y \vdash \alpha}{U' \vdash \delta \quad U', z, V' \vdash \lambda y. \alpha}}{U', v, T', V' \vdash [\lambda y. \alpha][z := v(\beta)]}$$

by:

$$\frac{\frac{T \vdash d \quad U, c, V, b \vdash a}{U, c/d, T, V, b \vdash a} [/L]}{U, c/d, T, V \vdash a/b} [/R] \qquad \frac{\frac{T' \vdash \delta \quad U', z, V', y \vdash \alpha}{U' \vdash \delta \quad U', v, T', V' \vdash \alpha[z := v(\beta)]}}{U', v, T', V' \vdash \lambda y. \alpha[z := v(\beta)]}$$

The terms $[\lambda y. \alpha][z := v(\delta)]$ and $\lambda y. \alpha[z := v(\delta)]$ are identical, as z and y are different variables.

C: Third, whenever a $\backslash L$ or $/L$ inference yields the right premise of another $\backslash L$ or $/L$ inference, and both inferences have *different* active categories,⁴ we can reverse the order of the inferences and shift the latter application to the left premise (cf. (20)) or the right premise (cf. (21)) of the former one. We only treat the $(/L, \backslash L)$ -case, where the rules are $/L$ and $\backslash L$, respectively. (The cases $(/L, /L)$, $(\backslash L, /L)$ and $(\backslash L, \backslash L)$ are analogous.)

So, suppose that conclusion $X, W, d \backslash c, Z \vdash e$ has been derived from left premise $W \vdash d$ and right premise $X, c, Z \vdash e$, and that $X, c, Z \vdash e$ is the result of applying $/L$ with an active category a/b different from c – that is, a/b occurs in X or Z :

$$\frac{W \vdash d \quad \frac{A \vdash b \quad B, a, C \vdash e}{X, c, Z \vdash e} [/L]}{X, W, d \backslash c, Z \vdash e} [\backslash L]$$

Then there are three possibilities: c occurs in A (see (20)), or c does not occur in A , and then c occurs in B (see (21)) or in C (analogous to (21)). In all cases we can reverse the order of the rules while keeping the same semantics. If c occurs in A , we can replace:

$$(20) \quad \frac{\frac{S, c, T \vdash b \quad U, a, V \vdash e}{W \vdash d \quad U, a/b, S, c, T, V \vdash e} [/L]}{U, a/b, S, W, d \backslash c, T, V \vdash e} [\backslash L] \qquad \frac{\frac{S', z, T' \vdash \beta \quad U', x, V' \vdash \epsilon}{U' \vdash \delta \quad U', y, S', z, T', V' \vdash \epsilon[x := y(\beta)]}}{U', y, S', W', v, T', V' \vdash [\epsilon[x := y(\beta)]] [z := v(\delta)]}$$

by:

$$\frac{\frac{W \vdash d \quad S, c, T \vdash b}{S, W, d \backslash c, T \vdash b} [\backslash L] \quad U, a, V \vdash e}{U, a/b, S, W, d \backslash c, T, V \vdash e} [/L] \qquad \frac{\frac{W' \vdash \delta \quad S', z, T' \vdash \beta}{S', W', v, T' \vdash \beta[z := v(\delta)]} \quad U', x, V' \vdash \epsilon}{U', y, S', W', v, T', V' \vdash \epsilon[x := y(\beta[z := v(\delta)])]}$$

The terms $[\epsilon[x := y(\beta)]] [z := v(\delta)]$ and $\epsilon[x := y(\beta[z := v(\delta)])]$ are identical, because $z \neq y$ and z does not occur in ϵ . If c occurs in B , we can replace:

$$(21) \quad \frac{T \vdash b \quad U, c, Y, a, V \vdash e}{W \vdash d} [L] \quad \frac{U, c, Y, a/b, T, V \vdash e}{U, W, d \setminus c, Y, a/b, T, V \vdash e} [\setminus L] \quad \frac{T' \vdash \beta \quad U', z, Y', x, V' \vdash \epsilon}{U', z, Y', y, T', V' \vdash \epsilon[x := y(\beta)]} \quad \frac{W' \vdash \delta}{U', W', v, Y', y, T', V'' \vdash [\epsilon[x := y(\beta)]] [z := v(\delta)]}$$

by:

$$\frac{T \vdash b \quad \frac{W \vdash d \quad U, c, Y, a, V \vdash e}{U, W, d \setminus c, Y, a, V \vdash e} [\setminus L]}{U, W, d \setminus c, Y, a/b, T, V \vdash e} [L] \quad \frac{T' \vdash \beta \quad \frac{W' \vdash \delta \quad U', z, Y', x, V' \vdash \epsilon}{U', W', v, Y', x, V' \vdash \epsilon[z := v(\delta)]}}{U', W', v, Y', y, T', V' \vdash [\epsilon[z := v(\delta)]] [x := y(\beta)]}$$

The terms $[\epsilon[x := y(\beta)]] [z := v(\delta)]$ and $[\epsilon[z := v(\delta)]] [x := y(\beta)]$ are identical, since $z \neq y$, $x \neq v$, z does not occur in β , and x does not occur in δ .⁵

Observation A entails that for every proof of a sequent, there is a semantically equivalent proof π in which all axiom instances are atomic: $at \vdash at$. For note that each non-atomic axiom instance can be decomposed in the way sketched above, and that every decomposition strictly decreases the sum of the degrees of the axiom instances. Hence, every proof can be transformed into such an equivalent proof π after finitely many decompositions. Moreover, given such a π , we can use observations B and C for obtaining a further normalized proof π'' , in which (i) no *major* (right-hand side) premise of a $\setminus L$ or $/L$ inference is the conclusion of a $\setminus R$ or $/R$ inference;⁶ and (ii) the same left-hand side category remains active whenever one goes down from axioms via major premises of $\setminus L$ and $/L$ inferences.

In order to see this, we define the notion 'maximal major path' (mmp) for *Cut*-free **L** proofs π : an mmp is a sequence $\langle \sigma_0, \dots, \sigma_p \rangle$ ($p \in \mathbb{N}$) of sequent occurrences in π , such that (i) σ_0 is an axiom instance; (ii) for all i , $0 \leq i \leq p$, σ_{i+1} is the conclusion of an inference with sole ($\setminus R$ or $/R$) or major ($\setminus L$ or $/L$) premise σ_i ; and (iii) σ_p is not the sole or major premise of an inference.⁷ Every sequent occurrence in a proof is on exactly one mmp, and every proof contains as many mmps as it contains axiom instances, i.e., at least one.

Now, consider a proof π with $n+1$ mmps, and focus on the main mmp $\mu = \langle \sigma_0, \dots, \sigma_p \rangle$ which culminates in the final conclusion σ_p of π . Let q be the number of $\setminus L$ and $/L$ inferences in μ , and r the number of $\setminus R$ and $/R$ inferences in μ . So, $\sigma_p = \sigma_{q+r}$, some sequent $T_{q+r} \vdash c_{q+r}$, and the sequent σ_0 is an axiom $c_0 \vdash c_0$. Moreover, for $0 \leq i \leq q+r$, (a) each σ_{i+1} obtained by $\setminus R$ or $/R$ is concluded from a subproof π_i of π with conclusion σ_i , and (b) each σ_{i+1} ($1 \leq i \leq q+r$) obtained by $\setminus L$ or $/L$ is concluded from two subproofs of π : π'_j (where $1 \leq j \leq q$) and π_i (with conclusion σ_i):

$$(a) \quad \frac{\frac{\vdots}{\sigma_i}}{\sigma_{i+1}} \pi_i \quad (b) \quad \frac{\pi'_j \quad \frac{\vdots}{\sigma_i}}{\sigma_{i+1}} \pi_i$$

The left-hand side subproofs π'_1 through π'_q have a total number of n mmps, which is one less than the whole proof π . Next, let $k(\mu) \leq r \cdot q$ be the number of pairs of sequents σ_i and σ_j in μ such that (i) $1 \leq i < j \leq q+r$; (ii) σ_i is the conclusion of a $\setminus R$ or $/R$ inference; and (iii) σ_j is the conclusion of a $\setminus L$ or $/L$ inference. Note that an application of transformation (19) to a $\setminus R$ or $/R$ inference which yields the major premise of a $\setminus L$ or $/L$ inference in an mmp ν always leads to an mmp ν' which (i) has the same length, (ii) has the same σ_{q+r} , (iii) builds on the same left-hand side subproofs $\pi'_1, \dots, \pi'_q$, and (iv) assigns the same interpretation to σ_{q+r} as ν ; but that $k(\nu') = k(\nu) - 1$:

$$\frac{\pi'_j \quad \frac{\pi_{i-1}}{\sigma_i}}{\sigma_{i+1}} \quad \rightsquigarrow_{(19)} \quad \frac{\pi'_j \quad \pi_{i-1}}{\frac{\sigma'_i}{\sigma_{i+1}}}$$

So, after $k(\mu)$ applications of transformation (19), we obtain a semantically equivalent proof π' of $T_{q+r} \vdash c_{q+r}$ from axiom $c_0 \vdash c_0$ such that in the main mmp μ' of π' no $\setminus R$ or $/R$

inference precedes a $\backslash L$ or $/L$ inference, i.e., $k(\mu') = 0$:

$$\begin{array}{c}
 \frac{\pi'_1}{T'_1 \vdash c'_1} \text{ [AXIOM]} \\
 \vdots \\
 \frac{\pi'_q}{T'_q \vdash c'_q} \\
 \vdots \\
 \frac{T'_{(q+r)-1} \vdash c'_{(q+r)-1}}{T_{q+r} \vdash c_{q+r}}
 \end{array}
 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} \backslash L \text{ and } /L \\ \\ \backslash R \text{ and } /R \end{array}$$

Focus on the sequence $\nu = \langle c_0 \vdash c_0, \dots, T'_q \vdash c'_q \rangle$, i.e., the main mmp of the subproof of π' with conclusion $T'_q \vdash c'_q$, and consider the first inference $\$L$ in ν with a different active category than the inference $\#L$ yielding the major premise of $\$L$ ($\$$ and $\# \in \{\backslash, /\}$). We can apply transformation (20) or (21). If (21) is applicable, inference $\$L$ reappears one place higher up in the main mmp, now yielding the major premise of $\#L$, and again having a different active category than the inference now yielding its major premise (footnote 5 (ii)):

$$\frac{\pi'_{j+1} \quad \frac{\pi'_j \quad \pi_{i-1}}{\sigma_i} [\#L]}{\sigma_{i+1}} [\$L] \quad \rightsquigarrow_{(21)} \quad \frac{\pi'_{j+1} \quad \pi_{i-1}}{\sigma_{i+1}} [\$L] \quad \frac{\pi'_j}{\sigma_i} [\#L]$$

Note that an application of transformation (21) to an mmp ν of length q always leads to an mmp with (i) the same length, (ii) the same σ_q . (iii) the same left-hand side subproofs $\pi'_1, \dots, \pi'_q$ (though differently ordered), and (iv) the same interpretation for σ_q . Since the new situation is structurally the same, transformation (21) can be applied again and again, until transformation (20) is applicable – for $\$L$ yielding conclusion σ_i ($2 \leq i \leq q$), this happens after at most i steps: when $\$L$ yields σ_2 , transformation (20) must be applied (cf. footnote 5 (i)). Transformation (20) removes inference $\$L$ from the mmp and adds it, together with its left-hand side subproof π'_{j+1} , to the left-hand side subproof π'_j of $\#L$, where the newly created left-hand side subproof π'_j has as many mmps as π'_j and π'_{j+1} together.⁸

$$\frac{\pi'_{j+1} \quad \frac{\pi'_j \quad \pi_{i-1}}{\sigma_i} [\#L]}{\sigma_{i+1}} [\$L] \quad \rightsquigarrow_{(20)} \quad \pi'_j \left[\frac{\pi'_{j+1} \quad \pi'_j}{\sigma_i} [\$L] \right] \frac{\pi_{i-1}}{\sigma_{i+1}} [\#L]$$

The resulting mmp ν' is one sequent shorter than ν , but its σ'_{q-1} equals σ_q of ν , and receives the same interpretation. The procedure can be reapplied to the first inference $\$L'$ in ν' with a different active category than the inference $\#L'$ yielding its major premise. If this is done s times, where s is the number of $\backslash L$ and $/L$ inferences having a different active category than the first inference in ν (thus $0 \leq s \leq q-1$), we obtain a semantically equivalent proof π'' of $T_{q+r} \vdash c_{q+r}$ such that in μ'' , the main mmp of π'' , no $\backslash R$ or $/R$ inference precedes a $\backslash L$ or $/L$ inference, and all $\backslash L$ and $/L$ inferences have the same active category:

$$\begin{array}{c}
 \frac{\pi''_1}{T'_1 \vdash c'_1} \text{ [AXIOM]} \\
 \vdots \\
 \frac{\pi''_{q-s}}{T'_q \vdash c'_q} \\
 \vdots \\
 \frac{T'_{(q+r)-1} \vdash c'_{(q+r)-1}}{T_{q+r} \vdash c_{q+r}}
 \end{array}
 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} \backslash L \text{ and } /L: \text{ same active category} \\ \\ \backslash R \text{ and } /R \end{array}$$

The finite procedure which led from π to π'' can be repeated for the left-hand side subproofs $\pi''_1, \dots, \pi''_{q-s}$ of π'' . Moreover, the task of normalizing the proofs π''_1 through π''_{q-s} involves

less work than our initial task of normalizing π , because we started with a proof with $n + 1$ mmps, whereas the total number of mmps in π_1'' through π_{q-s}'' equals n .

The above considerations can be summarized in the form of the calculus $\mathbf{L}^*$, which is given in (22) below. This calculus introduces the symbol $*$, which controls the activity of categories in derivations. Unlike the symbol $\dagger$, which was merely added to the $\mathbf{L}$ derivations (9) through (16) in order to enhance their readability, the symbol $*$ is a substantial ingredient of $\mathbf{L}^*$ derivations, since there is a special rule which deals with its behaviour:

$$(22) \quad \begin{array}{l} at^* \vdash at \text{ [AXIOM]} \\ \frac{T \vdash b^* \quad U, a^*, V \vdash at}{U, a/b^*, T, V \vdash at} [/L] \\ \frac{T \vdash b^* \quad U, a^*, V \vdash at}{U, T, b \backslash a^*, V \vdash at} [\backslash L] \\ \frac{T, b \vdash a^*}{T \vdash a/b^*} [/R] \\ \frac{b, T \vdash a^*}{T \vdash b \backslash a^*} [\backslash R] \\ \frac{U, a^*, V \vdash at}{U, a, V \vdash at^*} [*] \end{array} \quad \begin{array}{l} x \vdash x \\ \frac{T' \vdash \beta \quad U', u, V' \vdash \gamma}{U', w, T', V' \vdash \gamma[u := w(\beta)]} \\ \frac{T' \vdash \beta \quad U', u, V' \vdash \gamma}{U', T', w, V' \vdash \gamma[u := w(\beta)]} \\ \frac{T', v \vdash \alpha}{T' \vdash \lambda v. \alpha} \\ \frac{v, T' \vdash \alpha}{T' \vdash \lambda v. \alpha} \\ \frac{U', u, V' \vdash \gamma}{U', u, V' \vdash \gamma} \end{array}$$

The calculus in (22) observes the same conventions as regards categories, types, terms and variables as (6) above, with the exceptions that at represents an arbitrary atomic category, that γ refers to a term of type $\text{TYPE}(at)$, and that x denotes a variable of type $\text{TYPE}(at)$. Important properties of the calculus $\mathbf{L}^*$ are the following. Let $\gamma' = \gamma$ denote equivalence of terms γ' and γ (we will use $\gamma' \equiv \gamma$ for syntactic identity):

1. $T \vdash_{\mathbf{L}^*} c^*$ if and only if $T \vdash_{\mathbf{L}} c$.

Theorem 1: 2. if $T:T' \vdash_{\mathbf{L}^*} c^*:\gamma$, then $T:T' \vdash_{\mathbf{L}} c:\gamma$.

3. if $T:T' \vdash_{\mathbf{L}} c:\gamma$, there is a γ' such that $\gamma' = \gamma$ and $T:T' \vdash_{\mathbf{L}^*} c^*:\gamma'$.

Proof: We have seen that if $T:T' \vdash_{\mathbf{L}} c:\gamma$, then there is a *Cut*-free $\mathbf{L}$ proof π of $T:T' \vdash c:\gamma'$ where $\gamma' = \gamma$ and π has the following properties: (a) all axiom instances in π are atomic; (b) no major premise of a $\backslash L$ or $/L$ inference in π is the conclusion of a $\backslash R$ or $/R$ inference; and (c) the same category remains active whenever one goes down from axioms via major premises of $\backslash L$ and $/L$ inferences in π . But this suffices, for there is an $\mathbf{L}^*$ proof π' of $T:T' \vdash c^*:\gamma'$ iff there is a *Cut*-free $\mathbf{L}$ proof π of $T:T' \vdash c:\gamma'$ with (a), (b) and (c):

Note (a') that $\mathbf{L}^*$ axioms $at^* \vdash at$ involve only atomic categories; (b') that the major premise of a $\backslash L$ or $/L$ inference in $\mathbf{L}^*$ can only be an axiom or the conclusion of another $\backslash L$ or $/L$ inference (the asterisk must be on the left-hand side); and (c') that if a $\backslash L$ or $/L$ inference yields the major premise of another $\backslash L$ or $/L$ inference, then they have the same (asterisked) active category. On account of (a') through (c'), every *Cut*-free $\mathbf{L}$ proof π of $T:T' \vdash c:\gamma'$ having the properties (a) through (c) can be turned into an $\mathbf{L}^*$ proof π' of $T:T' \vdash c^*:\gamma'$ by (i) adding an asterisk to the left-hand side category of every axiom instance; (ii) adding an asterisk to the active category in every conclusion sequent of a $\backslash L$, $/L$, $\backslash R$ and $/R$ inference; and (iii) replacing every sequent of the form $U:U', a^*:u, V:V' \vdash at:\gamma$ which is not the major premise of a $\backslash L$ or $/L$ inference by the inference given in (23):

$$(23) \quad \frac{U:U', a^*:u, V:V' \vdash at:\gamma}{U:U', a:u, V:V' \vdash at^*:\gamma} [*]$$

And, conversely, every $\mathbf{L}^*$ proof π' of $T:T' \vdash c^*:\gamma'$ can be turned into a *Cut*-free $\mathbf{L}$ proof π

of $T:T' \vdash c:\gamma'$ with properties (a) through (c) by (i) replacing every inference of the form (23) by the sequent $U:U', a:u, V:V' \vdash at:\gamma$; and (ii) deleting all remaining asterisks. $\square$

4 Spurious Ambiguity

In the present section we will show that $\mathbf{L}^*$ is a solution to the spurious ambiguity problem:

Theorem 2: If π_1 and π_2 are $\mathbf{L}^*$ proofs of $T:T' \vdash c:\gamma$ and $T:T' \vdash c:\gamma'$ such that $\gamma = \gamma'$, then $\pi_1 = \pi_2$.

Proof: Suppose π_1 and π_2 are two different proofs of the same conclusion $T \vdash c$. Since axioms and non-axiom theorems in $\mathbf{L}^*$ never have the same degree, no sequent can be both an axiom and a non-axiom theorem. So, there must be at least one non-axiom sequent $T^\circ \vdash c^\circ$ which is derived from different premises in π_1 and π_2 . Inspection of $\mathbf{L}^*$ shows that the goal c° of this sequent cannot be of the form $b \backslash a^*$ or a/b^* , for the rules $\backslash R$ and $/R$ which yield conclusions with such goals have the property that their conclusion sequents, $T \vdash a \backslash b^*$ and $T \vdash a/b^*$, completely determine the form of their premise sequents: $b, T \vdash a^*$ and $T, b \vdash a^*$, respectively. So, c° is (i) an asterisked or (ii) a non-asterisked atomic category.

(i) If $c^\circ = at^*$, then $T^\circ \vdash c^\circ$ must have been obtained with the $*$ rule in both π_1 and π_2 . So, $T^\circ = U, a, V, b, W$ and the situation is as follows in π_1 and π_2 :

$$(24) \quad \frac{U:U', a^*:u, V:V', b:v, W:W' \vdash at:\gamma}{U:U', a:u, V:V', b:v, W:W' \vdash at^*:\gamma} [*] \quad \text{and} \quad \frac{U:U', a:u, V:V', b^*:v, W:W' \vdash at:\gamma'}{U:U', a:u, V:V', b:v, W:W' \vdash at^*:\gamma'} [*]$$

(ii) If $c^\circ = at$, then the left-hand side T' must contain one asterisked category d^* . Because $T^\circ \vdash c^\circ$ is not an axiom, d^* must be a non-atomic asterisked category a/b^* (or $b \backslash a^*$), and accordingly, $T^\circ \vdash c^\circ$ must have been obtained using $/L$ (or $\backslash L$) in both π_1 and π_2 . We only consider $/L$, since the $\backslash L$ case is analogous. So, d^* equals a/b^* and the situation must be as follows in π_1 and π_2 , where V is a non-empty sequence of categories:

$$(25) \quad \frac{U:U' \vdash b^*:\beta \quad S:S', a^*:u, V:V', W:W' \vdash at:\gamma}{S:S', a/b^*:w, U:U', V:V', W:W' \vdash at:\gamma[u:=w(\beta)]} [/L] \quad \text{and} \quad \frac{U:U', V:V' \vdash b^*:\beta' \quad S:S', a^*:u, W:W' \vdash at:\gamma'}{S:S', a/b^*:w, U:U', V:V', W:W' \vdash at:\gamma'[u:=w(\beta')]} [/L]$$

It turns out that the terms γ and γ' assigned in (24) (see **Fact 1** below) and the terms $\gamma[u:=w(\beta)]$ and $\gamma'[u:=w(\beta')]$ assigned in (25) (**Fact 2**) are non-equivalent. Consequently, $T^\circ \vdash c^\circ$ gets non-equivalent interpretations in π_1 and π_2 . Finally, it can be shown that any non-equivalence of the terms which different proofs assign to some intermediate sequent is inherited by the terms assigned to the conclusions drawn from that sequent (**Fact 3**). So, if $T^\circ \vdash c^\circ$ has non-equivalent interpretations in π_1 and π_2 , then $T \vdash c$, the final conclusion of both proofs, will get non-equivalent interpretations in π_1 and π_2 as well. $\square$

In footnote 2 it was observed that terms γ assigned to conclusion goals in *Cut*-free $\mathbf{L}$ proofs are always in β -normal form, i.e., they do not have subterms of the form $[\lambda v. \gamma'](\gamma'')$. In the proof of **Theorem 1** we have seen that the terms assigned to conclusion goals of $\mathbf{L}^*$ proofs constitute a subset of the conclusion goal terms assigned in *Cut*-free $\mathbf{L}$ proofs and, hence, are in β -normal form as well. But they also meet the stronger requirement of being in what we will call ' $\beta\bar{\eta}$ -normal form' (as can be seen by an induction on the length of $\mathbf{L}^*$ proofs):

- A term γ is in $\bar{\eta}$ -normal form ($\bar{\eta}$ -nf) iff every occurrence of a subterm $\gamma' \not\equiv \lambda v. \gamma''$ of functional type (a, b) is applied to some term γ''' in γ .⁹
- A term γ is in $\beta\bar{\eta}$ -nf iff γ is in β -nf and γ is in $\bar{\eta}$ -nf.

As was noted in footnote 3, equivalence in typed λ -calculus amounts to identity of $\beta\eta$ -nfs, i.e., two terms γ and δ are equivalent ($\gamma = \delta$) iff their $\beta\eta$ -nfs $\gamma^{\beta\eta}$ and $\delta^{\beta\eta}$ are syntactically identical ($\gamma^{\beta\eta} \equiv \delta^{\beta\eta}$), modulo renaming bound variables. Together with the fact that $\gamma^{\beta\eta} \equiv \delta^{\beta\eta}$ iff $\gamma^{\beta\bar{\eta}} \equiv \delta^{\beta\bar{\eta}}$ (see footnote 9), this yields that $\gamma = \delta$ iff $\gamma^{\beta\bar{\eta}} \equiv \delta^{\beta\bar{\eta}}$. Terms assigned to conclusion goals of $\mathbf{L}^*$ proofs are in $\beta\bar{\eta}$ -nf, so they are equivalent iff they are identical, a fact which will be exploited in the corollaries of the following lemma. Let the function $\#$ count the number of arguments of $c \in \text{CAT}$: $\#(at) = 0$ and $\#(a/b) = \#(b \setminus a) = \#(a) + 1$.

Lemma: If $U:U', c^*:w, V:V' \vdash at : \gamma$, then for $n = \#(c)$: $\gamma \equiv w(\gamma_1) \dots (\gamma_n)$.

Proof: by induction on the degree of c . If $c \in \text{ATOM}$, then $\#(c) = 0$ and $U, c^*, V \vdash at$ is an axiom, so $\gamma \equiv w$. If c is a/b (or $b \setminus a$), then $U, c^*, V \vdash at$ is the conclusion of $\setminus L$ (or $/L$):

$$\frac{V_1:V'_1 \vdash b:\beta^* \quad U:U', a^*:u, V_2:V'_2 \vdash at:\gamma}{U:U', a/b^*:w, V_1:V'_1, V_2:V'_2 \vdash at:\gamma[u:=w(\beta)]} [L]$$

Apply the induction hypothesis to γ : $\gamma \equiv u(\gamma_1) \dots (\gamma_n)$. Now, $\#(a/b) = \#(a) + 1$ and as u occurs only once in γ , the term $\gamma[u:=w(\beta)]$ is identical to $w(\beta)(\gamma_1) \dots (\gamma_n)$. $\square$ **Fact 1:** Consider (24): $u \neq v$, so $u(\gamma_1) \dots (\gamma_n) \equiv \gamma \neq \gamma' \equiv v(\gamma'_1) \dots (\gamma'_m)$, so $\gamma \neq \gamma'$.

Fact 2: Consider (25): $\gamma \equiv u(\gamma_1) \dots (\gamma_{\#(a)})$ and $\gamma' \equiv u(\gamma'_1) \dots (\gamma'_{\#(a)})$. Hence $\gamma[u:=w(\beta)] \equiv w(\beta)(\gamma_1) \dots (\gamma_{\#(a)})$ and $\gamma'[u:=w(\beta')] \equiv w(\beta')(\gamma'_1) \dots (\gamma'_{\#(a)})$. Since the sequence of categories V is non-empty, it contains a category c , which is associated with a variable v which – considering $\gamma[u:=w(\beta)]$ – occurs freely in some γ_i ($1 \leq i \leq \#(a)$) but not in β , and which – considering $\gamma'[u:=w(\beta')]$ – occurs freely in β' but not in any γ'_j ($1 \leq j \leq \#(a)$). That is, $\gamma[u:=w(\beta)] \neq \gamma'[u:=w(\beta')]$ and, consequently, $\gamma[u:=w(\beta)] \neq \gamma'[u:=w(\beta')]$.

Fact 3: Consider the following two inferences:

$$\frac{T:T' \vdash b^*:\beta \quad U:U', a^*:u, V:V' \vdash at:\gamma}{U:U', a/b^*:w, T:T', V:V' \vdash at:\gamma[u:=w(\beta)]} [L] \quad \text{and} \quad \frac{T:T' \vdash b^*:\beta' \quad U:U', a^*:u, V:V' \vdash at:\gamma'}{U:U', a/b^*:w, T:T', V:V' \vdash at:\gamma'[u:=w(\beta')]} [L]$$

Suppose that $\beta \neq \beta'$ or $\gamma \neq \gamma'$. Then $\beta \neq \beta'$ or $\gamma \neq \gamma'$. Since $\gamma = u(\gamma_1) \dots (\gamma_n)$ and $\gamma' = u(\gamma'_1) \dots (\gamma'_n)$ ($n = \#(a/b)$), it must hold that $\gamma[u:=w(\beta)] \equiv w(\beta)(\gamma_1) \dots (\gamma_n) \neq w(\beta')(\gamma'_1) \dots (\gamma'_n) \equiv \gamma'[u:=w(\beta')]$, for we supposed that $\beta \neq \beta'$ or $\gamma \neq \gamma'$, and the latter means that for some i , $1 \leq i \leq n$, $\gamma_i \neq \gamma'_i$. Hence also $\gamma[u:=w(\beta)] \neq \gamma'[u:=w(\beta')]$. Furthermore, note that (i) the $\setminus L$ case is analogous; (ii) the semantics of the $*$ rule is identity; and (iii) as to the rules $\setminus R$ and $/R$: if $\alpha \neq \alpha'$, i.e., $\alpha \neq \alpha'$, then $\lambda v. \alpha \neq \lambda v. \alpha'$, i.e., $\lambda v. \alpha \neq \lambda v. \alpha'$. Therefore, non-equivalence of terms assigned to some sequent is always inherited by the terms assigned to the conclusions drawn from that sequent.

Returning to our two examples, it appears that the unambiguous sequent $c/c, c/c, c \vdash c^*$ is indeed derived only once in $\mathbf{L}^*$, whereas it has two *Cut*-free $\mathbf{L}$ derivations. Modulo $*$ inferences, derivation (26) has the structure of (9). There is no $\mathbf{L}^*$ proof with the structure of (10), since the main mmp of (10) contains two different active left-hand side categories.

$$(26) \quad \frac{\frac{\frac{c^* \vdash c [*]}{c \vdash c^*} \quad c^* \vdash c}{c/c^*, c \vdash c} [L] \quad c^* \vdash c}{c/c, c \vdash c^*} [L] \quad \frac{c/c^*, c/c, c \vdash c}{c/c, c/c, c \vdash c^*} [L]$$

The two-way ambiguous sequent $s/(n \setminus s), (n \setminus s)/n, (s/n) \setminus s \vdash s$ has six *Cut*-free $\mathbf{L}$ deriva-

tions, but is derived twice in L^* :

$$(27) \frac{\frac{n^* \vdash n \quad [*]}{n \vdash n^*} \quad \frac{n \vdash n^* \quad s^* \vdash s \quad [\backslash L]}{n, n \backslash s^* \vdash s} \quad [\backslash L]}{n, (n \backslash s)/n^*, n \vdash s} \quad [\backslash L]$$

$$\frac{\frac{n, (n \backslash s)/n, n \vdash s^* \quad [*]}{n, (n \backslash s)/n, n \vdash s^*} \quad [\backslash R]}{n, (n \backslash s)/n \vdash s/n^*} \quad [\backslash R] \quad \frac{s^* \vdash s \quad [\backslash L]}{n, (n \backslash s)/n, (s/n) \backslash s^* \vdash s} \quad [\backslash L]$$

$$\frac{n, (n \backslash s)/n, (s/n) \backslash s^* \vdash s^* \quad [*]}{n, (n \backslash s)/n, (s/n) \backslash s^* \vdash s^*} \quad [\backslash R] \quad \frac{s^* \vdash s \quad [\backslash L]}{(n \backslash s)/n, (s/n) \backslash s^* \vdash n \backslash s^*} \quad [\backslash R]$$

$$\frac{(n \backslash s)/n, (s/n) \backslash s^* \vdash n \backslash s^* \quad s/(n \backslash s)^*, (n \backslash s)/n, (s/n) \backslash s \vdash s \quad [*]}{s/(n \backslash s), (n \backslash s)/n, (s/n) \backslash s \vdash s^*} \quad [\backslash L]$$

$$(28) \frac{\frac{n^* \vdash n \quad [*]}{n \vdash n^*} \quad \frac{n \vdash n^* \quad s^* \vdash s \quad [\backslash L]}{n, n \backslash s^* \vdash s} \quad [\backslash L]}{n, (n \backslash s)/n^*, n \vdash s} \quad [\backslash L]$$

$$\frac{\frac{n, (n \backslash s)/n, n \vdash s^* \quad [*]}{n, (n \backslash s)/n, n \vdash s^*} \quad [\backslash R]}{(n \backslash s)/n, n \vdash n \backslash s^*} \quad [\backslash R] \quad \frac{s^* \vdash s \quad [\backslash L]}{s/(n \backslash s)^*, (n \backslash s)/n, n \vdash s} \quad [\backslash L]$$

$$\frac{s/(n \backslash s), (n \backslash s)/n, n \vdash s^* \quad [*]}{s/(n \backslash s), (n \backslash s)/n, n \vdash s^*} \quad [\backslash R] \quad \frac{s^* \vdash s \quad [\backslash L]}{s/(n \backslash s), (n \backslash s)/n \vdash s/n^*} \quad [\backslash R]$$

$$\frac{s/(n \backslash s), (n \backslash s)/n \vdash s/n^* \quad s/(n \backslash s), (n \backslash s)/n, (s/n) \backslash s^* \vdash s \quad [*]}{s/(n \backslash s), (n \backslash s)/n, (s/n) \backslash s \vdash s^*} \quad [\backslash L]$$

These derivations are structurally similar and semantically equivalent to (11) and (14), respectively. The remaining *Cut*-free L derivations of the sequent do not have L^* counterparts: (15) and (16) because of a non-atomic axiom instance $n \backslash s \vdash n \backslash s$; (13) due to a $\backslash R$ inference which yields the major premise of a $/L$ inference; and (12) on account of an mmp with two different active left-hand side categories.

5 Partial Deduction and Proof Nets

In the present section we will discuss two alternative approaches to the problem of spurious ambiguity which were proposed in Moortgat (1990) and Roorda (1991).

Moortgat (1990) suggests that *partial deduction* offers a way of tackling the problem of spurious ambiguity for L . The basic idea of partial deduction is that theorem proving is preceded by a compilation of the *lexicon*, i.e., the set of categories which occur in the left-hand side of the sequents to be derived and interpreted. This set of categories serves as the basis for the generation of axioms and rules which are to replace the original axioms and rules of L .¹⁰ The idea is that since they have more internal structure than the rules of L , the new rules restrict the derivational freedom of the original system, thus eliminating spurious ambiguity. Unfortunately, Moortgat (1990) does not contain a formal definition of partial deduction, let alone a proof that the method is sound (i.e., preserves all sequents and interpretations) and complete (i.e., eliminates all spurious ambiguity). The paper only treats two examples: the categories $(a_1 \backslash a_2)/a_3$ and $(s/(n \backslash s)) \backslash s$ (where a_1, a_2, a_3, n and s are atomic). The first order¹¹ category $(a_1 \backslash a_2)/a_3$ gives rise to a single inference rule:

$$(29) \frac{T_2 : T'_2 \vdash a_3 : \gamma_2 \quad T_1 : T'_1 \vdash a_1 : \gamma_1}{T_1 : T'_1, (a_1 \backslash a_2)/a_3 : v_0, T_2 : T'_2 \vdash a_2 : v_0(\gamma_2)(\gamma_1)} \quad [(a_1 \backslash a_2)/a_3]$$

The higher order category $(s/(n \backslash s)) \backslash s$, on the other hand, results in a *set* consisting of two derived inference rules. One corresponds to the category $(s/(n \backslash s)) \backslash s$ itself, and the other to the (only) argument $n \backslash s$ of the (only) argument $s/(n \backslash s)$ of $(s/(n \backslash s)) \backslash s$.¹²

$$(30) \frac{T_1 : T'_1, n \backslash s : v_1 \vdash s : \gamma_1}{T_1 : T'_1, (s/(n \backslash s)) \backslash s : v_0 \vdash s : v_0(\lambda v_1. \gamma_1)} \quad [(s/(n \backslash s)) \backslash s] \quad \frac{T_1 : T'_1 \vdash n : \gamma_1}{T_1 : T'_1, n \backslash s : v_0 \vdash s : v_0(\gamma_1)} \quad [n \backslash s]$$

We will show that a formal definition can be given which is consistent with the examples treated by Moortgat and indeed solves the problem of spurious ambiguity. First, we define two functions from categories to sets of categories. For $c \in \text{CAT}$, $A(c)$ is the set consisting of the arguments of c and $A^A(c)$ is the set consisting of the arguments of the arguments of c : $A(at) = A^A(at) = \emptyset$; $A(a/b) = A(b/a) = \{b\} \cup A(a)$; $A^A(a/b) = A^A(b/a) = A(b) \cup A^A(a)$.

If C is a set of categories, then C^+ is the set which includes C and contains any argument of (an argument of an argument of) ^{n} an argument of a category $c \in C$ (where $n \in \mathbb{N}$):

Let $C \subseteq \text{CAT}$. C^+ is the smallest set such that $C \subseteq C^+$ and for all $c \in C^+$: $A^A(c) \subseteq C^+$.

For $(a_1 \setminus a_2)/a_3$ and $(s/(n \setminus s)) \setminus s$, this yields: $A((a_1 \setminus a_2)/a_3) = \{a_1, a_3\}$; $A^A((a_1 \setminus a_2)/a_3) = \emptyset$. So $\{(a_1 \setminus a_2)/a_3\}^+ = \{(a_1 \setminus a_2)/a_3\}$. And $A((s/(n \setminus s)) \setminus s) = \{s/(n \setminus s)\}$; $A^A((s/(n \setminus s)) \setminus s) = \{n \setminus s\}$; $A(n \setminus s) = \{n\}$, so $A^A(n \setminus s) = \emptyset$ and $\{(s/(n \setminus s)) \setminus s\}^+ = \{(s/(n \setminus s)) \setminus s, n \setminus s\}$. Next, let f be the following function from interpreted sequents to interpreted sequents:¹³

$f(T:T' \vdash at : \gamma) = T:T' \vdash at : \gamma$; $f(T:T' \vdash a/b : \lambda v. \gamma) = f(T:T', b:v \vdash a : \gamma)$; and $f(T:T' \vdash b \setminus a : \lambda v. \gamma) = f(b:v, T:T' \vdash a : \gamma)$.

The function g assigns an axiom to every atomic category and a derived inference rule to every compound category (recall that $\#$ counts the number of arguments of $c \in \text{CAT}$):

$$g(at) = at : v_0 \vdash at : v_0;$$

$$g(a/b) = \frac{f(T_{m+1}:T'_{m+1} \vdash b : \lambda v_1 \dots \lambda v_{\#(b)}. \gamma_{m+1}) \quad \sigma_m \quad \dots \quad \sigma_1}{\mathbf{X}, a/b : v_0, T_{m+1}:T'_{m+1}, \mathbf{Y} \vdash at : v_0(\lambda v_1 \dots \lambda v_{\#(b)}. \gamma_{m+1})(\delta_m) \dots (\delta_1)} \text{ and}$$

$$g(b \setminus a) = \frac{f(T_{m+1}:T'_{m+1} \vdash b : \lambda v_1 \dots \lambda v_{\#(b)}. \gamma_{m+1}) \quad \sigma_m \quad \dots \quad \sigma_1}{\mathbf{X}, T_{m+1}:T'_{m+1}, b \setminus a : v_0, \mathbf{Y} \vdash at : v_0(\lambda v_1 \dots \lambda v_{\#(b)}. \gamma_{m+1})(\delta_m) \dots (\delta_1)} \text{ iff } g(a) \text{ has}$$

premises $\sigma_m, \dots, \sigma_1$ and conclusion $\mathbf{X}, a : v_0, \mathbf{Y} \vdash at : v_0(\delta_m) \dots (\delta_1)$. (Note that $m = \#(a)$.)

The expressions T_i will represent non-empty finite sequences of categories. As before, we will assume that in the context of a proof all variables v_0 assigned to an axiom instance or introduced in the conclusion of a derived inference rule are different.

For example: $\#(a_1) = \#(a_3) = 0$, so $f(T_i:T'_i \vdash a_1 : \lambda v_1 \dots \lambda v_{\#(a_1)}. \gamma_i) = T_i:T'_i \vdash a_1 : \gamma_i$ and $f(T_i:T'_i \vdash a_3 : \lambda v_1 \dots \lambda v_{\#(a_3)}. \gamma_i) = T_i:T'_i \vdash a_3 : \gamma_i$. Besides, $g(a_2) = a_2 : v_0 \vdash a_2 : v_0$, so:

$$g(a_1 \setminus a_2) = \frac{T_1:T'_1 \vdash a_1 : \gamma_1}{T_1, a_1 \setminus a_2 : v_0 \vdash a_2 : v_0(\gamma_1)}, \text{ and}$$

$$g((a_1 \setminus a_2)/a_3) = \frac{T_2:T'_2 \vdash a_3 : \gamma_2 \quad T_1:T'_1 \vdash a_1 : \gamma_1}{T_1:T'_1, (a_1 \setminus a_2)/a_3, T_2:T'_2 \vdash a_3 : v_0(\gamma_2)(\gamma_1)}.$$

Note that $g((a_1 \setminus a_2)/a_3)$ is the same rule as (29).

$\#(s/(n \setminus s)) = 1$, so $f(T_i:T'_i \vdash s/(n \setminus s) : \lambda v_1 \dots \lambda v_{\#(s/(n \setminus s))}. \gamma_i) = T_i:T'_i, n \setminus s : v_1 \vdash s : \lambda v_1. \gamma_i$. Hence, and because $g(s) = s : v_0 \vdash s : v_0$:

$$g((s/(n \setminus s)) \setminus s) = \frac{T_1:T'_1, n \setminus s : v_1 \vdash s : \gamma_1}{T_1:T'_1, (s/(n \setminus s)) \setminus s : v_0 \vdash s : v_0(\lambda v_1. \gamma_1)}.$$

$\#(n) = 0$, so $f(T_i:T'_i \vdash n : \lambda v_1 \dots \lambda v_{\#(n)}. \gamma_i) = T_i:T'_i \vdash n : \gamma_i$. And $g(s) = s : v_0 \vdash s : v_0$, so:

$$g(n \setminus s) = \frac{T_1:T'_1 \vdash n : \gamma_1}{T_1:T'_1, n \setminus s : v_0 \vdash s : v_0(\gamma_1)}.$$

The rules $g((s/(n \setminus s)) \setminus s)$ and $g(n \setminus s)$ are identical to the ones in (30). Thus, our formal definition is consistent with the examples of Moortgat (1990).

Note that all **PD** theorems have atomic goal categories. Call $c_1 : v_1, \dots, c_n : v_n \vdash at : \gamma$ a **PD** ('Partial Deduction') theorem if and only if it is derivable using the axioms and rules in $\{g(c) \mid c \in \{c_1, \dots, c_n\}^+\}$. It can be shown that there exists a (total) function h from **L*** proofs of sequents $T:T' \vdash at^* : \gamma$ with asterisked atomic goal categories to **PD** proofs of $T:T' \vdash at : \gamma$. On account of **Theorem 1** (see section 3 above), this entails that partial deduction preserves all (interpretations of) **L**-derivable sequents with atomic goal categories. Moreover, the function h is a surjection. On account of **Theorem 2** (see section 4 above), this entails that partial deduction is devoid of spurious ambiguity, in that it derives every

interpretation only once. The function h is defined as follows. Let $I_1, \dots, I_{\#(b)} \in \{\backslash R, / R\}$:

$$\begin{aligned}
(a) \quad & h\left(\frac{at^*:w \vdash at:w}{at:w \vdash at^*:w} [*]\right) = at:w \vdash at:w; \\
(b) \quad & h\left(\frac{\frac{\pi'}{[I_1]} \dots [I_{\#(b)}]}{T:T' \vdash b^*: \beta} \frac{\pi''}{[L]}\right) = \\
& \frac{U:U', a/b^*:w, T:T', V:V' \vdash at:\gamma[v:=w(\beta)]}{U:U', a/b:w, T:T', V:V' \vdash at^*:\gamma[v:=w(\beta)]} \pi'' [L] = \\
& \frac{h(\pi')}{U:U', a/b:w, T:T', V:V' \vdash at:\gamma[v:=w(\beta)]} \pi_n \dots \pi_1 [g(a/b)] \text{ iff} \\
& h\left(\frac{\pi''}{U:U', a:v, V:V' \vdash at^*:\gamma} [*]\right) = \frac{\pi_n \dots \pi_1}{U:U', a:v, V:V' \vdash at:\gamma} [g(a)]; \text{ and} \\
(c) \quad & h\left(\frac{\frac{\pi'}{[I_1]} \dots [I_{\#(b)}]}{T:T' \vdash b^*: \beta} \frac{\pi''}{[L]}\right) = \\
& \frac{U:U', T:T', b \backslash a^*:w, V:V' \vdash at:\gamma[v:=w(\beta)]}{U:U', T:T', b \backslash a:w, V:V' \vdash at^*:\gamma[v:=w(\beta)]} \pi'' [L] = \\
& \frac{h(\pi')}{U:U', T:T', b \backslash a:w, V:V' \vdash at:\gamma[v:=w(\beta)]} \pi_n \dots \pi_1 [g(b \backslash a)] \text{ iff} \\
& h\left(\frac{\pi''}{U:U', a:v, V:V' \vdash at^*:\gamma} [*]\right) = \frac{\pi_n \dots \pi_1}{U:U', a:v, V:V' \vdash at:\gamma} [g(a)].
\end{aligned}$$

That the function h has the properties that

- for every L^* proof π of a sequent $Z:Z' \vdash at^*:\delta$, $h(\pi)$ is a **PD** proof of $Z:Z' \vdash at:\delta$
 - for every **PD** proof π , there is an L^* proof π^* such that $h(\pi^*) = \pi$
- is shown by two rather intricate inductions on the degree of the conclusion of π . The interested reader is referred to Chapter 4 of Hendriks (1992).

Roorda (1991) introduces so-called *proof nets* (invented for linear logic by Girard (1987)) for the Lambek calculus. Basically, a proof net is a structure of linked signed categories. For the product-free Lambek calculus, the following links are distinguished:

$$\begin{array}{cccccc}
\frac{\bar{a} \quad \bar{t}}{at} \quad \frac{+ \quad +}{at} & \frac{\bar{a} \quad \bar{t}}{at} \quad \frac{+ \quad +}{at} & \frac{\bar{a} \quad \bar{b}}{b \backslash a} \backslash 1 & \frac{\bar{b} \quad \bar{a}}{a/b} / 1 & \frac{+ \quad \bar{a}}{b \backslash a} \backslash 2 & \frac{\bar{a} \quad +}{a/b} / 2
\end{array}$$

Let X, Y, Z denote sequences of signed categories, and $tc(A)$ the sequence of *terminal* (signed) categories of proof net A . The set of proof nets PN is the smallest set such that

- $\frac{\bar{a} \quad \bar{t}}{at} \in PN$ and $\frac{+ \quad +}{at} \in PN$; $tc(\frac{\bar{a} \quad \bar{t}}{at}) = \bar{a}, \bar{t}$ and $tc(\frac{+ \quad +}{at}) = +, +$;
- if $A \in PN$ and $tc(A) = X, \bar{a}, \bar{b}, Y$ and X or Y is non-empty, then $C \in PN$, where C results from applying $\backslash 1$ to $\bar{a}$ and $\bar{b}$; $tc(C) = X, b \backslash a, Y$;
- if $A \in PN$ and $tc(A) = X, \bar{b}, \bar{a}, Y$ and X or Y is non-empty, then $C \in PN$, where C results from applying $/ 1$ to $\bar{b}$ and $\bar{a}$; $tc(C) = X, a/b, Y$;
- if $A \in PN$ and $tc(A) = X, \bar{a}, Y$ and $B \in PN$ and $tc(B) = Z, \bar{b}$, then $C \in PN$, where C results from applying $\backslash 2$ to $\bar{b}$ and $\bar{a}$; $tc(C) = X, Z, b \backslash a, Y$;

- (e) if $A \in \text{PN}$ and $tc(A) = X, \bar{b}^+, Y$ and $B \in \text{PN}$ and $tc(B) = \bar{a}, Z$, then $C \in \text{PN}$, where C results from applying $\setminus 2$ to $\bar{b}^+$ and $\bar{a}$; $tc(C) = X, \bar{b}^+ \setminus \bar{a}, Z, Y$;
- (f) if $A \in \text{PN}$ and $tc(A) = X, \bar{a}, Y$ and $B \in \text{PN}$ and $tc(B) = \bar{b}^+, Z$, then $C \in \text{PN}$, where C results from applying $/2$ to $\bar{a}$ and $\bar{b}^+$; $tc(C) = X, \bar{a} / \bar{b}^+, Z, Y$; and
- (g) if $A \in \text{PN}$ and $tc(A) = X, \bar{b}^+, Y$ and $B \in \text{PN}$ and $tc(B) = Z, \bar{a}$, then $C \in \text{PN}$, where C results from applying $/2$ to $\bar{a}$ and $\bar{b}^+$; $tc(C) = X, Z, \bar{a} / \bar{b}^+, Y$.

Due to (d), (e), (f) and (g), two separate proof nets can be combined into one; application of $\setminus 2$ and $/2$ has the effect of *inserting* proof net B into A .¹⁴ The above definition is equivalent to one in which *two* sets, PN^1 and PN^2 , are defined as the *smallest* sets observing (a') through (h'), where clause (b') and (c') are obtained from (b) and (c) by replacing 'PN' by ' PN^2 '; clause (d') and (f') are obtained from (d) and (f) by replacing ' $A \in \text{PN}$ ', ' $B \in \text{PN}$ ' and ' $C \in \text{PN}$ ' by ' $A \in \text{PN}^1$ ', ' $B \in \text{PN}^2$ ' and ' $C \in \text{PN}^1$ ', respectively; clause (e') and (g') are obtained from (e) and (g) by replacing ' $A \in \text{PN}$ ', ' $B \in \text{PN}$ ' and ' $C \in \text{PN}$ ' by ' $A \in \text{PN}^2$ ', ' $B \in \text{PN}^1$ ' and ' $C \in \text{PN}^1$ ', respectively; and clause (a') and (h') read as follows:

- (a') $\bar{a}t \quad \bar{a}t \in \text{PN}^1$ and $\bar{a}t \quad \bar{a}t \in \text{PN}^1$; $tc(\bar{a}t \quad \bar{a}t) = \bar{a}t, \bar{a}t$ and $tc(\bar{a}t \quad \bar{a}t) = \bar{a}t, \bar{a}t$;
- (h') if $A \in \text{PN}^1$ and $tc(A) = X, \bar{a}t, Y$, then $C \in \text{PN}^2$, where C results from applying ax to A and $tc(C) = X, \bar{a}t, Y$.

The equivalence of both definitions – i.e., $\text{PN} = \text{PN}^2$ – is shown by induction on the degree of $tc(C)$, employing that for $C \in \text{PN}$: (i) there is one $\bar{c}^+$ in $tc(C)$; and (ii) if $tc(C) = X, \bar{b}^+ \setminus \bar{a}, Y$ ($X, \bar{a} / \bar{b}^+, Y$), then for some $A \in \text{PN}$: C results from applying clause (b) (clause (c)) to A .

PN^1 consists of proof nets with one missing ax link. This alternative definition has the advantage of allowing an inductive definition of the semantics for proof nets. Let $C[v := \gamma]$ denote the result of replacing all occurrences of v in $C \in \text{PN}^1$ by γ ; let the sets of variables occurring in A and B be disjoint; and let w be a variable not occurring in A or B :

- (a'') $\bar{a}t : v \quad \bar{a}t : v$ and $\bar{a}t : v \quad \bar{a}t : v \in \text{PN}^1$; $tc(\bar{a}t : v \quad \bar{a}t : v) = \bar{a}t : v, \bar{a}t : v$ and $tc(\bar{a}t : v \quad \bar{a}t : v) = \bar{a}t : v, \bar{a}t : v$;
- (b'') if $A \in \text{PN}^2$ and $tc(A) = X, \bar{a} : v, \bar{b} : \alpha, Y$ and X or Y is non-empty, then $C \in \text{PN}^2$, where C results from applying $\setminus 1$ to $\bar{a} : v$ and $\bar{b} : \alpha$ in A ; $tc(C) = X, \bar{b} \setminus \bar{a} : \lambda v. \alpha, Y$;
- (c'') if $A \in \text{PN}^2$ and $tc(A) = X, \bar{b} : \alpha, \bar{a} : v, Y$ and X or Y is non-empty, then $C \in \text{PN}^2$, where C results from applying $/1$ to $\bar{b} : \alpha$ and $\bar{a} : v$ in A ; $tc(C) = X, \bar{a} / \bar{b} : \lambda v. \alpha, Y$;
- (d'') if $A \in \text{PN}^1$ and $tc(A) = X, \bar{a} : v, Y$ and $B \in \text{PN}^2$ and $tc(B) = Z, \bar{b} : \beta$, then $C \in \text{PN}^1$, where C results from applying $\setminus 2$ to $\bar{b} : \beta$ in B and $\bar{a} : w(\beta)$ in $A[v := w(\beta)]$; $tc(C) = X', Z, \bar{b} \setminus \bar{a} : w, Y'$ for $X', \bar{a} : w(\beta), Y' = tc(A[v := w(\beta)])$;
- (e'') if $A \in \text{PN}^2$ and $tc(A) = X, \bar{b} : \beta, Y$ and $B \in \text{PN}^1$ and $tc(B) = \bar{a} : v, Z$, then $C \in \text{PN}^1$, where C results from applying $\setminus 2$ to $\bar{b} : \beta$ in A and $\bar{a} : w(\beta)$ in $B[v := w(\beta)]$; $tc(C) = X, \bar{b} \setminus \bar{a} : w, Z', Y$ for $\bar{a} : w(\beta), Z' = tc(B[v := w(\beta)])$;
- (f'') if $A \in \text{PN}^1$ and $tc(A) = X, \bar{a} : v, Y$ and $B \in \text{PN}^2$ and $tc(B) = \bar{b} : \beta, Z$, then $C \in \text{PN}^1$, where C results from applying $/2$ to $\bar{a} : w(\beta)$ in $A[v := w(\beta)]$ and $\bar{b} : \beta$ in B ;

$tc(C) = X', \bar{a}/b:w, Z, Y'$ for $X', \bar{a}:w(\beta), Y' = tc(A[v:=w(\beta)])$;
 (g'') if $A \in \text{PN}^2$ and $tc(A) = X, \bar{b}^\dagger:\beta, Y$ and $B \in \text{PN}^1$ and $tc(B) = Z, \bar{a}:v$, then $C \in \text{PN}^1$,
 where C results from applying $/2$ to $\bar{a}:w(\beta)$ in $B[v:=w(\beta)]$ and $\bar{b}^\dagger:\beta$ in A ;
 $tc(C) = X, Z', \bar{a}/b:w, Y$ for $Z', \bar{a}:w(\beta) = tc(B[v:=w(\beta)])$;
 (h'') if $A \in \text{PN}^1$ and $tc(A) = X, \bar{a}^\dagger:\gamma, Y$, then $C \in \text{PN}^2$, where C results from applying ax
 to A ; $tc(C) = X, \bar{a}^\dagger:\gamma, Y$.

The proof nets C defined in (b'') , (c'') , (d'') , (e'') , (f'') , (g'') and (h'') will be called $\backslash 1(A)$, $/1(A)$, $\backslash 2(A, B^{w/v})$, $\backslash 2(A, B^{w/v})$, $/2(A^{w/v}, B)$, $/2(A, B^{w/v})$ and $ax(A)$, respectively. This semantics is equivalent to the interpretation procedure in Roorda (1991), which gives a non-inductive characterization of proof nets and constructs their λ -terms correspondingly. It is also closely related to $\mathbf{L}^*$ – via the function k from $\mathbf{L}^*$ proofs to the power set of $\text{PN}^1 \cup \text{PN}^2$:

- $k(at:v^* \vdash at:v) = \{\bar{a}^\dagger:v \ \bar{a}^\dagger:v, \bar{a}^\dagger:v \ \bar{a}^\dagger:v\}$;
- for $\pi[b:v, T:T' \vdash a^*:\alpha]$: $k(\frac{\pi}{T:T' \vdash b \backslash a^*:\lambda v.\alpha} [\backslash R]) = \{\backslash 1(A) \mid A \in k(\pi)\}$;
- for $\pi[T:T', b:v \vdash a^*:\alpha]$: $k(\frac{\pi}{T:T' \vdash a/b^*:\lambda v.\alpha} [/R]) = \{/1(A) \mid A \in k(\pi)\}$;
- for $\pi_1[T:T', \vdash b^*:\beta]$ and $\pi_2[U:U', a^*:v, V:V' \vdash at^*:\gamma]$:

$$k(\frac{\pi_1}{U:U', T:T', b \backslash a^*:w, V:V' \vdash at:\gamma[v:=w(\beta)]} \frac{\pi_2}{[\backslash L]}) =$$

$$\{\backslash 2(A^{w/v}, B) \mid A \in k(\pi_2) \ \& \ B \in k(\pi_1) \ \& \ tc(B) = \bar{T}:T', \bar{b}^\dagger:\beta\} \cup$$

$$\{\backslash 2(A, B^{w/v}) \mid A \in k(\pi_1) \ \& \ B \in k(\pi_2) \ \& \ tc(B) = \bar{a}:v, \bar{V}:V', \bar{a}^\dagger:\gamma, \bar{U}:U'\};$$
- for $\pi_1[T:T', \vdash b^*:\beta]$ and $\pi_2[U:U', a^*:v, V:V' \vdash at^*:\gamma]$:

$$k(\frac{\pi_1}{U:U', a/b^*:w, T:T', V:V' \vdash at:\gamma[v:=w(\beta)]} \frac{\pi_2}{[/L]}) =$$

$$\{/2(A^{w/v}, B) \mid A \in k(\pi_2) \ \& \ B \in k(\pi_1) \ \& \ tc(B) = \bar{b}^\dagger:\beta, \bar{T}:T'\} \cup$$

$$\{/2(A, B^{w/v}) \mid A \in k(\pi_1) \ \& \ B \in k(\pi_2) \ \& \ tc(B) = \bar{V}:V', \bar{a}^\dagger:\gamma, \bar{U}:U', \bar{a}:v\};$$
- for $\pi[U:U', a^*:v, V:V' \vdash at:\gamma]$: $k(\frac{\pi}{U:U', a:v, V:V' \vdash at^*:\gamma} [*]) = \{ax(A) \mid A \in k(\pi)\}$.

For all non-asterisked interpreted sequents $T:T \vdash c:\gamma$, we define $cp(T:T \vdash c:\gamma)$, the set of *cyclic permutations* of $T:T \vdash c:\gamma$, as $\{\bar{T}_2:T_2, \bar{c}^\dagger:\gamma, \bar{T}_1:T_1 \mid T:T = T_1:T_1, T_2:T_2\}$.

Note that $cp(c_1:v_1, \dots, c_n:v_n \vdash c:\gamma)$ contains $n+1$ members, and that:

- for $\pi[T:T' \vdash c^*:\gamma]$: $k(\pi) = \{A \in \text{PN}^2 \mid tc(A) \in cp(T:T' \vdash c:\gamma)\}$; and
- for $\pi[U:U, c^*:v, V:V' \vdash at:\gamma]$: $k(\pi) = \{A \in \text{PN}^1 \mid tc(A) \in cp(U:U, c:v, V:V' \vdash at:\gamma)\}$.

This is proven by induction on the length of π . So k assigns a subset of PN^2 to every $\mathbf{L}^*$ proof of a sequent with an asterisked goal and a subset of PN^1 to every $\mathbf{L}^*$ proof of a sequent with a non-asterisked goal. Moreover, the set $\{k(\pi) \mid \pi \text{ is an } \mathbf{L}^* \text{ proof}\}$ partitions $\text{PN}^1 \cup \text{PN}^2$: by induction on the number of links in A , we have that for $A \in \text{PN}^1 \cup \text{PN}^2$, there is exactly one π such that $A \in k(\pi)$. Since for every L-derivable interpreted sequent $c_1:v_1, \dots, c_n:v_n \vdash c:\gamma$ there is one $\mathbf{L}^*$ derivation of $c_1:v_1, \dots, c_n:v_n \vdash c^*:\gamma'$ such that $\gamma = \gamma'$, the above means that every reading of an L-derivable sequent $c_1, \dots, c_n \vdash c$

corresponds to $n + 1$ proof nets. Roorda (1991, p. 27) claims that ‘a proof net can be considered as a parallelized sequent proof and as such it lacks the spurious ambiguity of (even Cut-free) sequent proofs. A proof net is a concrete structure, not merely an abstract equivalence class of derivations [...]’. However, we have just seen that the set of proof nets has to be partitioned into equivalence classes if it is to serve as a solution to the spurious ambiguity problem: for every reading of an L-derivable sequent $c_1, \dots, c_n \vdash c$, there are $n + 1$ proof nets. These $n + 1$ proof nets are equivalent in that they make up a set of cyclic permutations. Nonetheless, one cannot isolate one specific representant – e.g., the proof net with its positive terminal category in rightmost position – from this set, since the definition of the notion ‘proof net’ requires *all* permutations (cf. especially clause (b) and (c)). And on the other hand, it is not possible to consider this set of permutations as a monolithic object, since sometimes a specific representant (viz., B in clause (d) through (g)) is isolated from this set. So, ironically, it is the proof net approach which has to resort to ‘abstract’ equivalence classes, whereas sequent normalization (and partial deduction) associate one concrete object with each interpretation of a derivable sequent: its L^* (PD) proof.

Notes

¹ Consider Moortgat’s treatment (1990, p. 151) of the crucial case 5 of Lambek’s *Cut* elimination algorithm (cf. section 2 below), in which:

$$\frac{\frac{1U, d : v \vdash b : \beta \quad [R]}{U \vdash b/d : \lambda v. \beta} \quad \frac{V \vdash d : \delta \quad 2T, b : [\lambda v. \beta](\delta), W \vdash c : \gamma \quad [L]}{V, b/d : \lambda v. \beta, V, W \vdash c : \gamma} \quad [Cut]}{T, U, V, W \vdash c : \gamma}$$

is replaced by:

$$\frac{1'U, d : \delta \vdash b : \beta[v := \delta] \quad 2'T, b : \beta[v := \delta], W \vdash c : \gamma[[\lambda v. \beta](\delta) := \beta[v := \delta]] \quad [Cut]}{V \vdash d : \delta \quad T, U, d : \delta, W \vdash c : \gamma[[\lambda v. \beta](\delta) := \beta[v := \delta]] \quad [Cut]}{T, U, V, W \vdash c : \gamma[[\lambda v. \beta](\delta) := \beta[v := \delta]]}$$

(The terms assigned to T, U, V and W have been omitted.) In order to be able to make the transition from the premises 1 and 2 to 1' and 2', respectively, Moortgat needs something like: If $\pi[U : U', a : \alpha, V : V' \vdash_{(5)} c : \gamma]$, then for all α' of type $\text{TYPE}(a)$: $\pi[U : U', a : \alpha', V : V' \vdash_{(5)} c : \gamma[\alpha := \alpha']]$. (As in (7), $\pi[c_1 : \gamma_1, \dots, c_n : \gamma_n \vdash_{(5)} c : \gamma]$ denotes the fact that proof π assigns interpretation $\gamma_1, \dots, \gamma_n \vdash \gamma$ to its conclusion sequent $c_1, \dots, c_n \vdash c$.) But such a lemma is not available, as can be seen from the following derivations – where α, α', β and γ have types $\text{TYPE}(a)$, $\text{TYPE}(a)$, $\text{TYPE}(a \setminus (b/a))$ and $(\text{TYPE}(a), \text{TYPE}(a))$:

$$\frac{\frac{a : \gamma(\alpha) \vdash a : \gamma(\alpha) \quad b : \beta(\alpha)(\gamma(\alpha)) \vdash b : \beta(\alpha)(\gamma(\alpha)) \quad [\setminus L]}{a : \gamma(\alpha), a \setminus b : \beta(\alpha) \vdash b : \beta(\alpha)(\gamma(\alpha)) \quad [L]} \quad \frac{a : \gamma(\alpha), (a \setminus b)/a : \beta, a : \alpha \vdash b : \beta(\alpha)(\gamma(\alpha))}{a : \gamma(\alpha) \vdash a : \alpha'} \quad \frac{a : \gamma(\alpha) \vdash a : \gamma(\alpha) \quad b : \beta(\alpha)(\gamma(\alpha)) \vdash b : \beta(\alpha)(\gamma(\alpha)) \quad [\setminus L]}{a : \gamma(\alpha), a \setminus b : \beta(\alpha) \vdash b : \beta(\alpha)(\gamma(\alpha)) \quad [L]} \quad \frac{a : \gamma(\alpha), (a \setminus b)/a : \beta, a : \alpha' \vdash b : \beta(\alpha')(\gamma(\alpha))}{a : \gamma(\alpha), (a \setminus b)/a : \beta, a : \alpha' \vdash b : \beta(\alpha')(\gamma(\alpha))}$$

Since no variable-binding operator is present, it seems plausible to assume that both occurrences of α in the term assigned to the conclusion goal of the former derivation are free, and that, hence, $[\beta(\alpha)(\gamma(\alpha))][\alpha := \alpha'] = \beta(\alpha')(\gamma(\alpha'))$. But the term assigned to the conclusion goal of the latter derivation is $\beta(\alpha')(\gamma(\alpha'))$. Note that the following FACT does hold in semantics (5): If $\pi[U : U', a : u, V : V' \vdash_{(5)} c : \gamma]$ and U', u, V' is a sequence of different variables, then for all terms α of type $\text{TYPE}(a)$: $\pi[U : U', a : \alpha', V : V' \vdash_{(5)} c : \gamma[u := \alpha']]$. It is proven by induction on the length of the derivation.

² That is, case 5 does not involve identity of terms, but equivalence under β -conversion (also called λ -conversion): $[\lambda v. \gamma'](\gamma'') = \gamma'[v := \gamma'']$, if γ'' is free for v in γ' . It takes an induction on the length of the derivation to see that terms γ assigned to conclusion goals in *Cut*-free proofs are always in β -normal form, i.e., they do not have subterms of the form $[\lambda v. \gamma'](\gamma'')$.

³ By η -conversion, the following equivalence is meant: $\lambda v. \gamma'(v) = \gamma'$, if v does not occur freely in γ' . A term γ is in η -normal form if it does not have subterms of the form $\lambda v. \gamma'(v)$. Equivalence in typed λ -calculus amounts to identity of $\beta\eta$ -normal forms in the following sense: for every term γ there is a unique term $\gamma^{\beta\eta}$ which is in $\beta\eta$ -normal form (i.e., it is both in β -normal form and in η -normal form) and which can be obtained from γ by performing a finite number of β -conversions and η -conversions (where renaming of bound variables is allowed); two terms γ and δ are equivalent iff their $\beta\eta$ -nfs $\gamma^{\beta\eta}$ and $\delta^{\beta\eta}$ are syntactically identical (modulo renaming bound variables).

⁴ The notions 'same' and 'different' active category are defined as follows: in a structure

$$\frac{T_2 \vdash b_2 \quad \frac{T_1 \vdash b_1 \quad U, a, V \vdash c}{T_3 \vdash c} [\$L]}{T_4 \vdash c} [\#L] \quad (\$ \text{ and } \# \in \{\backslash, /\})$$

the inferences $\$L$ and $\#L$ have the same active category iff $\$ = \backslash$, $\# = \backslash$, $T_3 = U, T_1, b_1 \backslash a, V$ and $T_4 = U, T_1, T_2, b_2 \backslash (b_1 \backslash a), V$; or $\$ = \backslash$, $\# = /$, $T_3 = U, T_1, b_1 \backslash a, V$ and $T_4 = U, T_1, (b_1 \backslash a) / b_2, T_2, V$; or $\$ = /$, $\# = \backslash$, $T_3 = U, a / b_1, T_1, V$ and $T_4 = U, T_2, b_2 \backslash (a / b_1), T_1, V$; or $\$ = /$, $\# = /$, $T_3 = U, a / b_1, T_1, V$ and $T_4 = U, (a / b_1) / b_2, T_2, T_1, V$; otherwise, they have different active categories.

⁵ Note (i) that transformation (21) cannot arise when the right-hand side premise of the first inference is an axiom, because the left-hand side of $U, c, Y, a, V \vdash e$ contains at least the two categories c and a ; and (ii) that if transformation (21) is executed in a configuration such as the one indicated below, where on the left-hand side $\@L$ and $\$L$ have the same active category and $\$L$ and $\#L$ have different active categories, then on the right-hand-side $\@L$ and $\#L$ (as well as $\#L$ and $\$L$) have different active categories:

$$\frac{\frac{T_3 \vdash b \quad \frac{T_1 \vdash a \quad T_2 \vdash d}{T_4 \vdash d} [\@L]}{T_5 \vdash c \quad T_6 \vdash d} [\#L]}{T_7 \vdash d} [\$L] \quad \sim (21) \quad \frac{T_3 \vdash b \quad \frac{T_1 \vdash a \quad T_2 \vdash d}{T_4 \vdash d} [\@L]}{T_5 \vdash c \quad T_6' \vdash d} [\#L]}{T_7 \vdash d} [\$L]$$

⁶ Consequently, every major premise of a $\backslash L$ or $/L$ inference must be an (atomic) axiom instance at $\vdash at$ or the conclusion of another $\backslash L$ or $/L$ inference. Since $\backslash L$ and $/L$ identify the goals of their major premise and conclusion, in π'' every $\backslash L$ and $/L$ inference must derive a conclusion sequent with an atomic goal: $T \vdash at$.

⁷ Hence σ_p must be the left-hand side premise of a $\backslash L$ or $/L$ inference, or the final conclusion of π .

⁸ I.e., the new left-hand side subproofs $\pi'_1, \dots, \pi'_{j-1}, \pi'_j, \pi'_{j+2}, \dots, \pi'_q$ of ν' still have a total of n mmps.

⁹ Let η -expansion (' η -conversion') be defined as the replacement of a subterm occurrence $\gamma' \neq \lambda v. \gamma''$ of functional type (a, b) which is not applied to any term γ''' by $\lambda v. \gamma'(v)$, where v does not occur freely in γ' . One can show that η -expansion is strongly normalizing and Church-Rosser: for every term γ there is a unique term γ^η which is in η -normal form, and which can be obtained from γ by performing a finite number of η -expansions (where renaming of bound variables is allowed). This is done by induction on the complexity of (the type of) γ , using the facts that (i) variables of non-functional (atomic) type are in η -nf; (ii) η -expansion of variables of type (a, b) only introduces variables of less complex type; and (iii) η -expansion of complex terms $\lambda v. \gamma'$ and $\gamma'(\gamma'')$ only involves η -expansion of less complex terms. As η -expansion respects β -normality, it holds for every γ and δ that $\gamma^{\beta\eta} \equiv \delta^{\beta\eta}$ iff $\gamma^{\beta\eta} \equiv \delta^{\beta\eta}$.

¹⁰ Moortgat says that 'partial deduction of the lexicon adds non-trivial axioms to the trivial axioms and logical rules' of L , such that the resulting system 'can do without the original logical rules' (1990, p. 388; small caps added, italics sic), but in the sequel we will see that the addition involves both axioms and rules.

¹¹ The order of a category c , $o(c)$, is defined as follows: $o(c) = 0$ if $c \in \text{ATOM}$; $o(a/b) = o(b \backslash a) = \max(o(a), o(b) + 1)$. E.g., $o(n) = 0$, $o(n \backslash s) = 1$, $o(s/(n \backslash s)) = 2$, but $o(s \backslash (n \backslash s)) = 1$.

¹² Different from Moortgat (1990), but in keeping with our modification (6) of his semantics (5), the left-hand side categories $(a_1 \backslash a_2) / a_3$, $(s/(n \backslash s)) \backslash s$ and $n \backslash s$ are assigned a variable v_0 .

¹³ Strictly speaking, f is partial, since terms assigned to goal categories a/b or $b \backslash a$ are not necessarily of the form $\lambda v. \gamma$. Below we will also employ a straightforward extension of f to interpreted sequents with asterisked goals: $f(T: T' \vdash at^* : \gamma) = T: T' \vdash at^* : \gamma$; $f(T: T' \vdash a/b^* : \lambda v. \gamma) = f(T: T', b: v \vdash a^* : \gamma)$; and $f(T: T' \vdash b \backslash a^* : \lambda v. \gamma) = f(b: v, T: T' \vdash a^* : \gamma)$. Note that the terms of goal categories in L^* derivations are in η -nf (see section 4 above), so the restriction of f to L^* -derivable sequents is total.

¹⁴ Examples are given in Chapter 4 of Hendriks (1992), which also contains proofs of the claims made below.

References

- Benthem, J. van: 1986, *Essays in Logical Semantics*, Reidel, Dordrecht.
 Benthem, J. van: 1991, *Language in Action. Categories, Lambdas and Logic*, North Holland, Amsterdam.
 Bouma, G.: 1989, Efficient Processing of Flexible Categorical Grammar, In *Proceedings of the Fourth Conference of the European Chapter of the ACL*, Manchester.
 Hendriks, H.: 1992, *Studied Flexibility*, Dissertation, University of Amsterdam.
 Hepple, M.: 1990, *The Grammar and Processing of Order and Dependency: a Categorical Approach*, Dissertation, University of Edinburgh.
 König, E.: 1989, Parsing as Natural Deduction, In *Proceedings of the 27th Annual Meeting of the ACL*, Vancouver.
 Lambek J.: 1958, The Mathematics of Sentence Structure. *American Mathematical Monthly* 65, 154-169.
 Moortgat M.: 1988, *Categorical Investigations: Logical and Linguistic Aspects of the Lambek Calculus*, Foris, Dordrecht.
 Moortgat M.: 1990, Unambiguous Proof Representations for the Lambek Calculus. In: M. Stokhof and L. Torenvliet (eds.), *Proceedings of the Seventh Amsterdam Colloquium*, ILLI, Amsterdam.
 Roorda, D.: 1991, *Resource Logics: Proof-Theoretical Investigations*, Dissertation, University of Amsterdam.

SUBDELETION AND THE LAMBEK CALCULUS*

Petra Hendriks
Department of Dutch
University of Groningen

Abstract

In this article I will consider a specific case of obligatory deletion, viz. Subdeletion, from a categorial perspective. An analysis of Subdeletion will be given within the framework of the Lambek Calculus (cf. Lambek 1958). The Lambek Calculus will be extended with a partial logic for the extraction operator, which was originally introduced by Moortgat (1988) to account for non-peripheral extraction. I will argue that Subdeletion is not a deletion operation at all, but rather the categorial equivalent of binding, and that comparative constructions involve two missing quantifiers rather than one.

1. *Introduction*

Categorial grammars are known for the burden they place upon the lexicon. Having a minimal amount of syntactic rules, they encode syntactic information as properties of lexical elements. From this perspective, deletion has to be regarded not as a syntactic operation but as a property of the deleted element or its syntactic environment. In general, two types of deletion are distinguished: when an element may be omitted and the meaning of the sentence doesn't change, optional deletion is said to apply, whereas we speak of obligatory deletion when, although the obligatorily deleted element is never allowed to occur, the sentence is interpreted as if this element does occur. In this case, the surrounding material behaves as if the obligatorily deleted element were in fact lexically present.

Because categorial grammars are monostratal, there is no deeper level of representation where the (optionally or obligatorily) deleted element, and therefore also its meaning, are still present while at surface structure deletion has taken place. When an element is deleted, it is entirely gone and no trace or empty position is left behind. The question arises, then, in what way deletion has to be represented in a categorial grammar for the meaning of the deleted element still to be available in the derivation of the sentence. Also, it is not clear how interaction takes place between the obligatorily deleted element, which is not present and has never been present in the derivation, and its syntactic environment. In this article, I will investigate the possibilities of a lexicalist approach to a specific case of obligatory deletion, Subdeletion.

2. Evidence for the presence of a quantifier

In Bresnan (1973, 1975, 1976), the comparative construction is investigated, and a rule is introduced to account for optional deletion of the compared element in the clause following *than* or *as*; this type of deletion is known as Comparative Deletion and is illustrated in (1). Bresnan (1975, 1976) argues that the rule she proposes for Comparative Deletion can also account for obligatory deletion of just the quantifier of the compared element. An example of this type of deletion, Subdeletion, can be seen in (2); according to Bresnan, the deleted quantifier is of the form *x many* or *x much*.

Before I go any further, let me first make some remarks concerning the terminology that I will use in this article. Because clausal comparatives are very much coordination-like, I will refer to the clause preceding *than* or *as* in examples like (1) and (2) as the first conjunct of the comparative, to the clause following *than* or *as* as the second conjunct, and to *than* and *as* themselves as comparative conjunctions (cf. Hendriks 1991). The term 'compared element' or 'compared constituent' will be used for the italicized phrases in sentences (1) and (2), and the elements *more*, *fewer*, *less*, *-er* and *as* will be referred to as the comparative operators, because a *than/as*-clause cannot occur without one of these elements being present in the first conjunct. The deletion site is indicated by the dash. Note that the term 'deletion' here is meant purely as a descriptive term; no reduction operation, which operates on a full comparative to yield a reduced comparative, is assumed.

- (1) a Mary ate *more bagels* than Susan ate ____.
b This building is *taller* than it seems ____.
c Mary drives *as carefully* as I drive ____.
- (2) a Mary ate *more bagels* than Susan ate ____ donuts.
b This building is *taller* than it is ____ wide.
c Mary drives *as carefully* as I drive ____ carelessly.

It seems questionable whether these two types of deletion should be accounted for by one rule. I will return to this matter in section 5, and give justification for distinguishing the two types of deletion. For the time being we will only be concerned with the type of deletion shown in (2).

According to Bresnan, the sentences in (2) involve obligatory deletion of a quantifying element in the second conjunct of the comparative. She gives syntactic as well as semantic evidence for the claim that there is a quantifier missing from the *than/as*-clause.

A first argument for this missing quantifier can be found in the interpretation of the sentence; sentence (2b), for example, is not understood as meaning that the building is wide (in fact, the building might be very narrow) but rather that the building is tall to a certain extent *x* and wide to a certain extent *y*, and $x > y$. In clausal comparatives, there is

always a comparison between two quantities or degrees. However, these quantities or degrees are never specified; the relation between the two is a relative one.

The strongest syntactic argument for the missing quantifier is the fact that no overt quantifier can appear in that position:

- (3) Mary ate more bagels than Susan ate (*two/*the/*some) donuts.

Related to this argument is the observation that can be made with respect to the subcategorization of certain verbs. Some verbs require a measure phrase complement, for example the verb *to weigh*. In the second conjunct of a comparative sentence, however, these verbs not only do not need a quantifying expression, they do not even allow a measure phrase as their complement, as can be seen in (4):

- (4) a This mouse weighs six ounces.
 b * This mouse weighs __ ounces.
 c * John weighs more pounds than this mouse weighs six ounces.
 d John weighs more pounds than this mouse weighs __ ounces.

Not only lexical, but also 'missing' measure phrases seem to be able to satisfy the subcategorization requirements of verbs like *to weigh*.

Another argument is the fact that the second conjunct of a comparative is not always a complete, grammatical sentence. In (5a), the *as*-clause *there were of us* is ungrammatical; however, if we would insert a quantifying element immediately before *of*, the result is a perfectly acceptable sentence: *there were two/some/that many of us*.

- (5) a There weren't as many of them as there were __ of us.
 b * There weren't as many of them as there were __ us.
 c * There weren't as many of them as there were __ of men.
 d There weren't as many of them as there were __ men.

A fifth argument can be found in the pattern of *of*-insertion. An NP can be immediately preceded by a quantifier, like *that many*, *so much*, *however many*, etc. If the NP is definite, *of*-insertion must take place. If, on the other hand, the NP is indefinite, *of* is not allowed between the quantifier and the NP. The same pattern of presence and absence of the preposition *of* is found in comparatives, although the quantifier is never overtly present in the second conjunct of the comparative. In (5a,b) the NP is definite and requires the occurrence of *of*; in (5c,d) no *of* is possible between the position from which the quantifier is missing, and the indefinite NP.

French (Pinkham 1982) and Dutch (Bennis 1978) also present evidence for a missing quantifier in the *than/as*-clause. In French, the element *en* may occur in the second conjunct of the comparative, replacing the compared element:

- (6) Il a acheté plus de livres qu'il ne pouvait en porter.
'he has bought more of books than he PRT could PRT carry'

The compared element *de livres* ('of books') is replaced by the proform *en*. This particle *en* has several uses. Pinkham shows that the element *en* in comparatives is identical to the element *en* which is used with quantifiers; therefore it is very plausible to assume a missing quantifier in French comparatives. Because of the similarity of the comparative construction in French and English, this missing quantifier in French is most likely to have a counterpart in English. The same line of reasoning can be followed for Dutch quantitative *er* ('there'). Just like French *en*, Dutch quantitative *er* occurs in the presence of a quantifying expression. Quantitative *er* is also able to replace the compared element in the second conjunct of a comparative (Bennis 1978).

- (7) a Jan kocht meer boeken dan Marie boeken kocht.
'John bought more books than Mary books bought'
b Jan kocht meer boeken dan Marie er kocht.
'John bought more books than Mary there bought'

A final argument for the assumed gap is the behavior of comparatives with respect to coordination. Two or more constituents of the same lexical type can, in principle, be conjoined by the coordinator *or*. So conjunction of the two Ss *Bill saw cows* and *Pete talked to the dog* gives a grammatical result. However, coordination of these two sentences is blocked when the coordinate construction is embedded in a comparative clause:

- (8) a [Bill saw cows] or [Pete talked to the dog].
b * John saw more horses than [[Bill saw cows] or [Pete talked to the dog]].

Sentence (8b) has been taken from Gazdar (1981). The difference in grammaticality between (8a) and (8b) could very well be accounted for by assuming that in (8b) the two sentences are not of the same type, unlike in (8a). This would be the case if one of the two sentences is missing a quantifying element and the other is not. In *Pete talked to the dog* no element can function as the compared element because no element in this sentence can be quantified. Therefore, *Pete talked to the dog* cannot be lacking a quantifier and must be a full sentence. In *Bill saw cows*, on the other hand, the NP *cows* can be quantified and hence can function as the compared element. Because *Bill saw cows* is a sentence with a gap it is no longer of the lexical type S, so the conjunction in sentence (8b) is not between two elements of the same lexical type and is therefore ruled out. This assumption would predict that if the two conjoined sentences that are embedded in the comparative clause both contain a quantifiable element, the sentence is grammatical again. This prediction is borne out by the following sentence:

- (9) John saw more horses than [[Bill saw cows] or [Pete talked to dogs]].

In *Pete talked to dogs* the NP *dogs* can be quantified, just like *cows* in the other conjunct, so both conjuncts are sentences lacking a quantifier now. Because the two conjuncts are of the same type, the comparative sentence in (9) is a well-formed sentence.

From the observations above we can conclude that in the second conjunct of clausal comparatives a quantifying element is missing. This quantifying element is never allowed to occur but can be detected through the behavior of other elements in the sentence, and through the way in which the sentence is understood. What syntactic process exactly is responsible for the absence of this quantifier, has been a matter of debate in generative literature in the mid seventies. In the next section I will discuss the three most important analyses of Subdeletion: the deletion approach (Bresnan), the movement approach (Chomsky) and the binding approach (Pinkham).

3. Previous analyses of Subdeletion

In Bresnan's (1975, 1976) analysis, Subdeletion and Comparative Deletion are accounted for by one and the same rule, given in (10) (Bresnan 1976:372).

$$(10) \quad \begin{array}{cccccc} [X''X'' \dots] & [S'' \dots [X''X'' \dots] \dots] & & & & \\ 1 & 2 & 3 & 4 & 5 & 6 \end{array} \quad \Rightarrow \quad \emptyset$$

Bresnan states that the compared constituent in the second conjunct is deleted under identity with the compared constituent in the first conjunct. This type of deletion is called 'deletion across a variable', because the antecedent of the deletion operation (the compared constituent in the first conjunct) may be arbitrarily far from the deletion site.

Chomsky (1977), on the other hand, suggests that Comparative Deletion involves Wh-movement, moving the compared constituent to the COMP-position of the *than/as*-clause, where it subsequently is deleted. Such a movement analysis would account for the island effects that Comparative Deletion exhibits, and also for some other properties that Chomsky considers to be characteristic for Wh-movement. Chomsky notes that this Wh-movement approach might apply to Subdeletion as well, depending on whether Subdeletion really exhibits the Wh-diagnostics. Under this analysis, the quantifier would be moved to the COMP-position of the *than/as*-clause, where deletion takes place.

A third analysis of Subdeletion is the binding analysis proposed by Pinkham (1982). In this proposal a null element Q is assumed to be base generated in the *than/as*-clause, occupying the specifier position of the

compared element. This variable Q must be bound by the comparative conjunction in COMP, under certain conditions stated for quantifier binding. Pinkham assumes furthermore, following Bresnan (1973), that the *than/as*-clause is generated in the QP containing the comparative operator in the first conjunct, and is moved to the end of the sentence by a rule of extraposition.

We will see in sections 4 and 5 that these three analyses are incompatible with the facts. The deletion approach, the movement approach and the binding approach can each explain only part of the properties of Subdeletion that are listed in the next section.

4. Characteristics of Subdeletion

The most striking characteristic of Subdeletion is the fact that it is obligatory. As we already saw in section 2, a quantifier must be missing from the second conjunct of the comparative (see example 3).

Secondly, an important aspect in the interpretation of comparatives is the relation between the two compared elements. In comparatives, there is always a comparison between two quantities or degrees, one referred to by the first conjunct and one referred to by the second conjunct. The element in the first conjunct from which the quantity or degree is compared, is introduced by one of the elements *more*, *fewer*, *less* or *as* or followed by the suffix *-er*. The element in the second conjunct of which the quantity or degree is compared, can never be modified. This element is the element that follows the missing quantifier.

A third property of Subdeletion is the fact that it does not seem to show any of the diagnostics for gaps. One of these diagnostics is the *that*-trace effect: the complementizer *that* cannot be followed by a deletion site in English. In sentence (11a), a sentence taken from Grimshaw (1987), the position from which the quantifier is obligatorily missing is immediately to the right of the complementizer. Nevertheless, the sentence is acceptable. If the gap is the result of Comparative Deletion, on the other hand, as in sentence (11b), the sentence is ungrammatical.

- (11) a Even fewer books were published than we expected that __
magazines would be.
b * Even fewer books were published than we expected that __
would be.

According to Grimshaw, the ungrammaticality of (11b) is due to the *that*-trace effect: if *that* is omitted, the sentence becomes grammatical:

- (12) Even fewer books were published than we expected __ would be.

Another indication of the presence of a gap is the possibility of auxiliary contraction. It is generally assumed that contraction of the auxiliary is

blocked by the presence of a gap to the right. Grimshaw (1987:665) shows that, although contraction in sentences involving Subdeletion is not completely perfect, these sentences are much better than the sentences in which the contracted auxiliary is followed by a gap that is the result of Comparative Deletion:

- (13) a There's more meat than there is/'s __ rice.
 b I thought there was more meat than there is/*'s __.

So there is a sharp distinction between Comparative Deletion and Subdeletion with respect to the diagnostics for the presence of a gap: Comparative Deletion results in a gap, with all the characteristics of a gap, but the missing quantifier that is the result of Subdeletion does not display any of these diagnostics.

Subdeletion is an unbounded phenomenon. This is illustrated by the contrast between (14a) and (14b). In (14a), the Subdeletion site occurs embedded in a sentence in the *than*-clause. There is a clear contrast in grammaticality between this sentence and (14b), a sentence in which Gapping has applied in an embedded clause. Gapping is a bounded deletion rule, and therefore Gapping of the finite verb of a sentence which is embedded inside the second conjunct, is not allowed.

- (14) a ? John drank more beer than [Bill believes [that Mary has drunk vodka]].
 b * John drank beer and [Bill believes [that Mary __ vodka]].

A fifth property of Subdeletion manifests itself in its behavior in island configurations. As was already noticed by Bresnan (1975:60), Subdeletion violates the Left Branch Condition. In general, extraction out of left branch positions is not possible. So in (15a), the measure phrase modifier in left branch position, *how many*, cannot be moved away from the element it modifies, *records*. In (15b), however, the same type of phrase is lacking from the compared noun phrase *records* in a comparative clause.

- (15) a * [How many]_i did John buy [t_i records]?
 b Bill bought more books than John bought [__ records].
 (16) a * [How]_i does Mary drive [t_i carefully]?
 b Mary drives as carefully as I drive [__ carelessly].

The same holds for the adverb phrases in (16): extraction of the left branch modifier *how* is not possible, but this element can be absent as a result of Subdeletion.

Corver (1990) claims that Subdeletion can also occur in other typical island configurations, in contrast with Comparative Deletion. Corver states that subject NPs are considered to be islands with respect to transformations, and therefore do not allow Comparative Deletion. Subdeletion, on the other hand, is permitted in subject NPs. This is illustrated by the examples in (11). However, as we saw earlier, the

ungrammaticality of Comparative Deletion of the subject NP in (11b) is due to the *that*-trace effect. So the island character of subject NPs is not that clear. Another typical island configuration, which Corver claims that is violated by Subdeletion, is the Dutch PP. Again, sentences in which Subdeletion has applied, are much better than sentences in which Comparative Deletion has taken place.

- (17) a Jan heeft voor meer voetbalclubs gevoetbald dan hij voor __ tennisclubs getennist heeft.
'John has for more soccerteams played-soccer than he for tennisclubs played-tennis has'
- b * Jan heeft voor meer clubs gevoetbald dan hij voor __ getennist heeft.
'John has for more teams played-soccer than he for played-tennis has'

However, Subdeletion within Dutch PPs can also be explained without having to assume an island violation. Van Riemsdijk (1978) gives examples in which extraction out of a Dutch prepositional phrase is allowed:

- (18) a Waar_i heb je [op t_i] gerekend?
'where have you on counted'
- b Ik heb [met de meeste mensen t_i] gesproken [die er waren]_i.
'I have with the most people spoken who there were'
- c omdat zij [de boom]_i op blote voeten [in t_i] klommen
'because they the tree on bare feet into climbed'

Movement of *r*-pronouns (18a), modifying clauses (18b) and objects of motional postpositions (18c) is possible out of a Dutch prepositional phrase. So the island character of the Dutch PP is relative with respect to the nature of the extracted element. It is not unlikely that the subdeleted quantifier is not sensitive to the island character of the PP either, and behaves more or less like an *r*-pronoun with respect to movement. Evidence supporting this claim can be found in the behavior of Subdeletion in PPs introduced by the preposition *sinds* ('since'). *Sinds*, like all temporal prepositions, never allows extraction (Van Riemsdijk 1978:201). Even *r*-pronouns cannot be extracted from a *sinds*-PP, as (19a) shows. The subdeleted quantifier behaves similarly: Subdeletion within *sinds*-PPs is clearly much worse than Subdeletion within other PPs:

- (19) a * Hij is er_i sinds t_i ziek.
'he is there since ill'
- b ??Hij is sinds meer maanden werkeloos dan zij sinds __ weken ziek is.
'he is since more months unemployed than she since weeks ill is'

Bresnan (1975) points out that Subdeletion observes all other island

constraints:

- (20) a * John bought more books than Bill believed [the claim that Mary had bought __ records].
b * John bought more books than Bill wondered [why Mary had bought __ records].
c * John bought more books than [that Mary had bought __ records] was likely.

Violation of the Complex NP Constraint (20a), the Wh-island Constraint (20b) and the Sentential Subject Constraint (20c) results in ungrammaticality. So, summarizing, Subdeletion observes all island constraints, except for the Left Branch Condition.

Another property of Subdeletion is the fact that the missing quantifier can be absent from any position in the sentence. The compared element may occur in sentence-initial position (11a), in sentence-final position (2), or anywhere in between (17a). Moreover, there does not have to be any parallelism between the compared element in the first conjunct and the compared element in the second conjunct. The compared elements at both sides of the comparative conjunction do not need to occur in the same structural position, nor do they have to have the same syntactical function in the sentence, as (21) illustrates:

- (21) Fred has quoted more professors than __ professors have quoted him.

The compared element *professors* in the first conjunct is a non-topicalized direct object, whereas the compared element *professors* in the second conjunct is a subject in sentence initial position. So no parallelism is required between the two compared elements in a simple comparative.

Comparatives are very much coordination-like: they behave like coordinate constructions and unlike subordinate constructions with respect to deletion and movement (Hendriks 1991). Deletion operations like Gapping, Right Node Raising and Conjunction Reduction may apply in coordinate constructions, but not in subordinate ones. These deletion operations are also allowed in comparatives, as is illustrated by the sentences (22) - (24):

- (22) a Mary kissed John and Sue __ Bill.
b * Mary kissed John when Sue __ Bill.
c Mary kissed more boys than Sue __ girls.
(23) a John hates __ and Mary likes books on modern painting.
b * John bought __ because Mary likes books on modern painting.
c More men buy __ than women borrow books on modern painting.
(24) a Mary drinks vodka and __ smokes cigars.

- b * Mary drinks vodka when __ smokes cigars.
- c More women drink vodka than __ smoke cigars.

The a-sentences involve coordination, the b-sentences subordination and the c-sentences comparativization. Gapping (22), Right Node Raising (23) and Conjunction Reduction (24) may apply in both coordinated sentences and comparatives, but not in subordinated sentences. Another similarity between coordinated sentences and comparatives is their behavior with respect to movement. In both types of construction, an element can be moved only if it is moved from all conjuncts simultaneously, i.e. in an Across-the-board (ATB) way. This accounts for the observations in (25) and (26): the Wh-element *which actor* cannot be extracted from one conjunct only, as in (25a) and (26a), but has to be moved in an ATB manner, as the grammaticality of the b-sentences demonstrates. Movement out of a subordinated clause is not restricted to ATB application.

- (25) a * Which actor_i does John like t_i and Mary hates Tom?
- b Which actor_i does John like t_i and Mary hate t_i?
- (26) a * Which actor_i do as many men like t_i as women hate Tom?
- b Which actor_i do as many men like t_i as women hate t_i?

A subordinated clause can be fronted without causing ungrammaticality, as can be seen by the sentences (28). In coordinate constructions (27) and comparatives (29), on the other hand, this type of movement is ruled out.

- (27) a John bought two books and Mary bought three records.
- b * And Mary bought three records, John bought two books.
- (28) a John got a lot of flowers because he was ill.
- b Because John was ill, he got a lot of flowers.
- (29) a John bought more books than Mary bought records.
- b * Than Mary bought records, John bought more books.

A final property of Subdeletion seems to be the possibility of multiple comparison (Von Stechow 1984). The following sentence is taken from Von Stechow:

- (30) More dogs ate more rats than cats ate mice.

Von Stechow claims that this sentence must be interpreted as involving two pairs of compared elements. The first comparison is between the amount of dogs and the amount of cats, and the second one is between the amount of rats and the amount of mice. This interpretation is paraphrased in (31).

- (31) The number of dogs that ate rats is greater than the number of cats that ate mice & the number of rats that were eaten by dogs is greater than the number of mice that were eaten by cats.

However, most people find this interpretation impossible to get. Because of the oddness of a multiple comparison sentence with two different

comparative operators, it is highly likely that the second occurrence of *more* is a vacuous one:

(32) ??More dogs ate less rats than cats ate mice.

Summarizing the results thus far, Subdeletion has the following properties:

- A) A quantifier is obligatorily missing from the *than/as*-clause.
- B) The element following this missing quantifier stands in a certain relation to, and is compared with, an element in the first part of the comparative clause. This element is introduced by a comparative operator (*more*, *fewer*, *less*, *-er* or *as*).
- C) The missing quantifier does not display any of the diagnostics for gaps.
- D) Subdeletion is an unbounded phenomenon.
- E) The missing quantifier observes all island constraints on movement and deletion, except for the Left Branch Condition.
- F) The quantifier can be absent from any position in the *than/as*-clause. No parallelism with respect to function or position in the sentence is required between the two compared elements.
- G) The comparative conjunctions *than* and *as* behave like coordinate conjunctions, and unlike subordinate conjunctions, with respect to deletion and movement.
- H) Double comparison seems to be possible.

In the next section I will show that neither a deletion approach nor a movement approach can account for all of the properties observed with Subdeletion. Pinkham's binding approach seems to be more promising, but needs to be reformulated to a great extent.

5. The inadequacy of previous accounts of Subdeletion

A first problem with Bresnan's treatment of Subdeletion is the fact that Subdeletion and Comparative Deletion are represented by the same rule. In the previous section, we observed several differences between Subdeletion and Comparative Deletion: Subdeletion is obligatory, Comparative Deletion is optional; Comparative Deletion leaves a gap, Subdeletion does not; Subdeletion is possible in Dutch PPs, Comparative Deletion is not. However, even if Subdeletion would be accounted for by a separate deletion rule, there still would be no way to account for the fact that Subdeletion does not leave a gap. Moreover, the fact that Subdeletion obeys all island constraints (except for the Left Branch Condition) also cannot be explained by a deletion approach.

Chomsky's movement approach is able to account for the fact that Subdeletion observes island constraints like the Complex NP Constraint, the Wh-island Constraint, and the Sentential Subject Constraint, but cannot explain why the Left Branch Condition is not observed. More importantly, the fact that Subdeletion does not leave a gap also remains unexplained

under a movement analysis. Additional evidence against a movement analysis of Subdeletion is presented by Corver (1990). Corver shows that Subdeletion is still possible even if there is no suitable landing site available for the moved quantifier, for example in topicalized VPs and in participial phrases modifying a noun.

An analysis based on binding, on the other hand, would not encounter any problems with respect to the absence of a gap. Assuming a null quantifier to be base generated in the second conjunct of a comparative would account for the absence of any of the diagnostics for the presence of a gap, since no operation has applied to remove the quantifier. However, although a treatment of Subdeletion in terms of binding certainly is the most promising of the three, some problems arise with a binding analysis such as Pinkham's as well. First of all, it is not clear why the conditions on Q-binding should be practically identical to the conditions on movement. Subdeletion observes all conditions on movement, except for the Left Branch Condition. If a binding relation holds between the comparative conjunction and the null quantifier Q, this Q-binding must be subject to the Complex NP Constraint, the Wh-island Constraint and the Sentential Subject Constraint. To make the right predictions, Pinkham has to stipulate that these conditions on movement also hold for Q-binding. Furthermore, an analysis of comparatives in which the *than/as*-clause is generated under the QP in the first conjunct, would imply a subordinate structure for the comparative sentence, which would be in contradiction with the observed coordination-like behavior of comparatives. This coordination-like behavior is very hard to explain if the *than/as*-clause is assumed to be a postmodifier of the comparative operator. Finally, multiple occurrences of *more* are rather problematic, whether the second occurrence of *more* is vacuous or not, if a *than*-clause is assumed to be base generated under every QP dominating *more*.

6. Subdeletion and the Lambek Calculus

In this section I will discuss the possibilities of an alternative approach to Subdeletion, based on the Lambek Calculus. The Lambek Calculus is a surface-oriented and linear order preserving categorial grammar, thus avoiding movement and deletion rules. Like all categorial grammars, it favors radical lexicalism, encoding syntactic information in its lexical category specification. The combinatory power of categorial grammars lies in the division of categories into complete and incomplete expressions; incomplete expressions are modelled as functors, complete expressions as arguments. In a bidirectional categorial grammar like the Lambek Calculus, functors are built up out of primitive (argument) types by means of the type constructors / and \. An incomplete expression of type X/Y prefixes itself to expressions of type Y, yielding expressions of type X. A

functor of type $Y \backslash X$ is satisfied by an argument expression of type Y to its left. An example of this can be found in (33).

- (33) John sleeps
NP NP\S

Here the proper name *John* is a primitive, a complete expression; the intransitive verb *sleeps*, on the other hand, is incomplete: it combines with a noun phrase to yield a sentence. In combining two expressions, a very close parallel is said to hold between syntax and semantics: this is expressed by the Curry-Howard isomorphism, which states that there is a correspondence between intuitionistic natural deductions (the syntactic derivations of expressions) and typed lambda terms (the semantic terms associated with those expressions). Every syntactic rule corresponds to a semantic operation. So the application of the functor $NP \backslash S$ to its argument NP in (33) corresponds to the application of the meaning of functor expression to the meaning of the argument expression. Besides considering *sleeps* in (33) an incomplete expression, it is also possible to consider *John* an incomplete expression, which combines with an intransitive verb of type $NP \backslash S$ to form an expression of type S.

- (34) John sleeps
S/(NP\S) NP\S

Combination of the functor $NP \backslash S$ with its argument NP to yield an S, as in (33), is the result of functional application. Functional application is one of the reduction rules that are derivable as theorems in the Lambek Calculus (Lambek 1958). The logic of the Lambek Calculus consists of one axiom and for each logical connective two inference rules (a left rule and a right rule) regulating their behavior. These rules of inference can be expressed in the Gentzen sequent calculus notation (cf. Moortgat 1988). Sequents are of the form $\Gamma \Rightarrow A$, where Γ is a sequence of types. The sequent $\Gamma \Rightarrow A$ asserts that there is a deduction of the succedent type A from the antecedent types Γ , under the restriction that Γ must be non-empty.

- (35) a axiom: $A \Rightarrow A$
b [/R]: $\Gamma \Rightarrow A/B$ if $\Gamma B \Rightarrow A$
c [\R]: $\Gamma \Rightarrow B \backslash A$ if $B \Gamma \Rightarrow A$
d [/L]: $\Gamma A/B \Delta \Phi \Rightarrow C$ if
 $\Delta \Rightarrow B$ and $\Gamma A \Phi \Rightarrow C$
e [\L]: $\Gamma \Delta B \backslash A \Phi \Rightarrow C$ if
 $\Delta \Rightarrow B$ and $\Gamma A \Phi \Rightarrow C$

A, B, C are single occurrences of a type and Γ, Δ, Φ are sequences of types. The axiom in (35a) is the axiom of identity. The right rules [/R] and [\R] introduce an occurrence of one of the type forming connectives in the succedent type. Application of the two inference rules [/L] and [\L] eliminates an occurrence of the type forming connective / and \,

respectively, in the antecedent.

In order to prove the validity of a sequent $\Gamma \Rightarrow A$ I will use the Gentzen proof procedure. Lambek (1958), in his original presentation of the calculus, already uses this proof-search strategy, and its merits are further investigated by Moortgat (1988). The proof search occurs in a top-down manner, from a proof-solving perspective. Starting with the sequent of which we want to prove the validity, and applying the inference rules for the connectives backward, we work downward towards axiom sequents. The Lambek-validity of a sequent is proved if all 'branches' of the proof tree are reduced to axiom sequents. The inference rules are of the format '*conclusion*' if '*premise*'; backward application of the inference rules derives the premise from the conclusion: '*conclusion*' $\Rightarrow$ '*premise*'. A backward applied left rule removes a connective from the antecedent types Γ of the conclusion, given a number of premises, and a right rule removes a connective from the succedent type A of the conclusion if applied backward. Because every time an inference rule applies, a connective is removed from a finite sequent, the number of steps that have to be taken to decide on the validity or invalidity of a sequent is also finite. So the Gentzen proof strategy establishes the decidability of the Lambek Calculus. An illustration of the Gentzen proof procedure is given in (36).

(36)

John	puts	the book			(on the table)
NP	((NP\S)/PP)/NP	NP/N	N	$\Rightarrow$	S/PP
					[/R]
NP	((NP\S)/PP)/NP	NP/N	N	PP	$\Rightarrow$ S
					[/L]
N	$\Rightarrow$ N	NP	((NP\S)/PP)/NP	NP	PP $\Rightarrow$ S
					[/L]
axiom	NP $\Rightarrow$ NP	NP	((NP\S)/PP)	PP	$\Rightarrow$ S
					[/L]
	axiom	PP	$\Rightarrow$ PP	NP	NP\S $\Rightarrow$ S
					[/L]
		axiom	NP $\Rightarrow$ NP	S	$\Rightarrow$ S
					[/L]
			axiom	axiom	

The bold face connectives are the active connectives, i.e. the connectives that are removed. They correspond to the connectives in the inference rules. The first step in (36) is backward application of the right rule for the connective /. The complete string 'NP, ((NP\S)/PP)/NP, NP/N, N' corresponds to Γ in the formula, S corresponds to A and PP is type B in the formula. (36) proves that reduction of the types assigned to the string 'John puts the book' to type S/PP is Lambek-valid: all branches are reducable to axioms.

Now that the logic of the Lambek Calculus has been defined, we can turn back to the sentences involving Subdeletion. The most direct approach to Subdeletion in a categorial grammar would be to consider the clause following *than* or *as* an incomplete sentence, missing a quantifying element. In the Lambek framework outlined above this would boil down to assigning the comparative conjunction the lexical type $(S\S)/(S/(XP/X))$, where XP/X is the missing quantifying element. The polymorphic type X is the type of the compared element, and can be an N, A, etc. (37) is a proof

(37)	They drank more vodka than we drank lemonade	
	$NP, (NP \setminus S) / NP, NP / N, N, (S \setminus S) / (S / (NP / N)), NP, (NP \setminus S) / NP, (NP / N) \setminus NP \Rightarrow S$	[/L]
<u>$N \Rightarrow N$</u>	$NP, (NP \setminus S) / NP, NP, (S \setminus S) / (S / (NP / N)), NP, (NP \setminus S) / NP, (NP / N) \setminus NP \Rightarrow S$	[/L]
<u>axiom</u>	$NP \Rightarrow NP \quad NP, NP \setminus S, (S \setminus S) / (S / (NP / N)), NP, (NP \setminus S) / NP, (NP / N) \setminus NP \Rightarrow S$	[\L]
<u>axiom</u>	$NP \Rightarrow NP \quad S, (S \setminus S) / (S / (NP / N)), NP, (NP \setminus S) / NP, (NP / N) \setminus NP \Rightarrow S$	[/L]
<u>axiom</u>	$NP, (NP \setminus S) / NP, (NP / N) \setminus NP \Rightarrow S / (NP / N) \quad S, S \setminus S \Rightarrow S$	[/R]
	$NP, (NP \setminus S) / NP, (NP / N) \setminus NP, NP / N \Rightarrow S \quad S \Rightarrow S \Rightarrow S$	[/L]
	$(NP / N) \setminus NP, NP / N \Rightarrow NP \quad NP, NP \setminus S \Rightarrow S \quad \text{axiom axiom}$	[\L]
	<u>fail</u>	<u>$NP \Rightarrow NP \quad S \Rightarrow S$</u>
	<u>axiom</u>	<u>axiom</u>

$$(38) \quad [\uparrow R]: \quad r_1 \ r_2 \Rightarrow B \uparrow A \quad \text{if} \quad r_1 \ A \ r_2 \Rightarrow B$$

247

right rule for the extraction operator to the Lambek Calculus has as a consequence that a functor expression with a non-peripheral gap cannot be satisfied by simple functional application of this functor to its argument, since there is no way to eliminate $\uparrow$. The only way to use a type $B\uparrow A$ in the derivation is to have it as the argument of a gap-filler type, i.e. as the argument of a functor over gapped arguments (the type of such a functor is $C/(B\uparrow A)$ or $(B\uparrow A)\backslash C$).

Adding a partial logic for the extraction operator $\uparrow$ to the Lambek Calculus not only makes it possible to analyze non-peripheral extraction within the extended Lambek framework, it also allows for the derivation of sentence (2a). The second conjunct of the comparative can now be analyzed as an incomplete sentence that is missing a quantifying element somewhere in it, not necessarily peripherally. If the comparative conjunction is assigned the lexical (gap-filler) type $(S\backslash S)/(S\uparrow(NP/N))$, sentence (2a) can be derived without any problem. However, if we assign this type to the comparative conjunction, any sentence can precede the comparative conjunction as long as its lexical type is an S . This is in contradiction with the facts; as observed before, it is not the comparative conjunction that seems to trigger the comparative construction, but rather the comparative operator (*more*, *fewer*, *less*, *-er* or *as*). Absence of the comparative operator blocks the comparative construction. Therefore I will propose an analysis for Subdeletion, which also captures this observation (the analysis I will propose for compared nouns is assumed to hold for other categories as well). The only assumption I have to make for my analysis involves the lexical type of 'quantifiable' constituents. 'Quantifiable' constituents are constituents that can be preceded by a quantifying expression. I will assume that quantifiable elements are incomplete expressions, lacking a quantifier. This kind of analysis for quantifiable elements is not new; a missing quantifier is also proposed by Kayne (1975, 1984) for French objects of the form *de* + articleless NP in certain environments, to account for the ECP effects that these constructions display.

In categorial terms these incomplete quantifiable constituents would be functors over a quantifier argument. The missing quantifiers are the categorial analogues of variables in semantics. The quantifier type Q is shorthand for NP/N , so the full type of a quantifiable common noun is $(NP/N)\backslash NP$. The type transition from N to $(NP/N)\backslash NP$ is Lambek-valid; it is established through simple type lifting. Satisfaction of the functor $Q\backslash XP$ by a quantifier argument and application of a higher order functor to its argument $Q\backslash XP$ are the categorial analogues of binding of this variable in semantic terms. The Gentzen proof tree in (39) shows that sentence (2a) now is derivable as follows, given the type assignment below:

(39)	They	drank	more	vodka	than	we	drank	lemonade	
	NP, (NP\S)/NP, (Q _c \NP)/(Q\NP), Q\NP, ((StQ _c)\S)/(StQ), NP, (NP\S)/NP, Q\NP => S								
									[/L]
	Q\NP=>Q\NP	NP, (NP\S)/NP, Q _c \NP, ((StQ _c)\S)/(StQ), NP, (NP\S)/NP, Q\NP => S							[/L]
axiom		NP, (NP\S)/NP, Q _c \NP, (StQ _c)\S => S						NP, (NP\S)/NP, Q\NP=>StQ	[/L]
			[tR]						[tR]
		NP, (NP\S)/NP, Q _c \NP=>StQ _c		S=>S				NP, (NP\S)/NP, Q, Q\NP=>S	[/L]
									[tR]
		NP, (NP\S)/NP, Q _c , Q _c \NP=>S		axiom				Q=>Q NP, (NP\S)/NP, NP=>S	[/L]
			[tR]						[/L]
	Q _c =>Q _c	NP, (NP\S)/NP, NP=>S						axiom NP=>NP NP, NP\S=>S	[/L]
			[/L]						[/L]
axiom		NP=>NP NP, NP\S=>S						axiom NP=>NP S=>S	[/L]
			[/L]						[/L]
		axiom NP=>NP S=>S						axiom axiom	
		axiom axiom							

The comparative conjunction *than* is looking for a sentence with a missing quantifier Q to its right and for a sentence with a missing quantifier Q to its left. The quantifier in the first conjunct has to bear the subscript 'c'; this subscript is introduced when the comparative operator applies to the compared element.

7. Consequences of the proposed analysis

A) The comparative conjunctions *than* and *as* require incomplete sentences at both sides, because they are functors looking both to the left and to the right for an incomplete sentence as their argument. These incomplete sentences lack a quantifying element. Thus the obligatorily missing quantifiers result from the lexical type of the comparative conjunction. The claim that both conjuncts are incomplete sentences, leads to the prediction that, since the second conjunct is subject to island restrictions, the first conjunct should be too. This prediction seems to be borne out by the following sentences:

- (40) a * John gave [the girl who owns more Porsches] a book than Bill gave ___ boys a record.
 b * [That John had bought more books] was likely than Mary bought ___ records.

In (40a), the compared element is embedded in a relative clause, in (40b) in a sentential subject. These sentences are just as bad as their counterparts in which the island violation is in the right conjunct:

- (41) a * John gave more girls a book than Bill gave [the girl who owns Porsches] a record.
 b * John bought more books than [that Mary had bought ___ records] was likely.

As was observed in section 3, the compared element in the second conjunct can occur embedded. Barbara Partee pointed out to me that this

is also possible with the compared element in the first conjunct:

- (42) a The ad explicitly stated [that they would hire more syntacticians] than I expect our department to hire phonologists (though I no longer remember the exact number).
b That department plans [to hire more syntacticians next year] than any other department has ever hired linguists in a single year.

The compared element is embedded in a *that*-clause in (42a) and in an infinitive in (42b).

B) Both the clause preceding *than* or *as* and the clause following this comparative conjunction are functors ranging over quantifier arguments (they are of type $S+Q$). The comparative conjunction establishes a relation between these two quantifying elements by taking these two functors as its arguments.

C) Subdeletion does not show any of the diagnostics for gaps, because it is intrinsically different from 'deletion' operations like Comparative Deletion and Gapping. Subdeletion is no deletion at all, but merely the effect of subcategorization requirements of the comparative conjunction.

D & E) The compared constituents in a comparative construction are incomplete for a quantifying element at their left hand side. The Left Branch Condition must be a constraint on any material between the 'left branch' element and the constituent it modifies. Because the quantifier is not present in the sentence, the Left Branch Condition does not apply. From the lexical type assignment on, the obligatorily incomplete compared constituent is identical to a constituent from which a subpart has been extracted or deleted: they are of type $A\backslash B$ or B/A . The information about the missing argument has to be passed on to sentence level by combining the elements in the sentence. Under the assumption that island conditions are (syntactic or semantic) restrictions on the combination of elements, it is obvious that all island conditions hold for Subdeletion as well. In particular, Subdeletion corresponds to unbounded movement.

F) From the logic of the extraction operator it follows directly that the absent quantifier can be absent from any position in the sentence. Moreover, because the two extraction operators (the one in the first conjunct and the one in the second conjunct of the comparative) function independently, the two compared elements do not have to occupy identical positions in the sentence. Note that the proposed analysis does not hinge on the choice of this particular operator. An analysis in terms of the structural modalities $>Perm$ and $<Perm$ (Morrill, Leslie, Hepple & Barry 1990) would look rather similar. The important point is that we need some

kind of controlled permutation; how this controlled permutation is established, is not relevant for the proposed analysis.

G) The type of the comparative conjunction, $((S \uparrow Q_c) \backslash S) / (S \uparrow Q)$, is analogous to the type of coordinate conjunctions like *and* and *or*, which is $(S \backslash S) / S$ for sentence conjunction. This might explain why comparatives are so much coordination-like in their behavior with respect to deletion and movement.

H) If sentences containing more than one occurrence of *more* are interpreted as involving comparison of pairs of compared elements, the lexical type of the comparative conjunction can easily be extended to account for these cases as well. The type of the comparative conjunction then would be $((((S \uparrow Q_c) \uparrow Q_c') \backslash S) / ((S \uparrow Q) \uparrow Q'))$, where a relation must be defined between two pairs of quantifiers: Q_c and Q , and Q_c' and Q' . If, on the other hand, the second occurrence of *more* is merely a vacuous occurrence, the two indices 'c', introduced by the two occurrences of *more*, would have to merge by a mechanism like feature unification.

7. Conclusions

In this article I have argued that Subdeletion is neither deletion nor movement. Rather, Subdeletion is the effect of the subcategorization requirements of the comparative conjunctions *than* and *as*. These elements require an incomplete sentence, lacking a quantifier, at both sides. The missing quantifiers turn up in the lifted type of quantifiable constituents. Incorporation of this assumption into the framework of the Lambek Calculus, extended with a partial logic for the extraction operator $\uparrow$, allows us to account for the observed properties of Subdeletion.

This analysis differs from previous analyses of Subdeletion in the assumption that the compared elements in the two conjuncts of the comparative have identical structures. A quantifier is missing from both conjuncts, but can be detected in the second conjunct only, because the incomplete compared element in the first conjunct is immediately absorbed by the comparative operator through functional application.

Because Subdeletion is no obligatory deletion, the question arises whether obligatory deletion exists at all. It is not unlikely that optional deletion, which deletes an element under identity with an element in the other conjunct, is the only kind of deletion possible. This matter, however, is left for further research, as is the problem of how to analyze these cases of optional deletion in a lexicalist framework.

Acknowledgements

* This research was supported by grants from the Netherlands Organization for Scientific Research (NWO), grant nos. SIR 11-414 and R 30-337, and by a grant from the Dutch Network for Language, Logic and Information (NNTLI). I would like to thank Ale de Boer, Jack Hoeksema, Polly Jacobson, Michael Moortgat, Barbara Partee, Henriëtte de Swart and Frans Zwarts for their valuable comments on earlier drafts of this article.

References

- Bennis, H.: 1978, *Comparative Deletion is Subdeletion*, MA thesis, University of Amsterdam, Amsterdam
- Bresnan, J.: 1973, Syntax of the Comparative Clause Construction in English, *Linguistic Inquiry* 4(3), 275-343
- Bresnan, J.: 1975, Comparative Deletion and Constraints on Transformations, *Linguistic Analysis* 1(1), 25-74
- Bresnan, J.: 1976, Evidence for a Theory of Unbounded Transformations, *Linguistic Analysis* 2(4), 353-393
- Chomsky, N.: 1977, On Wh-movement, in A. Akmajian, P. Culicover & T. Wasow (eds.) *Formal Syntax*, Academic Press, New York, 71-132
- Corver, N.: 1990, *The Syntax of Left Branch Extractions*, Ph.D. thesis, KUB, Tilburg
- Gazdar, G.: 1981, A Phrase Structure Syntax for Comparative Clauses, in T. Hoekstra, H. van der Hulst & M. Moortgat (eds.) *Lexical Grammar*, Foris, Dordrecht, 165-180
- Grimshaw, J.: 1987, Subdeletion, *Linguistic Inquiry* 18(4), 659-669
- Hendriks, P.: 1991, The Coordination-like Structure of Comparatives, in F. Drijkoningen & A. van Kemenade (eds.) *Linguistics in the Netherlands 1991*, John Benjamins Publishing Co., Amsterdam/Philadelphia, 41-50
- Kayne, R.S.: 1975, *French Syntax*, MIT press, Cambridge, Massachusetts
- Kayne, R.S.: 1984, *Connectedness and Binary Branching*, Foris, Dordrecht
- Lambek, J.: 1958, The Mathematics of Sentence Structure, *American Mathematical Monthly* 65, 154-170
- Moortgat, M.: 1988, *Categorial Investigations. Logical and Linguistic Aspects of the Lambek Calculus*, Ph.D. thesis, University of Amsterdam, Amsterdam
- Morrill, G., N. Leslie, M. Hepple & G. Barry: 1990, Categorial Deductions and Structural Operations, in G. Barry & G. Morrill (eds.) *Studies in Categorial Grammar*, Edinburgh Working Papers in Cognitive Science, Volume 5, Centre for Cognitive Science, University of Edinburgh, Edinburgh, 1-21
- Pinkham, J.: 1982, *The Formation of Comparative Clauses in French and English*, Indiana University Linguistics Club, Bloomington, Indiana
- Riemsdijk, H. v.: 1978, *A Case Study in Syntactic Markedness*, Ph.D. thesis, University of Amsterdam, Amsterdam
- Stechow, A. v.: 1984, Comparing Semantic Theories of Comparison, *Journal of Semantics* 3, 1-79

Command and Domain Constraints in a Categorical Theory of Binding

Mark Hepple

Cambridge University Computer Laboratory

1 Introduction

This paper presents a categorical account of binding, which was first described in Hepple (1990).¹ The grammatical framework of the account is an extended version of the Lambek calculus. I will here address only cases where personal and reflexive pronouns are bound (i.e. excluding, for example, referential uses of personal pronouns).² The paper focusses particularly on two classes of constraint that are observed to limit the possibilities for binding. These constraints have received a considerable amount of attention in attempts to formulate accounts of binding, and have presented particular problems for categorical accounts. The two classes of constraint that concern us are: *command* constraints, which relate to asymmetries that are observed for which phrase positions may bind which others, and *domain* constraints, which involve requirements on the dominating phrases within which a pronoun's antecedent must either be present or absent.

In what follows, I first present a formulation of the Lambek calculus, before outlining the basis of the account of binding. Then, I show how the Lambek account may be adapted to appropriately characterise constraints on binding, addressing domain and command constraints in turn. The treatment of domain constraints involves extending the calculus with modal operators which may be used to specify 'linguistic boundaries'. The treatment of command constraints is based on adapting the insight of Montague Grammar work, which links grammatical hierarchy to argument order. Finally, the resulting approach is illustrated by applying it to the problem of long-distance reflexivisation in Icelandic, a phenomenon of particular interest because of its unusual locality behaviour.

2 Lambek calculus

In this section, I will introduce the Lambek calculus (Lambek 1958, 1961). I use a natural deduction formulation of the calculus (see Morrill et al. 1990; Barry et al. 1991). Natural deduction proofs proceed from a number of initial assumptions, some of which may be "discharged" or "withdrawn" during the course of the proof. For the product-free Lambek calculus, we require the inference rules shown in (1). The notation of a type with dots above it is used to designate a proof of that type. Assumptions are simply individual types, as licensed by the hypothesis rule. Undischarged assumptions are termed *hypotheses*. Note that in this formulation, each type in a proof is associated with a lambda expression, its *proof term* (shown to the right of a colon). The elimination rule $/E$ states that a proof of A/B and a

proof of B may be combined as shown to construct a proof of A. The introduction rule /I states that given a proof of A, we may discharge some hypothesis B within the body of the proof to construct a proof of A/B. Square brackets are used to indicate a discharged assumption. Note that there is a side condition on this rule, which is required to maintain the constituent order significance of the directional slash.

(1) Hypothesis rule: $A:x$

Elimination rules:

$$\frac{\begin{array}{c} \vdots \\ A/B:f \end{array} \quad \begin{array}{c} \vdots \\ B:x \end{array}}{A:fx}/E$$

$$\frac{\begin{array}{c} \vdots \\ B:x \end{array} \quad \begin{array}{c} \vdots \\ A \backslash B:f \end{array}}{A:fx} \backslash E$$

Introduction rules:

$$\frac{\begin{array}{c} [B:x]^i \\ \vdots \\ A:f \end{array}}{A/B:\lambda x.f}/I^i \quad \text{where B is the right-most undischarged assumption in the proof of A}$$

$$\frac{\begin{array}{c} [B:x]^i \\ \vdots \\ A:f \end{array}}{A \backslash B:\lambda x.f}/I^i \quad \text{where B is the left-most undischarged assumption in the proof of A}$$

To demonstrate that some type combination $X_1, \dots, X_n \Rightarrow X_0$ is possible, a proof of X_0 is given having hypotheses $X_1, \dots, X_n$ (in that order) with zero or more discharged assumptions interspersed amongst them. The proof term for the conclusion of a proof expresses its meaning as a combination of the meanings of the proof's hypotheses, and the inference rules specify how this lambda expression is constructed. Each assumption has a unique variable for its proof term. The elimination and introduction rules correspond to semantic operations of functional application and abstraction, respectively. In general, proof terms are omitted to simplify presentation. The proof of the combination $a/b, b/c \Rightarrow a/c$ ("simple forward composition") in (2) illustrates this approach.

$$(2) \quad \frac{\frac{\frac{a/b:x \quad b/c:y \quad [c:z]^i}{b:(yz)}/E}{a:(x(yz))}/E}{a/c:\lambda z.(x(yz))}/I^i$$

The availability of rules allowing introduction of slash connectives in the Lambek calculus has the consequence that it allows great flexibility in the assignment of types to strings, which in turn allows considerable flexibility in the assignment of syntactic structure to sentences. Such flexibility (in this and certain other categorial approaches) has provided the basis for categorial accounts of a range of phenomena, including extraction and non-constituent coordination. Note that various categorial operations which are treated as primitive in other approaches are theorems of the Lambek calculus, for example, the composition and type-raising operations of Combinatory Categorial Grammar (Steedman, 1987; Szabolcsi, 1987),

Despite its appealing elegance and simplicity, the standard Lambek calculus is

clearly inadequate as a linguistic framework, being unable to handle a broad range of phenomena. Various proposals have been made for dealing with these shortcomings by augmenting the basic calculus with a range of further operators whose logical behaviour is such as to allow description of some aspect of the linguistic data. We shall consider two extensions of the basic calculus before going on to address the treatment of binding. Some further extensions will be introduced in context later on.

The first extension is required in the treatment of extraction. I assume a treatment of extraction (deriving ultimately from proposals by Ades and Steedman, 1982) in which extracted items are given higher order types. For example, a relative pronoun might be $(\text{rel}/(\text{s}/\text{np}))$, a type which seeks a 'sentence lacking a noun phrase', in effect 'abstracting' over the missing NP in the extraction domain. Natural deduction proofs of such examples involve an additional assumption which appears in the place of the missing element, and which is later discharged in a slash introduction inference. However, given the side conditions on these introduction rules (see above), the basic calculus can only handle cases of extraction where the site of the missing element would be peripheral in the overall constituent from which it is missing. This problem can be solved using an idea borrowed from linear logic, where *structural modalities* are employed to allow controlled involvement of structural rules.³ I shall employ an operator Δ , which has the inference rules (3). (The ΔI and ΔE rules give behaviour for Δ somewhat like that of necessity in the modal logic S4. In practice, the rule ΔI will not be required in this paper.) The ΔP permutation rule has the effect of undermining the ordering of Δ -marked types relative to other types in a proof.⁴ It is convenient to use a 'derived' inference rule $(/\Delta)\text{I}$ shown in (4), which is similar to $/\text{I}$, except that it lacks a side condition (since a Δ -marked assumption could always move to right peripheral position within a subproof under some sequence of permutation steps, so as to meet the side condition on $/\text{I}$). A relative pronoun $(\text{rel}/(\text{s}/\Delta\text{np}))$ allows for extraction from non-peripheral positions, as illustrated in (5).

(3) Rules for Δ :

$$\begin{array}{ccc} \frac{\vdots}{\Delta A:x} \Delta\text{E} & \frac{\vdots}{\Delta A:x} \Delta\text{I} & \text{where each hypothesis in} \\ & & \text{the proof of } A \text{ is a } \Delta\text{-type} \\ & & \frac{\vdots \quad \vdots}{B:y \quad A:x} \Delta\text{P} \quad \text{where } A \text{ or } B \\ & & \text{is a } \Delta\text{-type} \end{array}$$

(4) Derived inference rule $(/\Delta)\text{I}$:

$$\frac{\begin{array}{c} [\Delta B:x]^i \\ \vdots \\ A:f \end{array}}{A/\Delta B:\lambda x.f} (/ \Delta)\text{I}^i$$

$$\begin{array}{c}
 (5) \quad \frac{\frac{\text{which}}{\text{rel}/(s/\Delta np)} \quad \frac{\text{Mary}}{np} \quad \frac{\text{gave}}{s \backslash np/pp/np} \quad \frac{\frac{\frac{\text{to}}{pp/np} \quad \frac{\text{Bill}}{np}}{pp}}{[\Delta np]^i \Delta E} /E}{\frac{s \backslash np/pp}{s \backslash np} /E} /E \\
 \frac{s}{s/\Delta np} (/ \Delta)^i \\
 \frac{\text{rel}}{rel} /E
 \end{array}$$

The second extension is involved in the treatment of *locality* phenomena, such as island constraints on extraction and domain constraints on binding. Morrill (1989;1990) suggests that the basic Lambek calculus be augmented with an operator $\Box$ for handling locality constraints, which corresponds in its behaviour to a (S4-like) necessity operator. Hepple (1990) argues that Morrill's proposal is inadequate to deal with the observed complexities of locality phenomena, in that the single $\Box$ operator provides what is in effect a single notion of 'linguistic boundary'. Hepple (1990) develops Morrill's proposal by moving to a *polymodal* system, in which an indefinite number of interrelated modalities (notated by 'labelled boxes' $\Box_a, \Box_b, \dots$, etc) can be defined. A full understanding of this system would require a discussion of the interpretation of types for the Lambek calculus, which is beyond the scope of the present paper (see Hepple (1990) for a more complete exposition). Suffice it to say here that the interpretation of types is handled in terms of sets of strings, which are all subsets of some overall 'language' of strings $\mathcal{L}$. The label within each box names a particular string set (or 'sublanguage') which is used in its definition. Interderivability between distinct boxes arises given the statement of the modal inference rules, shown in (6), and particularly the introduction rule $\Box I$, which refers to relations between defining string sets.⁵

(6) Rules for $\Box_a$:

$$\frac{\vdots}{\Box_a A} \Box_a E \quad \frac{\vdots}{A} \Box_a I \quad \text{where each path to a hypothesis leads to an independent subproof of some modal type } \Box_a X \text{ such that } \mathcal{L}_\alpha \subseteq \mathcal{L}_\beta$$

The polymodal boxes are used to specify linguistic boundaries. In particular, a functor of the form $X/\Box_a Y$ (i.e. which seeks a modal argument) specifies that its argument constituent is potentially an enclosed domain. For example, a sentence complement verb such as *believes* might have a type: $s \backslash np/\Box_a s$ (for some modality $\Box_a$). This approach requires that lexical items must be to some extent modal, to allow modal constituents (such as $\Box_a s$) to be derived from them. For this purpose, I assume a sublanguage $\mathcal{L}_*$, whose modality is notated by the empty box $\Box$, and which is required to be the outermost type-forming operator of all lexical types (i.e. all lexical types are of the form $\Box X$). Since we want lexical types to be able to appear embedded under any modal domain, the sublanguage $\mathcal{L}_*$ is assumed to be a

subset of all the other modality defining sublanguages used. Some possible lexical types within this system are: $\Box np$ (noun phrase), $\Box(s \backslash np / np)$ (transitive verb), and $\Box(s \backslash np / \Box s)$ (sentence embedding verb). The basic idea for handling island constraints with this system is based around the interrelation of the modalities used in specifying modal boundaries and the types of extracted items. This is illustrated by the following example involving relativisation from an embedded clause (which omits the Δ operator to simplify):

$$\begin{array}{c}
 (7) \quad \frac{\frac{\text{which}}{\Box(\text{rel}/(s/\Box np))} \quad \frac{\text{Mary}}{\Box np} \quad \frac{\text{believes}}{\Box(s \backslash np / \Box s)} \quad \frac{\text{John}}{\Box np} \quad \frac{\text{ate}}{\Box(s \backslash np / np)} \quad \frac{[\Box np]^i}{np}}{\frac{\text{rel}/(s/\Box np)}{\Box E} \quad \frac{np}{\Box E} \quad \frac{s \backslash np / \Box s}{\Box E} \quad \frac{np}{\Box E} \quad \frac{s \backslash np / np}{\Box E} \quad \frac{np}{\Box E}}{\frac{s \backslash np}{\Box E}} \\
 \frac{\frac{s}{\Box I}}{\frac{\Box s}{/E}} \\
 \frac{s \backslash np}{/E} \\
 \frac{s}{/I^i} \\
 \frac{s/\Box np}{/E} \\
 \text{rel}
 \end{array}$$

Although this proof has been drawn as running through to completion, it is only in fact correct if the $\Box I$ rule has been correctly applied. In line with the remarks above, we know that $\mathcal{L}_* \subseteq \mathcal{L}_a$ and $\mathcal{L}_* \subseteq \mathcal{L}_b$, and hence the proof is correct just in case $\mathcal{L}_a \subseteq \mathcal{L}_b$. In this way, the grammar writer can specify the sensitivity of processes like movement to particular linguistic boundaries, by the choice of modalities in constructing lexical types. Hepple (1990) discusses this account in more detail in relation to the treatment of island constraints.

3 Binding: the basic method

I will next present the basic method of the account of binding. I treat reflexives and bound pronouns as being semantically identity functions over individuals (i.e. having lambda semantics $\lambda x.x$).⁶ The syntactic type of such pronouns is (in the most basic instance) $np/\Theta np$. This type contains an unfamiliar operator Θ , but if we ignore its presence for the moment, we can see that the syntactic type is also essentially an identity function. This type for bound pronouns serves to introduce an argument position which can be co-bound with some other argument position, and, given the identity semantics, binding the pronoun's argument is equivalent to binding the argument position that the pronoun itself occupies. The operator Θ serves to identify the pronoun's argument as one that is available for binding, and also prevents the pronoun incorrectly taking a lexical NP as argument (e.g. giving **himself John* as a NP). I shall not give any inference rules for Θ here, making it effectively a "dead" operator.⁷ Hence, Θ cannot be introduced or eliminated in

syntax, and so Θ occurrences must originate lexically, with the types of anaphors and bindable pronouns.

Binding is effected by the Binding Interpretation Rule (BIR), shown in (8). The BIR discharges a hypothesis ΘB within the proof (which must correspond to a bindable pronoun's argument since only such pronouns introduce Θ operators), but does not change the proof's conclusion type. In the semantics of the result constituent, the lambda operator binds two instances of the variable v (since v already appears as a subterm of f) and so co-binds the pronoun's argument with the first argument sought by the main functor of the proof. Note that the link between the types of the discharged assumption and the argument sought by the proof's conclusion type (i.e. types B and ΘB) allows a pronoun to determine the type of its binding argument (e.g. a pronoun $np/\Theta np$ requires a NP binder).

$$(8) \quad \text{Binding Interpretation Rule (BIR):} \quad \frac{\begin{array}{c} [\Theta B:x]^i \\ \vdots \\ C:f \end{array}}{C:\lambda x.fx} \text{BIR}^i \quad \text{where } C \text{ is } A \setminus B \text{ or } A/B$$

The BIR's use is illustrated in (9). This proof assigns the meaning (10a), which reduces to (10b) (since $\text{himself}' = \lambda x.x$).

$$(9) \quad \frac{\frac{\frac{\text{John}}{np} \quad \frac{\text{loves}}{s \setminus np/np} \quad \frac{\text{himself}}{np/\Theta np} \quad [\Theta np]^i}{np/E}}{s \setminus np \text{BIR}^i} \text{BIR}^i \quad \frac{s \setminus np}{s \setminus np \setminus E} \text{BIR}^i \quad \frac{s}{s \setminus np \setminus E} \text{BIR}^i$$

- (10) a. $(\lambda v.\text{loves}'(\text{himself}' v) v) \text{john}'$
 b. $(\text{loves}' \text{john}' \text{john}')$

4 Domain constraints on binding

We shall next consider this account in relation to locality constraints on binding. In the case of reflexives, locality constraints typically take the form that the binder of the reflexive must occur inside of some specified domain. Such limitations can be readily addressed within an account that employs the polymodal system for specifying linguistic boundaries described earlier. For example, consider the following attempt to derive the example **Mary_i thinks John loves herself_i*, using anaphor type $\square(np/\Theta np)$:

To complete this proof, the string *John loves herself* must be derived as a modal clause. However, the subproof includes an undischarged assumption $\ominus p$, corresponding to the argument of the reflexive. (Note that to get the intended reading, the BIR would need to apply at a later stage in the proof.) Since this assumption is not $\Box$ -modal, the side condition on the box introduction rule $\Box I$ is not met, and so the proof cannot be continued. Thus, with this reflexive type, a reflexive occurring within a modal domain must also be bound within this domain.

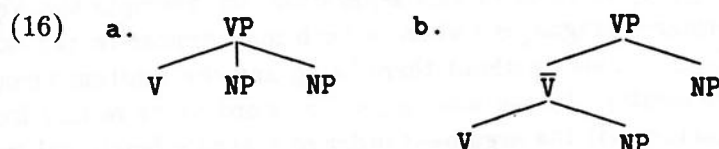
$$(12) \quad \frac{\frac{\frac{\Box(\Box np / \Theta np)}{\Box np / \Theta np} \Theta np}{\Box np / \Theta np} / E}{\frac{\Box np}{np} \Theta E}$$

Consider the case of personal pronouns in syntactically bound uses, as for example in *Every girl_i thinks John loves her_i*. Personal pronouns are not required to be bound *within* any particular domain. In the present account, such behaviour can be allowed by assigning them a type such as $\Box(\Box_{np}/\Theta_{np})$, where the value subtype np is marked with the modality $\Box$ (whose defining language $\mathcal{L}_\emptyset$ is assumed to be a subset of all the other defining languages used). The presence of this $\Box$ allows pronouns bearing this type to be syntactically bound from outside of *any* modal domain. A proof for the VP of *Every girl_i thinks John loves her_i* is shown in nextexp). This proof assigns a reading which simplifies to the expression in (14).

where heads, though not strictly functional, display explicit ordered subcategorisation which encodes obliqueness. Here, an *o-command* relation is employed, which is similar to F-command but stated purely in terms of order of subcategorisation and obliqueness, rather than function-argument structure.

Chierchia (1988) notes one advantage of F-command is that it correctly predicts the possibilities (15) for binding between the objects of double object constructions (since the first object is taken to be less oblique than the second), whereas c-command fails to predict the ungrammaticality of (15b) under the widely assumed structure shown in (16a). Barss and Lasnik (1986) also note this problem, and consider in addition the alternative structure in (15b), which fares even worse in predicting precisely the converse of the observed grammaticality/ungrammaticality for the examples in (15).

- (15) a. Mary showed John_i himself_i
 b. *Mary showed himself_i John_i



As a basis for handling command constraints within our Lambek framework, we might also allow argument order to encode relative obliqueness. This move, however, presents some difficulties. Firstly, a problem arises because the Lambek calculus is a highly flexible categorial system, and allows type transformations which may radically alter the structure of types, for example $x/y \backslash z \Rightarrow x \backslash z/y$ where two counterdirectional arguments of a function are reordered. Clearly, such flexibility threatens the use of argument order to encode relative obliqueness. This problem can be avoided by adopting two additional connectives, ϕ and $\flat$, which have elimination rules just like those for $/$ and $\backslash$, but have *no introduction rules*. This lack of introduction rules means that occurrences of these connectives cannot be created in syntax, and gives an asymmetric relation between these new connectives and their corresponding standard connectives, e.g. it is possible to 'convert' a ϕ into a $/$, as in $x\phi y \Rightarrow x/y$, but the converse $*(x/y \Rightarrow x\phi y)$ is not possible. Hence, occurrences of these connectives must originate in the lexicon and functions constructed with them must exhibit lexically given argument order. These connectives are used to specify *primitive subcategorisation*, i.e. lexically given functional structure, where argument order encodes obliqueness.

Secondly, problems arise with respect to the treatment of word order. The Lambek calculus, like various other categorial systems, allows only the use of concatenation for combining the strings associated with types that are combined. When combined with an assumption that argument order should encode relative obliqueness, this characteristic gives rise to some incorrect predictions about word order. For example, consider the sentence *John gave a book to Mary*, where *to Mary* is

the verb's most oblique complement, and hence should be the *first* argument of the verb's functional type. But then, given the use of just concatenation for combining strings, *to Mary* should appear next to *gave* in the observed word order. These problems are avoided in Montague Grammar by the use of non-concatenative operations for combining strings, most importantly *wrapping*.

Various 'wrap simulation' methods have been proposed within 'concatenative' categorial frameworks as a means for allowing the Montagovian treatment of grammatical relations to be adopted (e.g. Szabolcsi 1987; Jacobson 1987; Kang 1988). Jacobson (1987) suggests a method which incorporates a view that standard English word order involves verb movement. I will here outline a Lambek 'wrap simulation' approach, proposed in Hepple (1990), which also incorporates this idea. A role for verb movement in English word order has also been suggested in some non-categorial approaches (e.g. Larson, 1988; Koster, 1988), in part to avoid problems arising in the cases for which Montague Grammar uses wrapping.

The account of word order to be described factors apart the specification of a head's position from the specification of the order of its complements. Certain phenomena suggest the appropriateness of this separation, for example the Verb Second behaviour of Germanic languages (where a verb may appear in radically different positions in different clauses without there being any concomitant change in the order of its complements). In the new approach, word order results from the interaction of three factors: (i) the argument order of a head's functional type (corresponding to obliqueness), (ii) the directionality of each argument, (iii) a lexical process of *Head Location*, which determines the position of the head relative to its complements. Factors (i) and (ii) together determine the linear order of a head's complements, and would also determine the position of the head were it not for the involvement of factor (iii).

This approach requires that we adopt a non-standard view of the categorial lexicon, in which lexical type assignments are built up over several stages, with only the 'end product' of this process being available to syntax. This notion of 'stages in construction' is handled by allowing the lexicon to consist of a number of distinct 'compartments', where the types assigned to any word may differ in different compartments. A designated 'final' compartment specifies the type assignments that are made available to syntax, all other compartments being invisible to syntax. Type assignments in different compartments are related by lexical rules. Certain compartments specify *initial* type assignments (i.e. ones not resulting from the operation of a lexical rule).

The process of Head Location is effected by a lexical rule (whose input and output types I shall refer to as *prelocation* and *located* head types, respectively), and employs the general approach to extraction outlined above. A head having prelocation type H can be allowed to occur left peripherally in phrases of type X by giving it a lexical type $X/(X/\Delta H)$. This type-change is essentially an instance of type-raising, although not a theorem of the syntactic calculus. The Head Location rule for English verbs is shown in (17), where vp abbreviates possible VP types of English (i.e. types $s\downarrow A$, where A is any subject type), and $vp\phi$ stands for any type

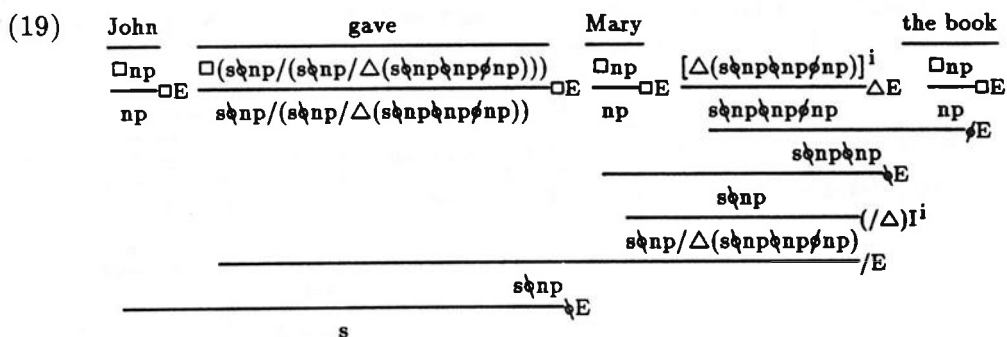
vp or function into vp. The located types produced by the rule only allow the verb to appear to the left of its own VP projection, not those of any dominating verbs, because, firstly, the 'movement' effected by the rule is bounded, and secondly, the verb must appear left peripherally to some VP projection bearing the same features for tense etc as the verb's prelocation type (since the three occurrences of vp in the output type are identical to the vp subtype of the input). These conditions can only be satisfied when the verb appears beside its own VP projection.

(17) English Verb Location:

$$vp\phi:f \Rightarrow vp/(vp/\Delta(vp\phi)): \lambda g.gf$$

Some verbs are shown in (18) with a prelocation type and the final lexical type assignment that it gives rise to following Head Movement and a final step adding an outermost $\square$. In the prelocation types, first objects have been assigned leftward directionality, and second objects rightward directionality. See Hepple (1990) for discussion of how directionality determines complement order in this approach, and how these directionalities for first and second arguments are arrived at. The derivation (19) illustrates this view of word order.

- (18)
- | | | |
|------|--------------------------|---|
| eat | $s\phi np\phi np$ | $\square(s\phi np/(s\phi np/\Delta(s\phi np\phi np)))$ |
| gave | $s\phi np\phi np\phi np$ | $\square(s\phi np/(s\phi np/\Delta(s\phi np\phi np\phi np)))$ |
| put | $s\phi np\phi np\phi pp$ | $\square(s\phi np/(s\phi np/\Delta(s\phi np\phi np\phi pp)))$ |



The approach to handling word order and subcategorisation just outlined has the characteristic that there is a stage in the lexicon at which the type assigned to a verb (or other head) is one whose argument order encodes relative obliqueness. Hepple (1990) shows how this characteristic allows relation changing phenomena to be handled by lexical rules, modifying proposals from Montague Grammar (e.g. Bach, 1980; Dowty, 1982). The final lexical type assignments of this approach, however, do not have an argument order that corresponds to relative obliqueness, being instead simply raised types produced via Head Location. Note though that the prelocated type of a head appears as a subtype of the final lexical type assignment that derives from it, and so the information of relative obliqueness encoded by the prelocation type is at least present. As we shall see next, in relation to the treatment of binding, this obliqueness information is still available from the final

(20) Binding Interpretation Rule (BIR):
$$\frac{\begin{array}{c} [\ominus B : x]^i \\ \vdots \\ C : f \end{array}}{C : \lambda x. fx} \text{BIR}^i$$
 where C is $A \dot{\wedge} B$ or $A \dot{\vee} B$

The approach is illustrated in (21), which shows a derivation for the VP *showed Mary_i herself_i*, with binding between first and second objects. No proof is possible for the converse binding **showed herself_i Mary_i*.

Examining this proof, it is possible to see why the obliqueness information of the prelocation type can still be exploited after Head Location. Observe that the prelocation type of the verb *does* participate within the proof, appearing as an additional hypothesis that is later discharged.

6 Long distance reflexivisation in Icelandic

In this section, we consider the phenomenon of long distance reflexivisation in Icelandic, which is of interest because of its unusual locality behaviour. I will show that this unusual behaviour can be straightforwardly handled within the framework described above, most importantly by exploiting the polymodal treatment of domain constraints.

The core data concerning Icelandic long distance reflexivisation (LDR) is described by Thrainsson (1976). Icelandic reflexivisation is in general clause bounded, but under certain conditions unbounded or long distance reflexivisation is possible. Firstly, Icelandic LDR is subject oriented, i.e. only subject phrases are possible binders. Secondly, the possibility of LDR is dependent upon the presence of subjunctive mood. LDR arises only when the reflexive appears in a complement possessing subjunctive mood, as shown by the following pair of sentences:

- (22) a. *Jón_i veit að María elskar sig_i
 Jon_i knows that Maria loves(I) REFL_i
 b. Jón_i segir að María elski sig_i
 Jon_i says that Maria loves(S) REFL_i

(*I* and *S* in the glosses of these examples indicate a verb in indicative or subjunctive mood, respectively.) In (22a), where the embedded clause has indicative mood, the reflexive cannot be bound by the matrix subject. In (22b), the embedded clause is in subjunctive mood, and long binding by the matrix subject is allowed. In example (23), any of the several subjects may act as antecedent, showing that binding may be made across any number of clause boundaries provided that the intervening clauses are all in subjunctive mood. Interestingly, the clause whose subject acts as binder does not itself need to be in subjunctive mood, only those embedded beneath it.

- (23) Jón_i segir að María_j telji að Haraldur_k vilji að Billi_l heimsæki sig_{i/j/k/l}
 Jon says that Maria believes(S) that Harold wants(S) that Billy visits(S) REFL

Subjunctive mood in Icelandic is introduced by particular items, such as subordinating conjunctions (e.g. *nema* 'unless' and *ef* 'if') and nonfactive verbs. For example, a nonfactive verb such as *segir* ('say') may introduce subjunctive mood, whereas a factive verb like *vita* ('know') may not, the latter generally taking an indicative complement. However, what has been called a *domino effect* is observed (Thrainsson, 1976) in which subjunctive mood introduced onto some complement may be propagated down through embedded complements, where subjunctive mood might not otherwise appear. In (23), the verb *segir* introduces the subjunctive mood of its complement, from whence subjunctive 'trickles down' to the clauses embedded beneath. I shall assume a simple lexical treatment of this 'domino effect', which is based on a generalisation about the distribution of the [+subjunctive] feature on

lexical types, which can be stated as the following 'licensing condition', governing type assignment in the lexicon.

(24) Domino Effect Condition:

Any functor that is itself marked as [+subjunctive] may specify [+subjunctive] on any of its complements.

This condition means that there is no limitation on the 'downward transmission' of the [+subjunctive] feature, with the consequence that the [subjunctive] feature may be specified on many constituents where it has no morphological realisation.

Let us next consider how the core facts about Icelandic LDR may be handled in the approach outlined above. The subject orientation characteristic may be handled by adopting a modified version of the BIR which is further restricted as to the functional type of proofs to which it may apply, e.g. if the BIR could apply only to proofs of VP types, then only clausal subjects would be possible binders. In fact, the relevant notion of 'subject' for Icelandic LDR is broader than this, including possessive NPs. The following version of the BIR allows for both of these kinds of antecedent subject (recall that the type of the binding argument is determined by the type assigned to the reflexive, and so does not need to be limited by the binding rule).

$$(25) \text{ Binding Interpretation Rule (BIR): } \frac{\begin{array}{c} \vdots \\ C:f \end{array}}{C:\lambda x.fx} \text{BIR}^i \quad \text{where } C \text{ is } A \&B \text{ or } A \&\&B, A \in \{s, np\}$$

I assume a type $\Box(\Box np / \ominus np)$ for long distance reflexives. Note that the value subtype of this type is marked with a modality $\Box$, whose presence, as we saw earlier, will make syntactic binding insensitive to the presence of any modal boundaries with a modality $\Box$ such that $\mathcal{L}_u \subseteq \mathcal{L}_\alpha$. The core data of can consequently be handled simply by including the condition (26) to constrain the possibilities for lexical type assignment.

(26) Subjunctive Boundaries Condition:

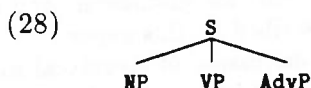
where a lexical functor seeks a complement of type $\Box X$, then:
 $\mathcal{L}_u \subseteq \mathcal{L}_\alpha$ holds *if and only if* the type X is marked [+subjunctive]

This condition has the consequence that modal boundaries marked on [+subjunctive] complements do not present barriers to binding of LD reflexives, and, since the requirement is biconditional, binding of LD reflexives is blocked by modal boundaries marked on any complement that is not [+subjunctive]. The condition is insensitive to the origin of the [+subjunctive] marking, and so its operation interacts appropriately with that of the Domino Effect condition to allow long binding even where subjunctive mood is present only due to the domino effect. Note that because the Subjunctive Boundaries Condition refers only to the featural status of the argument

of the functor, rather than the functor itself, the approach accounts quite straightforwardly for the perhaps otherwise puzzling characteristic of Icelandic LDR, that the mood of the clause in which the *antecedent* appears is irrelevant to the possibility of LDR.

We shall next briefly discuss a possible counterargument to this account. The above treatment of Icelandic LDR does not differ in its essential view of what is important in determining binding possibilities from the account as it was developed in relation to English, i.e. an antecedent must *o-command* the anaphor, and no 'blocking' boundaries may intervene between the two. Maling (1984) argues that *c-command* is irrelevant for Icelandic LDR, but these arguments apply also against *o-command*, and so warrant our attention. Maling's principal counterexamples (first noted by Thrainsson, 1976) involve the fact that a matrix subject may not bind a reflexive occurring in certain adverbial clauses, e.g. (27a,b), where subjunctive is present and where *c-command* holds under the structure that Maling assumes, shown in (28).

- (27) a. *Jón; yrði glaður ef Sigga byði sér;
 John; would-be(S) glad if Sigga invited(S) REFL;
 b. *Jón; kemur ekki nema Sigga bjóði sér;
 John; comes(I) not unless Sigga invites(S) REFL;



Clearly, the status of such cases as counterexamples rests on the correctness of the structure that Maling assumes. Approaching these examples from a categorial perspective, where a central assumption is that syntactic structure is essentially just a refinement of semantic 'structure', we would by default expect a type such as *s\s* for an adverbial clause like *nema Sigga bjóði sér*; (*unless Sigga invites(S) REFL*) since it semantically relates *propositions* rather than *predicates*. With this type for the adverbial, *o-command* does not hold and the impossibility of binding is expected. Note that binding is possible in cases involving "phrasal" rather than "sentential" adverbials, e.g. (29). Again on a semantic basis, we would expect a 'predicate modifying' type such as *(s\NP)\(s\NP)* for the adverbial, and given this type the subject argument *does o-command* the anaphor inside the adverbial, correctly predicting the grammaticality of binding.

- (29) Jón; kemur ekki án konu sinnar; /*hans;
 Joh comes not without wife(G) REFL; /*his[-refl];
 'John won't come without his wife'

Let me close with a few remarks about the significance of this account of Icelandic LDR. Because of its unusual locality behaviour, and because *c-command* has been argued irrelevant to it, Iceland LDR has been seen as belonging to a separate

subclass of cases in which certain reflexives may show discourse-governed coreferential behaviour. The present account not only demonstrates that it is possible to assimilate the treatment of these LD reflexives to the treatment of ordinary bound reflexives and pronouns, but also, by its combination of categorial structures with a command condition, serves to explain some otherwise mysterious aspects of the data, e.g. the asymmetry between phrasal and sentential adverbs. In so far as it successfully demonstrates that we must acknowledge the existence of non-local syntactic binding of reflexives, the account suggests that the demarcation between the possibilities for bound and coreferential uses of reflexives cannot be handled solely in terms of a notion of 'codependent' or 'coargument', as has been suggested in various recent proposals.

Notes

1. I would like to thank the following people for discussion of the ideas presented in this paper: Guy Barry, Elisabet Engdahl, Glyn Morrill, Mark Steedman, Anna Szabolcsi. The research was largely carried out at the Centre for Cognitive Science, Edinburgh, with the support of ESRC Research Studentship C00428722003, and this paper was prepared at the Cambridge University Computer Laboratory.
2. Cases in which a pronoun is not bound are familiar for personal pronouns, but perhaps less familiar for reflexives. Some recent accounts (e.g. Pollard and Sag, 1990) have emphasised the need to recognise a limited subclass of cases in which reflexives are not bound, but rather show coreferential behaviour which is subject to various discourse/semantic restrictions. These cases will not be discussed here, but see Hepple (1990) for discussion, and some proposals for handling these cases within the framework described by this paper.
3. See Morrill *et al* (1990) and Barry *et al*, (1991) for discussion of structural modalities, and proposals for their use in handling phenomena involving word order variation, optionality and iteration.
4. Note that the permutation rule ΔP is unusual in being *multiply conclusioned*. It is standard in Lambek work to assume that in all permissible derivations, a *single* conclusion type is derived from one or more premises. To maintain this restriction, we may stipulate that any *complete* proof must be single-conclusioned. It follows that a permutation inference may not be the final inference in any complete proof.
5. The $\Box I$ rule might seem to have an unnecessarily complicated side condition, in referring to 'independent subproofs'. To justify this condition, first consider a system with only a single necessity operator $\Box$. An intuitively sensible side condition for $\Box I$ would require that "every hypothesis of the proof is a mod-type", i.e. intuitively, if a conclusion is drawn purely on the basis of necessary assumptions, then we may infer that the conclusion is necessary. Such a rule turns out to be equivalent to one with a side condition that "each path to a hypothesis of the input proof leads to an independent subproof of a $\Box$ -type". Note firstly that a single type constitutes an independent subproof, and so the complex version of the rule can be applied to any proof that the simpler version of the rule can be applied to. Conversely, consider a case where a proof has a single non-modal hypothesis, which does occur within an independent subproof of a $\Box$ -type. If that subproof is excised from the overall proof, leaving its modal end-type as simply a hypothesis, this will yield a well-formed modified proof, one to which the simpler version of the $\Box I$ rule could apply. Finally, the excised subproof may be reinserted, yielding the same result as the more complex version of

the rule had applied to the original proof. This process generalises to cases where a proof has more than one independent subproof. By analogy, it should be clear that the $\Box I$ in (6) is equivalent to one with the simpler side condition that "each hypothesis is some modal type $\Box X$ such that $\mathcal{L}_\alpha \subseteq \mathcal{L}_\beta$ ".

6. The characteristic of treating pronouns as identity functions gives a link between the present account of binding and that of Jacobson (this volume). In fact, the two accounts have a considerable amount in common, although this is to some extent obscured by differences of formalism. Indeed, I believe that many of the insights of Jacobson's proposals can be readily incorporated into the present approach.

7. In Hepple (1990), Θ is a permutor structural modality, though having different modal behaviour from $\Box$. See Hepple (1990) for discussion of the advantages for the account that follow from the additional flexibility that the presence of these inference rules allows.

8. More specifically, Dowty (1980) suggests that disjoint reference arises from a "neo-Gricean conversational principle" which we might paraphrase thus: "Where two equally simple expressions A and B are available to express a meaning X, but where A unambiguously means X whilst B is ambiguous between meanings X and Y, then B should be reserved for expressing Y."

References

- Ades, A.E. and Steedman, M.J. 1982.: 'On the order of words.' *Linguistics and Philosophy*, 4. 517-558.
- Bach, E. 1980.: 'In defense of passive.' *Linguistics and Philosophy* 3, 297-341.
- Bach, E. and Partee, B.H. 1980.: 'Anaphora and semantic structure.' In K.J. Kreiman, and A.E. Ojeda (Eds), *Papers from the Parasession on Pronouns and Anaphora*, Chicago Linguistic Society.
- Barry, G., Hepple, M., Leslie, N. and Morrill, G. 1991.: 'Proof Figures and Structural Operators for Categorical Grammar', in: *Proceedings of the European ACL*, Berlin.
- Barss, A. and Lasnik, H. 1986.: 'A note on Anaphora and Double Objects.' *Linguistic Inquiry*, 17. 47-58.
- Chierchia, G. 1988.: 'Aspects of a categorial theory of binding.' In Oehrle, R., Bach, E. and Wheeler, D. (Eds), *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht. 125-151.
- Dowty, D. 1980. 'Comments on the Paper by Bach and Partee.' In K.J. Kreiman, and A.E. Ojeda (Eds), *Papers from the Parasession on Pronouns and Anaphora*, Chicago Linguistic Society.
- Dowty, D. 1982.: 'Grammatical relations and Montague grammar.' In P. Jacobson and G.K. Pullum (Eds), *The Nature of Syntactic Representation*, D. Reidel, Dordrecht, 79-130.
- Hepple, M. 1990.: *The Grammar and Processing of Order and Dependency: a categorial approach*. Unpublished Ph.D. thesis. Centre for Cognitive Science, University of Edinburgh.
- Jacobson, P. 1987.: 'Phrase Structure, Grammatical Relations and Discontinuous Constituents.' In Huck, G.J. and Ojeda, A.E. (Eds), *Syntax and Semantics*, 20: *Discontinuous Constituency*, Academic Press, New York. 27-69.

- Jacobson, P. *this volume*: 'Bach-Peters Sentences in a Variable-Free Semantics'.
- Kang, B. 1988.: *Functional Inheritance, Anaphora and Semantic Interpretation in a Generalized Categorical Grammar*, Ph.D. Thesis, Brown University, Providence RI.
- Koster, J. 1988.: 'The residual SOV structure of English.' *Groningen Papers in Theoretical Linguistics*, TENK 5, University of Groningen.
- Lambek, J. 1958.: 'The mathematics of sentence structure.' *American Mathematical Monthly* 65. 154-170.
- Lambek, J. 1961.: 'On the calculus of syntactic types.' In *Structure of Language and its Mathematical Aspects*, Proceedings of the Symposia in Applied Mathematics XII, American Mathematical Society, Providence.
- Larson, R. 1988.: 'On the double object construction.' *Linguistic Inquiry* 19. 335-391.
- Maling, J. 1984.: 'Non clause bounded reflexives in Modern Icelandic', *Linguistics and Philosophy*.
- Moortgat, M. 1990.: 'The logic of discontinuous type constructors.' In *Proceedings of the Symposium on Discontinuous Constituency*, Institute for Language Technology and Information, University of Tilburg.
- Morrill, G. 1989.: 'Intensionality, boundedness, and modal logic.' Research Paper EUCCS/RP-32, Centre for Cognitive Science, University of Edinburgh.
- Morrill, G. 1990.: 'Intensionality and boundedness.' *Linguistics and Philosophy*.
- Morrill, G., Leslie, N., Hepple, M. and Barry, G. 1990.: 'Categorial deductions and structural operations.' In Barry, G. and Morrill, G. (Eds), *Studies in Categorial Grammar*. Edinburgh Working Papers in Cognitive Science, Volume 5. Centre for Cognitive Science, University of Edinburgh.
- Pollard, C. and Sag, I.A. 1987.: *An Information-Based Approach to Syntax and Semantics: Volume 1 Fundamentals*. CSLI Lecture Notes, Number 13, Center for the Study of Language and Information, Stanford.
- Pollard, C. and Sag, I.A. 1990.: 'Anaphors in English and the Scope of Binding Theory.' Unpublished manuscript, Stanford University, CA.
- Reinhart, T. 1983. 'Coreference and Bound Anaphora: A Restatement of the Anaphora Questions.' *Linguistics and Philosophy* 6. 47-88.
- Steedman, M.J. 1987.: 'Combinatory Grammars and Parasitic Gaps.' *Natural Language and Linguistic Theory*, 5:3. 403-439.
- Szabolcsi, A. 1987.: 'Bound variables in syntax (are there any?)' In Groenendijk, J., Stokhof, M. and Veltman, F. (Eds), *Sixth Amsterdam Colloquium*, Institute for Language, Logic and Information, University of Amsterdam.
- Thrainsson, H. 1976.: 'Reflexives and Subjunctives in Icelandic.' *NELS*, 6. 225-239.

Belief Dependence, Revision and Persistence

Zhisheng Huang[†] and Peter van Emde Boas^{*}

^{*}ILLC; Department of Mathematics and Computer Science

[†]Center for Computer Science in Organization and Management(CCSOM)

University of Amsterdam

(Extended Abstract)

Abstract

A rational agent, when confronted with information inconsistent with his beliefs will either revise his beliefs in order to incorporate this new information, or persist in his beliefs and reject this information. A less rational agent even may revise his beliefs without there being a real conflict with the new information. In this paper, we propose a rational balance strategy between belief revision and belief persistence in a multiple agent environment. Our proposal is based on the framework of logic of belief dependence. The proposed strategy, called the *confidence priority strategy*, has the impact that the more confidently a formula φ is believed by an informant, the more credibility the belief is. The strategy is plausible for cooperative multiple agent environments. The distinct roles played by the agents are determined in order to capture an intuitive understanding of the strategy.

1 Introduction

Belief revision is one of major problems for belief management techniques and knowledge representation systems [3, 4, 10, 11]. *Belief revision* refers to the process in which a new sentence that is inconsistent with an agent's beliefs or knowledge is added. *Belief persistence* refers to the process in which an agent's belief or knowledge is preserved when confronted with new information. Belief revision and belief persistence are very important features for flexible and intelligent knowledge based systems. However, as far as we know, there exists no existing formalism or approach to study how to balance rationally between those two important processes.

We are looking for an intuitive approach by which the choice between belief revision and belief persistence can be determined by artificial rational agents. In other words, we are looking for a formalism where one can mechanically compute, given a description of the state of the world whether an agent will revise his beliefs or persist when confronted with new incompatible information. In reality the problem seems to largely depend on pragmatics and psychological notions such as intention, goal, and attitudes of agents.

In this paper we present a rational formalism which makes the choice between belief revision and belief persistence computable. Our proposal uses the notions and formalisms of logic of belief dependence as introduced by the first author at the seventh Amsterdam Colloquium [6].

2 Belief Dependence and Belief Revision

2.1 Logic for Belief Dependence

In multiple agent environments, it is frequently beneficial to enable agents to communicate their knowledge or beliefs among each other. Under such circumstances some agents may rely on others for their beliefs or knowledge. We called the phenomenon *belief dependence*. In [6, 8], a formal theory for belief dependence in a multiple agent environment is described, which can serve as a foundation for understanding rational behavior of artificial agents. In this subsection, we give an overview of the main notions in this approach.

Our logic involves in the first place the general notions of knowledge and belief, which are the equivalents of those notions in epistemic and doxastic logic. In our logic for belief dependence, we generally use $L_i\varphi$ to represent the fact that agent i knows or believes the formula φ . As is well known, the modal operator L represents an epistemic operator, when the logic is an S5 system, whereas L is a doxastic operator if the logic is a weak S5 system.

There exists a second important notion used for reasoning about dependent knowledge and beliefs; this notion is called the *dependent operator*, or alternatively *rely-on relation*, and it is denoted by $D_{i,j}$. Intuitively, we can give $D_{i,j}\varphi$ a number of different interpretations: "agent i relies on agent j about the formula φ ", "agent i depends on agent j about believing φ ", or even more specifically, "agent j is the credible advisor of agent i about φ ".

In the communication of knowledge and belief among agents, agents do not necessarily view knowledge and belief accepted from other agents as their own knowledge, even though they may originally have asked for such information. In terms of cognitive psychology, these beliefs are compartmentalized [12]. In logics for belief dependence, we therefore introduce a *compartment operator*, or alternatively called a *sub-belief operator*, written $L_{i,j}$. Intuitively, $L_{i,j}\varphi$ can be read "agent i believes φ due to agent j ". From the viewpoint of *minds society*, $L_{i,j}\varphi$ can be more intuitively interpreted as "agent i believes or knows φ on the mind frame indexed j ". Also, as argued for in [8], we claim that an appropriate procedure for formalizing information assimilation should involve two phases: *compartmentalization* and *incorporation* of information. Compartmentalized information consists of those fragments of information which are accepted and remembered as isolated beliefs and which are treated somewhat different from the beliefs that are completely believed, whereas incorporated information consists of the beliefs that are completely believed by the agents. In the logic for belief dependence, compartmentalized information is modeled by sub-beliefs $L_{i,j}\varphi$ for agent i , whereas incorporated information corresponds to general beliefs of agent i , namely, $L_i\varphi$. Note that in this interpretation the law that

$L_i\varphi \equiv L_{i,j}\varphi \wedge D_{i,j}\varphi$ is not universally valid.

In [8], several semantic models for logics of belief dependence are presented. In this paper, we select the **D**-model as the basic model. Formally, a *belief dependence D-model* is a tuple $M = (S, \pi, \mathcal{L}, \mathcal{D})$, where S is a set of states, $\pi(s, \cdot)$ is a truth assignment for each state $s \in S$, $\mathcal{L} : \mathbf{A}_n \rightarrow 2^{S \times S}$ is a collection of n binary accessibility relations on S , and $D : \mathbf{A}_n \times \mathbf{A}_n \times S \rightarrow 2^{\mathbf{L}_D}$ is the mapping describing the belief dependence: $D(i, j, s)$ is the set of formulas φ for which agent i depends on agent j in state s . Here $\mathbf{A}_n$ is a set of agent names, and $\mathbf{L}_D$ is the formal language for the logic of belief dependence.

2.2 Belief Revision

There exist three kinds of update operations used for belief maintenance [3, 4]. They are: *Expansion*, *Revision*, and *Contraction*.

Expansion: A new proposition φ is added to a given knowledge set K . Formally, the knowledge set that results from expanding K by a proposition φ is denoted $K + \varphi$.

Revision: A new proposition which is inconsistent with a knowledge set K is added, but in order to keep the resulting set consistent some of the old propositions in K must be removed. The resulting of revising K by a proposition φ is denoted $K \dot{+} \varphi$.

Contraction: A proposition in K is retracted without adding any new proposition. The resulting knowledge set of contracting K with respect to the formula φ is denoted $K \dot{-} \varphi$.

Suppose the added proposition is φ , the knowledge set K , and the base of the knowledge set be B . A possible definition for the expansion operation reads:

$$Cn(B) + \varphi \stackrel{\text{def}}{=} Cn(Cn(B) \cup \{\varphi\}) = Cn(B \cup \{\varphi\})$$

where Cn is the *consequence operation*, a closure operator which maps sets of propositions to set of propositions and which has the following properties:

inclusion $A \subseteq Cn(A)$

idempotency $Cn(A) = Cn(Cn(A))$

monotonicity $Cn(A) \subseteq Cn(B)$ whenever $A \subseteq B$

For revision and contraction, the following condition, called *Levi Identity*, is generally required to be satisfied:

$$K \dot{+} \varphi = Cn((K \dot{-} \neg \varphi) \cup \{\varphi\})$$

Although the the revision and contraction operations to a large extent are a matter of pragmatics, and seem to be outside of the scope of logical analysis, there have been proposed a number of intuitive plausible constraints in the work of Alchourrón, Gärdenfors, and Makinson; see [3, 4]. Gärdenfors presents a set of constraints on the contraction operator which are called the *Gärdenfors Postulates*:

Closure $A \dot{-} \varphi$ is a theory

Inclusion $A \dot{-} \varphi \subseteq A$

Vacuity If $\varphi \notin A$, then $A \dot{-} \varphi = A$

Success If $\varphi \notin Cn(\emptyset)$, then $\varphi \notin A \dot{-} \varphi$

Preservation If $Cn(\varphi) = Cn(\psi)$, then $A \dot{-} \varphi = A \dot{-} \psi$

Recovery $A \subseteq Cn((A \dot{-} \varphi) \cup \{\varphi\})$

$$(A \dot{-} \varphi) \cap (A \dot{-} \psi) \subseteq A \dot{-} (\varphi \wedge \psi)$$

If $\varphi \notin A \dot{-} (\varphi \wedge \psi)$, then $A \dot{-} (\varphi \wedge \psi) \subseteq A \dot{-} \varphi$.

3 Belief Maintenance in a Logic of Belief Dependence

As argued for in [8], we believe that an appropriate procedure for formalizing information assimilation should pass two phases: compartmentalization and incorporation of information.

For a multiple agent environment, we assume that some primitive rely-on relations about some propositions among those agents have been determined at the metalevel. We call this assumption the *initial role-knowledge assumption*. We believe that the assumption is appropriate and intuitive, because, in a multiple agent environment, some agents have to possess a minimal amount of knowledge about some other agents, in order to guarantee that they will start to communicate at all. Also, in a reliable communication network, assuming that agents are honest, no-doubt and something more [8], primitive rely-on relations often collapse into primitive communication relations, and this turns them into observable entities.

In [8], the formalism describing the first phase of information assimilation is described. This paper involves the problem of determining how the complete sub-belief and the complete rely-on relations can be captured, using the primitive rely-on relations. This paper also introduces some role-appraisal axioms such as "fool believer", and "stubborn believer".

In this paper, we consider the second phase of information assimilation in a multiple agent environment. Based on the role-appraisal information mentioned above, it turns out to be possible to determine some rational belief maintenance strategies, which enable us to figure out whether and how compartmentalized beliefs can be assimilated into the incorporated beliefs for the agents. In other words, this

role-appraisal information yields rational strategies for agents which determine how and whether their beliefs are revised or persisted.

More formally the process can be described as follows: if an agent i relies on another agent j about believing formula φ , and agent j believes φ , then agent i will initiate the process which eventually might result into accepting the belief φ . However, in the first stage of assimilation of knowledge and beliefs, agent i only accepts φ as a sub-belief $L_{i,j}\varphi \equiv D_{i,j}\varphi \wedge L_j\varphi$. Agent i does not necessarily accept the belief as his own incorporated belief.

In the second stage of assimilation of knowledge and belief agent i is supposed to change his compartmentalized belief into incorporated belief, i.e., change $L_{i,j}\varphi$ into $L_i\varphi$. However, if we just transform the sub-belief in this way for any agent and any formula φ , then we will find that the resulting beliefs may turn out to be inconsistent. In order to avoid such conflicts, some old beliefs of agent i should be removed. Alternatively, the agent may as well reject the new information in order to avoid the inconsistency. Thus, under those circumstances, we must use the belief update operations revision $\dot{+}$ and contraction $\dot{-}$ to describe the process precisely, and we will need further information in order to decide which operator will be invoked under which circumstances.

For a further formalization we need some further notations. If K is a knowledge set, we define $L_{i,j}^-(K) \stackrel{def}{=} \{\psi | L_{i,j}\psi \in K\}$, denoting the set of agent i 's sub-belief indexed j . Moreover, $L_i^-(K) \stackrel{def}{=} \{\varphi | L_i\varphi \in K\}$, denotes agent i 's belief set. The belief maintenance operator is denoted by Δ .

Formally, a *belief maintenance model*, or *BMM*, is an ordered pair $\langle K, \Delta \rangle$ where K is a collection of belief sets and $\Delta : K \times \text{Sent}(L) \rightarrow K$ is a function assigning a maintenance operation $\Delta(K, A)$ to any belief set $K \in K$ and any L -sentence A . We shall write $K\Delta\varphi$ as an alternative representation for $\Delta(K, \varphi)$. In particular, $L_i^-(K)\Delta L_{i,j}\varphi$ denotes the belief maintenance result on the formula set $L_i^-(K)$ upon confrontation with $L_{i,j}\varphi$.

In order to deal with different kinds of belief conflicts, we will use the following five types of update strategies¹:

- | | |
|------------------------|--|
| (Positive-revision) | $L_i^-(K)\Delta L_{i,j}\varphi = L_i^-(K)\dot{+}\varphi$ |
| (Negative-revision) | $L_i^-(K)\Delta L_{i,j}\varphi = L_i^-(K)\dot{+}\neg\varphi$ |
| (Persistence) | $L_i^-(K)\Delta L_{i,j}\varphi = L_i^-(K)$ |
| (Positive-contraction) | $L_i^-(K)\Delta L_{i,j}\varphi = L_i^-(K)\dot{-}\varphi$ |
| (Negative-contraction) | $L_i^-(K)\Delta L_{i,j}\varphi = L_i^-(K)\dot{-}\neg\varphi$ |

The update strategies of negative-revision and positive-contraction seem to be rather counterintuitive at first sight. However, they describe the behavior of an agent i which doesn't trust his informant j at all; if j believes something this is taken to be a good reason for not accepting it as an incorporated belief, so agent i will rather retract it from his own belief or even add its negation.

For the belief conflict mentioned above, where $\varphi \in L_{i,j}^-(K)$ and $\neg\varphi \in L_i^-(K)$, we have three reasonable and plausible choices:

¹In the following, we disregard $+$ in favor of the more rational $\dot{+}$.

1. $\varphi \in L_{i,j}^-(K), \neg\varphi \in L_i^-(K) \Rightarrow L_i^-(K) \Delta L_{i,j}\varphi = L_i^-(K) \dot{+} \varphi$.
(Agent i accepts the new belief φ on behalf of agent j 's believing φ , although agent i originally believed that φ is false.)
2. $\varphi \in L_{i,j}^-(K), \neg\varphi \in L_i^-(K) \Rightarrow L_i^-(K) \Delta L_{i,j}\varphi = L_i^-(K) \dot{-} \neg\varphi$.
(Agent i drops his disbelief in φ on behalf of agent j 's believing φ .)
3. $\varphi \in L_{i,j}^-(K), \neg\varphi \in L_i^-(K) \Rightarrow L_i^-(K) \Delta L_{i,j}\varphi = L_i^-(K)$.
(Although agent j believes φ , agent i believes that φ is false, and agent i persists in his belief.)

Note that for this special case the alternative of negative-revision, $L_i^-(K) \Delta L_{i,j}\varphi = L_i^-(K) \dot{+} \neg\varphi$, is useless, because $\neg\varphi \in L_i^-(K)$ implies $L_i^-(K) \dot{+} \neg\varphi = L_i^-(K)$. In other words, the strategy coincides with persistence. The same holds for the strategy of positive contraction.

The above analysis shows that there exist three different plausible choices for the belief conflict situation under consideration. In order to disambiguate this conflict situation we refine the notion of sub-belief and we introduce three new credibility operators in the formalism: *High-credibility sub-belief*, *Neutral-credibility sub-belief* and *Low-credibility sub-belief*, denoted $HL_{i,j}\varphi$, $NL_{i,j}\varphi$ and $LL_{i,j}\varphi$ respectively.

The formula $HL_{i,j}\varphi$ means that agent i views agent j as an agent with high credibility on φ , and $NL_{i,j}\varphi$ and $LL_{i,j}\varphi$ mean respectively that with neutral credibility and that with low credibility.

Based on those credibility operators, the three possible result of the above belief conflict are determined as follows:

- SH-Pr** $\varphi \in L_{i,j}^-(K) \wedge \neg\varphi \in L_i^-(K) \wedge HL_{i,j}\varphi \Rightarrow L_i^-(K) \Delta L_{i,j}\varphi = L_i^-(K) \dot{+} \varphi$.
(Because agent j is viewed as an agent with high credibility about φ , and agent j believes φ , agent i accepts the new belief φ overruling his original belief that φ is false.)
- SN-Nc** $\varphi \in L_{i,j}^-(K), \neg\varphi \in L_i^-(K) \wedge NL_{i,j}\varphi \Rightarrow L_i^-(K) \Delta L_{i,j}\varphi = L_i^-(K) \dot{-} \neg\varphi$.
(Because agent j is viewed as an agent with neutral credibility about φ and agent j believes φ , agent i retracts his original belief that φ is false)
- SL-Pe** $\varphi \in L_{i,j}^-(K), \neg\varphi \in L_i^-(K) \wedge LL_{i,j}\varphi \Rightarrow L_i^-(K) \Delta L_{i,j}\varphi = L_i^-(K)$.
(Although agent j believes φ , agent i believes that φ is false and agent i persists in this belief, since agent j has low credibility about φ .)

4 Role Analysis

In the previous section we have reduced the problem of determining which of the three meaningful belief revision strategies will be invoked in the case of a conflict between the new information and the agent's previous belief to a refinement of the notion of sub-belief into three new notions. This means, however that we now must

cope with a new problem involving these credibility operators; we must invent a strategy which decides which credibility operators will be the result of the first stage of the process of information assimilation under which circumstances. This choice must be based on the logic of belief dependence and not invoke other information outside the formalism.

Below are some plausible axioms for credibility operators. The credibility operator $HL_{i,j}\varphi$ is interpreted as "agent i views agent j as an agent with high credibility about φ ". If we suppose that each agent's perspective on credibility is correct and that these viewpoints are common knowledge among agents, then we can consider those notions concerning highness, neutralness, and lowness as an order relation. In other words, the axioms which we are going to describe are intended to describe the situation where credibility is directly linked to the true observable expertise of the agents. This expertise moreover should be common knowledge among agents. For this scenario, we have the following axioms for our credibility operators:

Axioms:

Irreflexivity $\neg HL_{i,i}\varphi$.

Asymmetry $HL_{i,j}\varphi \rightarrow \neg HL_{j,i}\varphi$.

Transitivity $HL_{i,j}\varphi \wedge HL_{j,k}\varphi \rightarrow HL_{i,k}\varphi$.

Reflexivity $NL_{i,i}\varphi$.

Symmetry $NL_{i,j}\varphi \rightarrow NL_{j,i}\varphi$.

Transitivity $NL_{i,j}\varphi \wedge NL_{j,k}\varphi \rightarrow NL_{i,k}\varphi$.

Definition $LL_{i,j}\varphi \stackrel{def}{=} HL_{j,i}\varphi$.

A little reflection shows that the above axioms can easily be violated in situations where agents are misinformed about each other's expertise, or where there exists disagreements among the agents.

Next we make some analysis on possible configurations among the relied-on relations. In order to obtain an intuitive understanding on this approach, we will use the notion of *role* to refer to different types of agents characterized in terms of being relied on in different configurations; this relates to role theory in social science[1, 9].

For a single agent, there exist the following different roles which the agent can perform (for the formula φ):

Isolated-A The agent relies on nobody including himself and is relied on by nobody:

$$(\neg \exists j)(D_{i,j}\varphi \vee D_{j,i}\varphi)$$

Isolated-B The agent relies on only himself, while nobody relies on him:

$$D_{i,i}\varphi \wedge (\neg \exists j \neq i)(D_{j,i}\varphi \vee D_{i,j}\varphi)$$

Learner The agent only relies on someone else: $(\exists j \neq i)D_{i,j}\varphi \wedge \neg D_{i,i}\varphi$.

Expert The agent relies on both himself and someone else: $D_{i,i}\varphi \wedge (\exists j \neq i)D_{i,j}\varphi$

Authority The agent is relied on both by himself and by someone else, but he relies on nobody else: $D_{i,i}\varphi \wedge (\exists j \neq i)D_{j,i}\varphi \wedge \neg(\exists j \neq i)D_{i,k}\varphi$.

Diffident agent The agent relies on nobody including himself, but is relied on by someone else: $(\exists j \neq i)D_{i,j}\varphi \wedge \neg(\exists k)D_{i,k}\varphi$

Among the above roles, role Isolated-A and role Isolated-B are isolated ones where agents rely on nobody else; in the studies of belief dependence these are trivial roles. The remaining roles are the fundamental roles which are worthy to be named and investigated in depth. For these roles we introduce the notations given below:

$$Learner_i\varphi \stackrel{def}{=} \neg D_{i,i}\varphi \wedge (\exists j \neq i)D_{i,j}\varphi$$

$$Expert_i\varphi \stackrel{def}{=} D_{i,i}\varphi \wedge (\exists j \neq i)D_{i,j}\varphi$$

$$Authority_i \stackrel{def}{=} D_{i,i}\varphi \wedge (\exists j \neq i)D_{j,i}\varphi \wedge \neg(\exists k \neq i)D_{i,k}\varphi.$$

$$Diffident-agent_i\varphi \stackrel{def}{=} (\exists j \neq i)D_{j,i}\varphi \wedge \neg(\exists k)D_{i,k}\varphi.$$

5 Roles and Credibilities

Using the fundamental roles described above, we now propose a strategy, called the *confidence priority strategy*, yielding an intuitive mechanism for choosing between the three credibility operators. This strategy enforces that the more confidently a formula φ is believed by an informer, the more credibility the belief is. We believe that the confidence priority strategy is reasonable and acceptable for cooperative multiple agent environments. If an agent firmly believes a fact φ , to the extent that others rely on him then this indicates that the agent must have strong evidence or a convincing justification for his belief. The other agents therefore should view the agent's belief as a belief with higher credibility.

According to the confidence priority strategy, the fundamental roles can be arranged intuitively in increasing order as diffident-agent, learner, expert, and authority. Note that in our formalism the notion of credibility is a relative one. For instances, an agents who is a learner will view agents that are expert or authority as agents with high credibility about φ . Whereas neutral credibility will be granted to peers, i.e. in the case where both agents are learners, experts, or authorities. The relative credibility relations can be formalized as follows:

Def-High-credibility $HL_{i,j}\varphi \stackrel{def}{=} Diffident-agent_i\varphi \wedge (Learner_j\varphi \vee Expert_j\varphi \vee Authority_j\varphi) \vee Learner_i\varphi \wedge (Expert_j\varphi \vee Authority_j\varphi) \vee Expert_i\varphi \wedge Authority_j\varphi$.

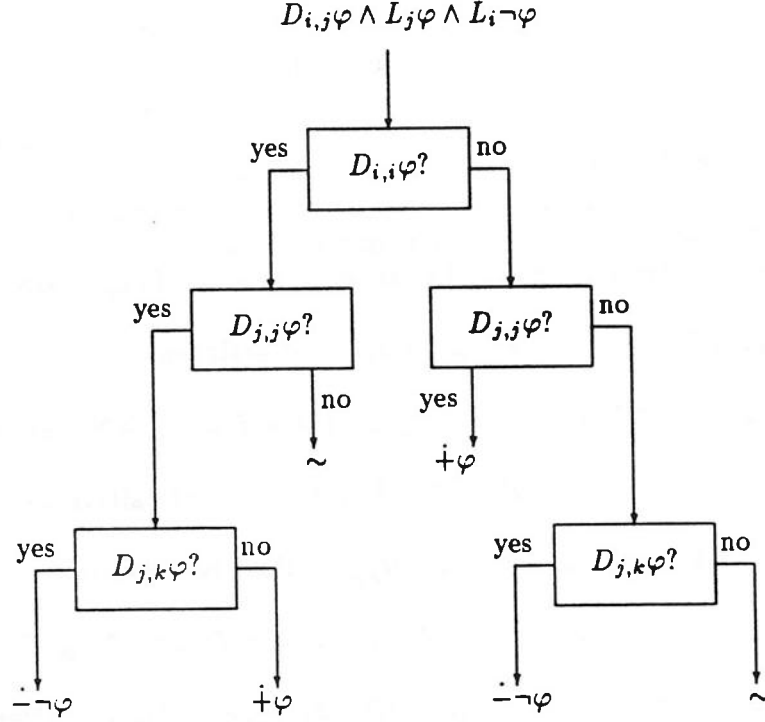


Figure 1: Decision Tree

Def-Neutral-credibility $NL_{i,j}\varphi \stackrel{def}{=} \text{Diffident-agent}_i\varphi \wedge \text{Diffident-agent}_j\varphi \vee \text{Learner}_i\varphi \wedge \text{Learner}_j\varphi \vee \text{Expert}_i \wedge \text{Expert}_j \vee \text{Authority}_i \wedge \text{Authority}_j\varphi$

Def-Low-credibility $LL_{j,i}\varphi \stackrel{def}{=} HL_{i,j}\varphi$

If we combine our proposals in this section and the previous one we obtain a computational strategy for the standard case of the belief conflict under consideration, i.e. the situation where $\varphi \in L_{i,j}^-(K) \wedge \neg\varphi \in L_i(K)$. For the three meaningful revision operators we will use the notations **Positive-revision**, **Persistence**, and **Negative-contraction** to denote the corresponding processes.

If we suppose that axiom (Ldf) $L_{i,j}\varphi \stackrel{def}{=} D_{i,j}\varphi \wedge L_j\varphi$ and the neutral axiom (D $\neg$) $D_{i,j}\varphi \equiv D_{i,j}\neg\varphi$ hold, we have, for the case of normal belief conflict that $\varphi \in L_{i,j}^-(K) \wedge \neg\varphi \in L_i(K)$ implies $D_{i,j}\varphi \wedge L_j\varphi \wedge L_i\neg\varphi$. Therefore, we also call the case in which $D_{i,j}\varphi \wedge L_j\varphi \wedge L_i\neg\varphi$ a normal belief conflict.

For this case of normal belief conflict, we have:

$D_{i,j}\varphi \wedge L_j\varphi \wedge L_i\neg\varphi \Rightarrow D_{i,j}\varphi \Rightarrow (D_{i,j}\varphi \vee D_{i,i}\varphi) \wedge (D_{i,j}\varphi \vee \neg D_{i,i}\varphi) \Rightarrow \text{Expert}_i\varphi \vee \text{Learner}_i\varphi$

In the case that agent i is an expert about φ , and if $\neg D_{j,j}\varphi$ holds, it follows that agent j is a learner or a diffident-agent, and we have $LL_{i,j}\varphi$. Therefore, the outcome

will be persistence, denoted by $\sim$ for short. On the other hand, if $D_{j,j}\varphi$ holds, and if there exists an agent k such that $D_{j,k}\varphi$, the agents j and i are both experts and therefore peers and the outcome will be negative-contraction. Otherwise the process would be positive-revision because agent j is an authority about φ .

For the other case that agent i is a learner about φ , we investigate once more whether $D_{j,j}\varphi$ holds. If this is the case that means that agent j at least is an expert and we have $HL_{i,j}\varphi$ and the process will be positive-revision. On the other hand, if $\neg D_{j,j}\varphi \wedge (\exists k \neq j)D_{j,k}\varphi$ holds, then agent j is a learner, and the outcome will be negative-contraction. In the remaining case the outcome is persistence.

Therefore, the strategy can be expressed as a set of logical sentences as follows:

$$D_{i,j}\varphi \wedge L_j\varphi \wedge L_i\neg\varphi \wedge D_{i,i}\varphi \wedge \neg D_{j,j}\varphi \Rightarrow \text{Persistence.}$$

$$D_{i,j}\varphi \wedge L_j\varphi \wedge L_i\neg\varphi \wedge D_{i,i}\varphi \wedge D_{j,j}\varphi \wedge (\exists k \neq j)D_{j,k}\varphi \Rightarrow \text{Negative-contraction.}$$

$$D_{i,j}\varphi \wedge L_j\varphi \wedge L_i\neg\varphi \wedge D_{j,j}\varphi \wedge \neg(\exists k \neq j)D_{j,k}\varphi \Rightarrow \text{Positive-revision.}$$

$$D_{i,j}\varphi \wedge L_j\varphi \wedge L_i\neg\varphi \wedge \neg D_{i,i}\varphi \wedge D_{j,j}\varphi \Rightarrow \text{Positive-revision.}$$

$$D_{i,j}\varphi \wedge L_j\varphi \wedge L_i\neg\varphi \wedge \neg D_{i,i}\varphi \wedge D_{j,j}\varphi \wedge (\exists k \neq j)D_{j,k}\varphi \Rightarrow \text{Negative-contraction.}$$

$$D_{i,j}\varphi \wedge L_j\varphi \wedge L_i\neg\varphi \wedge \neg D_{i,i}\varphi \wedge \neg D_{j,j}\varphi \wedge \neg(\exists k \neq j)D_{j,k}\varphi \Rightarrow \text{Persistence.}$$

Moreover, the strategy can be expressed as algorithm as it is shown in the figure.

6 Concluding Remarks

We have proposed a rational balance strategy between belief revision and belief persistence in a multiple agent environment. This strategy which is based on the framework of logic for belief dependence, is in principle effectively computable. The confidence priority strategy provides an intuitive, plausible and flexible approach to formalize the relationship between the fundamental roles and credibility operators.

On the other hand, we do not claim that the propose strategy is unique or best possible; we are convinced that there exist many other strategies which are also plausible and acceptable for multiple agent environments; such alternatives could be based on different analysis and perspectives on the rely-on relations in the models. To capture the other plausible strategies is an interesting topic for further research.

References

- [1] Bruce J. Biddle, *Role Theory*, Academic Press, 1979.
- [2] Jon Doyle, A Society of Mind-Multiple Perspectives, Reasoned Assumptions, and Virtual Copies, *Proceedings of the 8th Inter. Joint Conference on AI*, Vol.I,

1983, 309-314.

- [3] Peter Gärdenfors, *Knowledge in Flux—Modeling the Dynamics of Epistemic States*, MIT Press, Cambridge, Mass., 1988.
- [4] Peter Gärdenfors and David Makinson, Revision of knowledge systems using epistemic entrenchment. In M. Vardi, (ed.), *Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning about Knowledge*, Morgan Kaufmann, Los Altos, CA., 1988.
- [5] J. R. Hobbs and R. C. Moore, *Formal Theories of the Commonsense World*. Ablex, Norwood, N.J., 1985.
- [6] Zhisheng Huang, Dependency of Belief in Distributed Systems, in: M. Stokhof and L. Torenvliet (eds.) *Proceedings of The 7th Amsterdam Colloquium*, 637-662, 1990. Also available in: Institute for Language, Logic and Information, University of Amsterdam, Preprint ITLI LP-89-09, 1989.
- [7] Zhisheng Huang and Karen Kwast, Awareness, Negation and Logical Omniscience, in: J. van Eijck (ed.), *Proceedings of European Workshop on Logics in Artificial Intelligence*, (JELIA'90), Springer Lecture Notes in Artificial Intelligence, LNCS vol. 478 (1991), pp. 282-300.
- [8] Zhisheng Huang, Logics for Belief Dependence, in: E. Börger e.a. (eds.), *Proceedings of the 1990 Workshop on Computer Science Logic (CSL'90)*, Springer Lecture Notes in Computer Science, LNCS vol. 533 (1991), pp. 274-288. Also available in: Institute for Language, Logic and Information, Preprint LP-90-13, University of Amsterdam, 1990.
- [9] J. A. Jackson, *Role*. Cambridge University Press, 1972.
- [10] Martin, J. P. and Shapiro, C. S., Theoretical foundations for belief revision, in: J. Y. Halpern (ed.) *Theoretical Aspects of Reasoning about Knowledge: Proceedings of the 1986 conference* (Morgan-Kaufmann, Los Altos, CA, 1986) 383-398.
- [11] Nebel, B., *Reasoning and Revision in Hybrid Representation Systems*, Lecture Notes in Computer Science, Vol. 422, Springer-Verlag, 1990.
- [12] Potts, G., John, M., and Kirson, D., Incorporating New Information into Existing World Knowledge, *Cognitive Psychology* 21, (1989), 303-333.

Bach-Peters Sentences in a Variable-Free Semantics*

Pauline Jacobson
Dept. of Cognitive and Linguistic Sciences
Brown University

1. Introduction

This paper takes as its point of departure the hypothesis that surface structures directly receive a model-theoretic interpretation; hence the semantics needs no intermediate levels of representation such as Logical Form. Within the general program of direct surface interpretation, I'll argue for a semantics which makes no essential use of variables - such an idea was proposed in Quine (1966), and has been explored more recently within a good deal of work within Categorical Grammar such as Szabolcsi (1987; 1991), Hepple (1990), and Jacobson (1991; to appear). After first developing one implementation of a semantics without variables, I will then apply this to some facts surrounding Bach-Peters' sentences which were first discussed in Jacobson (1977).

2. Significance of Bach-Peters' sentences

The interest in Bach-Peters' sentences lies in their interaction with weak crossover, and in Jacobson (1977) I argued that this interaction provides crucial evidence for a level of Logical Form. To elucidate this, consider first some run-of-the-mill Bach-Peters' sentences such as those in (1) and (2):

- (1) The man_i who loves her_j kissed the woman_j who wrote to him_i.
- (2) The man_i who loves her_j kissed his_i wife_j.

The essential claim in Jacobson (1977) is that the second pronoun in such sentences is an ordinary bound variable, while the first is a paycheck pronoun. (This is quite similar to the analysis of Karttunen, 1971; the major difference is that he allowed for a second possibility in which the first pronoun is the bound variable and the second is the paycheck pronoun. Notice, however, that this in itself violates weak crossover, since object NPs in general cannot bind variables within the subject; I return to this point below.) It is worth pointing out that paycheck pronouns can, at least marginally precede their antecedents as in (3), and so the claim that the first pronoun here is paycheck pronoun is the "null hypothesis" regarding Bach-Peters' sentences:

- (3) Mary put it in the bank, but Sue put her paycheck in the Credit Union.

To flesh out the analysis in somewhat greater detail, consider the proposal regarding paycheck pronouns made by Cooper (1979), as modified by Engdahl (1986). Thus take a paycheck sentence like (4):

- (4) The woman who put her paycheck in the federally insured bank was wiser than the one who put it in the Brown Employees' Credit Union.

Under the Cooper/Engdahl proposal, the pronoun *it* here translates a free variable over functions from individuals to individuals applied to a variable over individuals and so the meaning of the second NP is as represented in (5):

- (5) ...than the x : x put $f(x)$ in the Brown Employees' Credit Union
(for f a variable of type $\langle e, e \rangle$)

Here the individual variable x is bound by the subject of the relative clause, while the variable f remains free and picks up a contextually salient function from individuals to individuals - in this case, the paycheck-function.

Let us then take the analysis of Bach-Peters' sentences of Jacobson (1977), modified in accordance with the Cooper/Engdahl analysis of paycheck pronouns. Using for the moment the vocabulary of Logical Form, the claim was that (1) and (2) have roughly the Logical Forms represented in (6) and (7):

- (6) [S [QP the x : $\text{man}'(x)$ & $\text{love}'(f(x))(x)$] [S [QP the y : $\text{woman}'(y)$ and $\text{wrote-to}'(x)(y)$] [S kissed(y)(x)]]]
- (7) [S [QP the x : $\text{man}'(x)$ and $\text{love}'(f(x))(x)$] [S [QP the y : $\text{wife-of}'(x)(y)$] [S kissed'(y)(x)]]]

Thus in the case of (6) - which is the LF for (1) - the variable f remains free and picks up some contextually salient function from individuals to individuals - here, it picks up the function mapping each person into the woman who loves them. In the case of (7) - which is the LF for (2) - the free variable f picks up the wife-function. (It should be noted that this is not exactly the analysis in either Karttunen (1971) or in Jacobson (1977), since we both took paycheck pronouns to be represented as full-blown NPs like "x's wife" or "the woman who wrote to x". There was, however, no motivation for supposing that paycheck pronouns had full representations of this sort at the level of LF; I argued only that at LF they must contain an occurrence of the variable x which is bound by the subject of the relative clause, and this is completely consistent with the Cooper/Engdahl analysis. I will, therefore, take the Cooper/Engdahl analysis as the point of departure for the discussion here.)

Thus assuming that it is basically correct to treat the first pronoun as a paycheck pronoun, the question arises as to whether the representation $f(x)$ for the paycheck pronoun is just a convenient representation of its meaning, or whether it is crucial that a paycheck pronoun actually have a complex representation of this sort at some significant level of representation such as LF. Now I assumed that Weak Crossover is a constraint on representations - and given this, I argued that the paycheck pronoun crucially must have a complex representation at some level. The reason is that it acts with respect to Weak Crossover as if it actually contained an instance of the variable x within it. The crucial sentences, then, are (8) and (9), which are impossible as Bach-Peters' sentences:

- (8) *The man_i who she_j loves kissed the woman_j who wrote to him_i.
(9) *The man_i who she_j loves kissed his_j wife_j.

In a nutshell, the idea is that if *she* in each of these has a fuller representation like $f(x)$, then these will exhibit weak crossover violations. The exact statement of why this is true depends to some extent on one's exact formulation of Weak Crossover. But for the sake of illustration, assume for convenience that Weak Crossover says that the trace of a

wh or a quantified NP cannot be co-indexed at LF with a variable that it does not c-command. What this means is that if we unpack the paycheck pronoun *she* in (8) and (9) at LF into the complex representation $f(x)$ we'll correctly predict that these violate Weak Crossover; I've illustrated this with the representations in (10) and (11):

(10) the man, x [who_x [$f(x)$ loves t_x]] kissed the woman who wrote to x (= (8))

(11) the man, x [who_x [$f(x)$ loves t_x]] kissed x 's wife (= (9))

Notice that in the good Bach-Peters' sentences in (1) and (2) there is no Weak Crossover violation, as here the trace does c-command the variable x within the representation of the paycheck pronoun. This is shown in (12) (the representation for (1)):

(12) the man, x [who_x [t_x loves $f(x)$]] kissed the woman who wrote to x

It should be noted that if one were to seriously work out this account under the assumption that Weak Crossover is a constraint on representations, then these facts raise some interesting questions about just how to formulate the constraint. However, since I'm ultimately going to reformulate Weak Crossover as a constraint on how meanings can be put together, I won't pursue this here. Suffice it to say that it's clear that if Weak Crossover is a constraint on representations, then it appears that the paycheck pronoun - i.e., the first pronoun in a Bach-Peters' sentence - has to be unpacked into a complex representation, and this representation has to contain an instance of the variable which is co-indexed with the relative pronoun or with its trace. (Rohrbacher (1991) provides another reanalysis of my original analysis. It seems to me that his is in much the same spirit as the analysis developed immediately above, although it is somewhat difficult to compare these as Rohrbacher does not spell out the full semantics associated with his analysis.)

Before leaving this, let me note one other point in passing. As mentioned above, Karttunen actually proposed that either pronoun could be the paycheck pronoun while the other is the bound variable. However, in Jacobson (1977) I noted that the first pronoun could not be the ordinary bound variable, because Weak Crossover again prevents this. In other words, consider again a sentence like (1). Suppose that the first pronoun - the pronoun *her* - were the ordinary bound variable, and the second pronoun - the pronoun *him* - were the paycheck pronoun. Then that too would constitute a Weak Crossover violation, since the object NP would be binding a pronoun which it does not c-command. It follows, then, that the second pronoun must be the ordinary bound variable, and this in turn means that it shouldn't behave as if it had a more complex "unpacked" representation with respect to Weak Crossover. And, in fact, this prediction is borne out; notice that in contrast to a sentence like (8), there is no Weak Crossover violation in (13):

(13) The man_{*j*} who loves her_{*j*} kissed the woman_{*j*} who he_{*j*} wrote to.

Here the second pronoun *he* is an ordinary bound variable and does not have the complex representation that we'd get if it were a paycheck pronoun and were unpacked to $f(x)$. Hence, there is predictably no Weak Crossover violation within the second relative clause.

What I want to do in the remainder of this paper, however, is to develop an account of variable binding which makes no essential use of either Logical Form or of variables. I'll show that under this account, Weak Crossover is a constraint on how meanings can be combined. Once we have this apparatus, then all of the above facts fall into place

without making any crucial use of either representations or of variables, and so these facts turn out to be quite compatible with direct interpretation of surface structures. Thus we will at this point leave Bach-Peters' sentences, and turn to the No Essential Variables approach of pronoun meanings.

3. *The No Essential Variables Approach to Pronouns*

We begin by considering some ordinary cases of variable binding, and in particular we turn to the three sentences in (14):

- (14) a. Every man_i told every woman_j that I thought that he_j saw her_j.
 b. Every man_i told every woman that I thought that he_j saw Mary.
 c. Every man told every woman that I thought that John saw Mary.

The standard approach to variable-binding is well-known, and so I won't say much about it here except to mention a few points. First the standard approach has two related advantages. One is that the most deeply embedded sentences in each of (14a) through (14c) have the same kind of meaning - all of these denote propositions relative to some way to assign values to variables. The fact that constituents with pronouns denote the same kind of things as similar looking constituents without pronouns appears to be a happy result since - modulo sentences containing resumptive pronouns - constituents with pronouns seem to have essentially the same syntactic distribution as corresponding constituents without pronouns. A second and related fact is that this means that the combinatorics is uniform. In other words, in all three cases in (14) the meaning of the most deeply embedded S combines with the meaning of *thought* in the same way.

But despite these advantages, there are also problems with the standard approach. First, linguistic expressions have meaning only relative to some way to assign values to variables. Put differently, semantics is more than just a mapping between surface structures and model-theoretic objects: surface structures are mapped into a model-theoretic object only relative to some assignment function which assigns values to variables. Of course one could instead claim that the meaning of, for instance, a sentence is a function from assignment functions to propositions. But this means that the assignment functions themselves are model-theoretic objects, which in turn means that the variables are model-theoretic objects. This point is discussed in some detail in Landman and Moerdijk (1983), and while it's hard to definitively show that variables are not model-theoretic objects, it would clearly be preferable to dispense with these as model-theoretic objects. Second, most accounts of variable binding crucially rely on some level of representation with indexed pronouns. If we seriously wish to pursue a theory with model-theoretic interpretation of surface structures we have to say that the pronouns "come" with an index - again, this is discussed in Landman and Moerdijk. What that means, then, is that there is not a single lexical item *he* in English, but rather an infinite number of them - *he*₁, *he*₂, , etc. (And note that under the Cooper/Engdahl analysis of paycheck pronouns discussed above the situation is even more complex; since a pronoun can translate as a variable over functions applied to a variable over individuals, there will need to also be an infinite number of pronouns with pairs of indices. In fact, as will be discussed below, this analysis requires pronouns to translate as variables over n-place functions applied to n-individuals. This means that, for any n, there are pronouns with n indices. Since n is guaranteed to be finite and since there are a denumerable number of possible indices for each position this doesn't add to the number of lexical items - there are still denumerably infinitely many of them - but it does complicate the picture.) The fact

that there are an infinite number of lexical items in itself is not necessarily objectionable, but what is peculiar is that all of the pronoun with, for example, one index ultimately make exactly the same contribution to the meaning; a fact which is not expected if these are different lexical items.

Thus we can now turn to the alternative account of variable binding - one which was first proposed in Quine (1966) and is also explored within Combinatory Logic. More recently, its linguistic implications have been explored in Szabolcsi (1987;1991), Hepple (1990), and Jacobson (1991; to appear). My implementation of this idea will be a bit different from that of Szabolcsi or Hepple (although it is closer to Hepple's). However, space precludes a detailed comparison of these various implementations; suffice it to say that mine is within the same general spirit as these others.

The basic idea is that a sentence containing a pronoun, such as the most deeply embedded S in (14b) actually denotes a function from individuals to propositions, which we will refer to as a property. A constituent with two pronouns - such as the most deeply embedded S in (14a) - simply denotes a function from two individuals to propositions - that is, a 2-place relation.

In other words, take a sentence like (15):

(15) Every man_i thinks that Mary said that he_j lost.

I have claimed that the lowest sentence *he lost* simply denotes the losing-property but this raises two questions: (1) How does it get this meaning? (2) How does the entire sentence eventually get the bound variable interpretation? To deal first with (1), we will assume that there is a single lexical item *he* in English, and that it just denotes the identity function on individuals. Moreover, assume that when its meaning combines with the meaning of *lost* it does so by function composition. Of course strictly speaking *he* denotes the identity function on male individuals, and so the sentence *he lost* means a bit more than just the intransitive verb *lost* does, since it denotes a partial function defined only for male individuals. For convenience, however, I'll systematically ignore the gender, number, and person information contained in pronouns.

Moving up to the next clause, we will assume that the meaning of *he lost* also combines with the meaning of *think* by function composition, and this in turn function-composes with the meaning of type-lifted *Mary*. The result, then, is that the constituent *that Mary said that he lost* denotes the property $\lambda x[\text{say}'(\text{lost}'(x))(m)]$. (There are other possibilities here; in the implementation sketched above *said-that-he-lost* is a function of type $\langle e, \langle e, t \rangle \rangle$ and function-composes with $\lambda P[P(m)]$. However, rather than type-lift *Mary* this could simply denote the individual *m* where this fills the "inner" argument position of *said-that-he-lost*. Nothing in the present account crucially hinges on which of these options turns out to be correct.)

At this point, three questions arise. First, how does the pronoun *he* eventually get bound in a sentence like (15)? Second, what happens for the case of free pronouns which do not get bound? And third, how do the combinatorics work in cases where there is more than one pronoun? I'll concentrate primarily on the first question, but will first briefly address the second and third. As to the case of free pronouns, consider a sentence with a free pronoun such as (16):

(16) Mary said that he lost.

I see no reason to assume that free pronouns are any different from bound ones. In other words, we will assume that here too *he* denotes the identity function on individuals, and

we will further assume that the meaning of *said that he lost* is put together exactly as in (15). What this means is that (16) ends up not denoting a proposition but rather a function from individuals to propositions. Of course to extract information out of this what the hearer generally does is to apply this function to some contextually salient individual, but I'll assume that the meaning of (16) is just a property.

Turning to a case with two pronouns, we can note that additional combinatorics will be needed for, e.g. (17):

(17) Every woman_i said that no man_j believes that the book she_j writes will please him_i.

Space precludes a detailed discussion of just how these combinatorics work and of how the syntax works with the semantics so that the meanings are combined in the appropriate ways. Let us simply assume here that the underlined constituent in (17) can have either of the meanings shown in (18); these are identical except that they differ in terms of the order in which the pronouns are expected to be "bound":

- (18) a. $\lambda x[\lambda y[\text{please}'(x)(\lambda z[\text{book}'(z) \ \& \ \text{write}'(z)(y)])]]$
 b. $\lambda y[\lambda x[\text{please}'(x)(\lambda z[\text{book}'(z) \ \& \ \text{write}'(z)(y)])]]$

The reader might wonder whether there is a natural set of operations which will yield both of these meanings. I hope to address this in detail in future work, but will just note here that one rather natural set of operations to yield (18a) will indeed also yield (18b). The reason that both meanings are necessary is that alongside of (17) - in which the pronouns are bound in crossed order - we also have (19) - in which they are bound in nested order:

(19) Every man_j said that no woman_i believes that the book she_j writes will please him_i.

I will return to this in Sec. 5, after the variable binding mechanism has been introduced.

The question of immediate relevance, then, is just how the combinatorics work so as to ensure that variable-binding can take place. Consider again a sentence like (15). Obviously, function composition can't keep happening "all the way up", or else the final result will be that the pronoun ends up a free pronoun. In fact, there are a variety of related ways to put the meanings together so as to effect variable binding. What I will now do is to show that one of these ways has considerable independent motivation, and thus nothing new is needed in order to get the effect of variable binding.

4. Functional Questions and Paycheck pronouns

To motivate the apparatus that I'll use for variable-binding, we turn now to the analysis of functional questions proposed in Groenendijk and Stokhof (hereafter, G&S) (1983) and Engdahl (1986). Thus consider a functional question like (20) on the reading where *his mother* is an appropriate answer:

(20) Who does every Englishman love?

To account for the relevant reading of (20), G&S and Engdahl analyze these in such a way that the gap corresponds to a variable over functions from individuals to individuals applied to a variable over individuals. In other words, the meaning of (20) can be represented - semi-informally - as shown in (21):

(21) what is the function f such that: $\text{every-Englishman}'(\lambda x[\text{love}'(f(x))(x)])$

Of course a full analysis of (20) requires recasting (21) in accordance with a semantics for questions; since the correct analysis of the semantics of questions is irrelevant for the discussion here I will not do this here. It is sufficient for our purposes to assume that the meaning of the constituent *every Englishman love* in (20) will be as shown in (22):

(22) $\lambda f[\text{every-Englishman}'(\lambda x[\text{love}'(f(x))(x)])]$

This will then occur as argument of *who* and the rest of the analysis can be formulated in accordance with either the semantics of questions in Karttunen (1977) or with the analysis in Groenendijk and Stokhof (1982); other analyses are undoubtedly possible as well.

Before continuing, however, it is interesting to note that under the standard approach to variables it is somewhat surprising that *his mother* is an appropriate answer to a functional question like (20). After all, under the standard analysis, *his mother* denotes - relative to an assignment function - an individual, whereas the question asks for the identity of some function of type $\langle e, e \rangle$. But notice that under the No Essential Variables approach, it is quite unsurprising that *his mother* is an appropriate answer. *his* denotes only the identity function on individuals; it function-composes with *mother'*, and so this entire NP denotes the mother-function which is, of course, of the right type to answer the question. For relevant discussion of the functional readings of such answers, see also Ginzburg (1992).

Returning now to the functional question itself, Engdahl (1986) notes two further points of relevance. First, she points out that this allows for a unified treatment of paycheck pronouns and what we will call functional gaps. Recall that under the Cooper analysis of paycheck pronouns as modified by Engdahl, a paycheck pronoun like that in (4) also translates as a variable over functions of type $\langle e, e \rangle$ applied to a variable over individuals. Second, Engdahl notes that the analysis must actually be generalized so that the gap can translate into a variable over n -place functions applied to n variables over individuals. Thus, for example, take the question-answer pairs in (23):

(23) What did each boy think that no teacher would like? The valentine he made for her.

The meaning of the question can be accounted for by allowing the gap to be a variable of type $\langle e, \langle e, e \rangle \rangle$ applied to two variables over individuals. Using W for a variable of the appropriate type, we can thus represent the meaning of (23) as in (24):

(24) what is the function W such that:
 $\text{every-boy}'(\lambda x[\text{hope}'(\text{no-teacher}'(\lambda y[\text{like}'(W(x)(y))(y))](x)])$

Incidentally, unless we take propositions to be "nominalized" in the sense of Chierchia (1984), we will presumably also need functional gaps which correspond to variables over types whose range are propositions as well as individuals; this will be needed in cases like (25) and (26):

(25) What does every man believe? That he is a fool.
 what is the P (for P a variable of type $\langle e, t \rangle$) such that:
 $\text{every-man}'(\lambda x[\text{believe}'(P(x))(x)])$

- (26) What does every man think no woman would believe? That he loves her.
 what is the R (for R a variable of type $\langle e, \langle e, t \rangle \rangle$) such that:
 $\text{every-man}'(\lambda x[\text{think}'(\text{no-woman}'(\lambda y[\text{believe}'(R(y)(x))(y))](x)])$

The same facts hold true for paycheck pronouns (see Cooper, 1979); thus the analysis must be generalized to allow a paycheck pronoun to correspond to a variable over n-place functions applied to n-individuals; this can be shown by (27), in which the pronoun *it* can be understood as "the poem he writes for her":

- (27) Every boy thinks that each teacher will love the poem that he writes for her, but every boy also thinks that each girl in the class will hate it.

The relevant meaning of (27) can be composed if the paycheck pronoun translates as $W(x)(y)$ where the variable y is bound by *each girl in the class* and the variable x is bound by *every boy* and where W picks up the contextually salient function of type $\langle e, \langle e, e \rangle \rangle$ which maps pairs individuals x and y into individuals z such that z is the poem that x writes for y . Note further that we also have paycheck pronouns which contain variables of type $\langle e, t \rangle$ and of type $\langle e, \langle e, t \rangle \rangle$; the latter is illustrated by (28):

- (28) Every greedy man wants each rich woman to believe that he's in love with her, but no honest man wants each rich woman to believe it.

I'll assume that this kind of analysis is basically right, but in Jacobson (1991; to appear) I proposed a slight revision of this. Under the Cooper/Engdahl/G&S analysis, the pronoun or gap translates as a variable over n-place functions, applied to n variables over individuals; we will refer to this as the complex meaning analysis. The idea in my revision is that the paycheck pronoun or functional gap is just a variable over functions of the appropriate type, and doesn't contain any occurrence of the individual variables. Of course I ultimately want to do away with variables altogether, but once we get rid of these individual variables, then it's a quick step to also get rid of any essential use of the functional variable. For now, we will continue to use the vocabulary of a theory with variables.

Thus consider again the functional question in (20). The idea of the proposed revision is that the gap is simply a "missing" function of type $\langle e, e \rangle$, and the individual to which which the functional gap applies is built in via a type-shift operation operating on the meaning of *love*. In other words, in its basic meaning *love* denotes an ordinary relation between individuals. However, we'll let it type-shift into a second meaning, where it denotes a relation between individuals and functions from individuals to individuals. We'll call this shifted meaning $z(\text{love})$ - to $z(\text{love})$ some function f is to be an x who ordinary-loves $f(x)$.

In general, then, we define the operation z as in (29):

- (29) Let F be a function of type $\langle X, \langle e, Y \rangle \rangle$ (for X and Y variables over types).
 Then $z(F)$ is a function of type $\langle \langle e, X \rangle, \langle e, Y \rangle \rangle$, where
 $z(F) = \lambda g[\lambda x[F(g(x))(x)]]$ (for g a variable of type $\langle e, X \rangle$).

Assume further that there is a general type-shifting operation as shown in (30):

- (30) For any expression a_1 of syntactic category A and with meaning of type $\langle X, \langle e, Y \rangle \rangle$, there is a (phonologically identical) expression a_2 of category A with meaning of type $\langle \langle e, X \rangle, \langle e, Y \rangle \rangle$, where $a_2' = z(a_1')$.

In a functional question like (22a), then, the gap doesn't have a complex meaning - it's simply a "missing" argument of type $\langle e, e \rangle$ where *loves* has undergone the z -operation. Informally, then, the way that the meaning is put together is as shown in (31):

- (31) what is the function f such that: $\text{every-Englishman}'(z(\text{love}')(f))$

To work this through a bit more formally, (32) shows just how the meaning of the constituent *every Englishman loves* is put together here - all that happens is that *every-Englishman* function-composes with $z(\text{love})$. This constituent then ends up with the same meaning as in the Engdahl/G&S analysis, but it gets this meaning without assigning a complex meaning to the gap:

- (32) $\text{every-Englishman}' \circ z(\text{love}') = \text{every-Englishman}' \circ \lambda f[\lambda x[\text{love}'(f(x))(x)]]$
 $= \lambda f[\text{every-Englishman}'(\lambda x[\text{love}'(f(x))(x)])]$

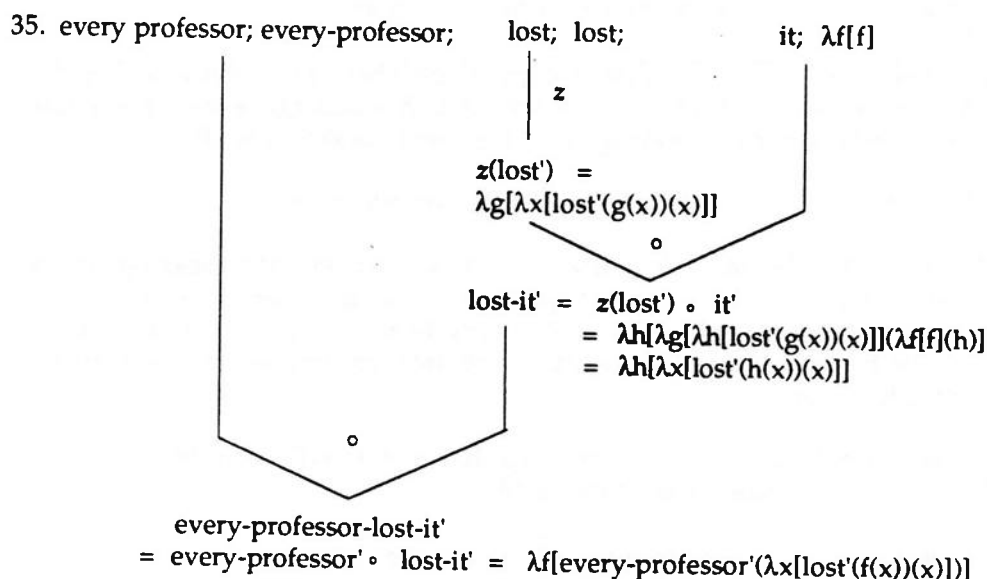
Paycheck pronouns work the same way. Take a case like (33):

- (33) Every graduate student deposited her paycheck, but every professor lost it.

Here the paycheck pronoun simply denotes a free variable of type $\langle e, e \rangle$, and *lost* shifts by z . Since ordinary personal pronouns in general can range over a variety of things (properties, propositions, etc.) it's not particularly surprising to find that it can also range over functions of this type. So, the meaning of (33) is put together as shown in (34a); (34b) shows that this is equivalent to the Cooper/Engdahl meaning:

- (34) a. $\text{every-professor}'(z(\text{lost}')(f)) =$
 b. $\text{every-professor}'(\lambda g[\lambda x[\text{lost}'(g(x))(x)]](f)) = \text{every-professor}'(\lambda x[\text{lost}'(f(x))(x)])$

Of course here I'm using part of the terminology of a theory with variables - I'm saying that the pronoun is a free variable over functions of type $\langle e, e \rangle$. But this is only for expository convenience - as discussed earlier we don't need any variables at all, and so strictly speaking the meaning of the paycheck pronoun is the identity function on functions of type $\langle e, e \rangle$. Hence the whole thing is actually put together as shown in (35):



Two points about this analysis should be made before continuing. First, the z operation actually needs to be generalized; as stated in (29) it allows a paycheck pronoun or a functional gap to only be applied to the individual which occupies the "next argument" slot of the shifting function. In other words, if we assume that the argument structure for a verb like *tell* is $\langle t, \langle e, \langle e, t \rangle \rangle \rangle$ then we have not allowed the subject argument to "bind" in a functional question like (36):

(36) What did every man tell Sue? That he was a fool.

This can be remedied quite simply, by generalizing z as shown in (37):

- (37) Let F be a function of type $\langle X, \langle A_1, \dots, A_n, \langle e, Y \rangle \rangle \rangle$ (for X and Y variables over types, and $A_1, \dots, A_n$ variables over a sequence of types)
 Then $z(F)$ is a function of type $\langle \langle e, X \rangle, \langle A_1, \dots, A_n, \langle e, Y \rangle \rangle \rangle$, where
 $z(F) = \lambda g[\lambda a_1, \dots, \lambda a_n[\lambda x[F(g(x))(a_1), \dots, (a_n)(x)]]]$
 (for g a variable of type $\langle e, X \rangle$, and $a_1, \dots, a_n$ variables of type $A_1, \dots, A_n$)

Second, and quite crucially, I am assuming that z is not a lexical operation; it can apply to expressions derived in the syntax as well as to lexical expressions. This will thus allow for an account of functional questions like (38):

(38) Who does every Englishman think that the Queen will invite to dinner? His mother.

Here *think that the Queen will invite to dinner* can first combine as it would in a normal question, to give the meaning shown in (39):

(39) $\lambda x[\text{think}'(\text{invite-to-dinner}'(x)(\text{Queen}))]$

This can then shift by z to give the meaning in (40):

$$(40) \lambda f[\lambda y[\lambda x[\text{think}'(\text{invite-to-dinner}'(x)(\text{Queen}'))](f(y))(y)]] = \lambda f[\lambda y[\text{think}'(\text{invite-to-dinner}'(f(y))(\text{Queen}'))(y)]]$$

Moreover, it was noted above that Engdahl's analysis of functional gaps (and paycheck pronouns) allows a gap (or pronoun) to translate as a variable over n-place functions applied to n individual variables; this generalization was needed for cases like (23), (27), and (28). In the case of functional gaps which are applied to n individuals no new apparatus is needed under the present approach. For example, consider again (23). The meaning of the phrase *each boy think that no teacher would like* should be (41):

$$(41) \lambda W[\text{each-boy}'(\lambda y[\text{think}'(\text{no-teacher}'(\lambda x[\text{like}'(W(y)(x))(x))](y)))]$$

This meaning is arrived at by combining the items as shown in (42); this is fleshed out in detail in (43) (an underlined constituent indicates that it has undergone *z*):

$$(42) \text{each-boy}' \circ z(\text{think}' \circ (\text{no-teacher}' \circ z(\text{like}')))$$

$$(43) z(\text{like}') = \lambda f[\lambda x[\text{like}'(f(x))(x)]]$$

$$\text{no-teacher-}\underline{\text{like}'} = \text{no-teacher}' \circ \underline{\text{like}'} = \lambda f[\text{no-teacher}'(\lambda x[\text{like}'(f(x))(x))]]$$

$$\text{think-no-teacher-}\underline{\text{like}'} = \text{think}' \circ \text{no-teacher-}\underline{\text{like}'} \\ = \lambda f[\text{think}'(\text{no-teacher}'(\lambda x[\text{like}'(f(x))(x)))]$$

$$z(\text{think-no-teacher-}\underline{\text{like}'}) = \lambda W[\lambda y[\text{think-no-teacher-}\underline{\text{like}'}(W(y))(y)]] \\ = \lambda W[\lambda y[\text{think}'(\text{no-teacher}'(\lambda x[\text{like}'(W(y)(x))(x))](y)]]$$

$$\text{each-boy-}\underline{\text{thinks-no-teacher-likes}'} = \text{each-boy}' \circ \underline{\text{think-no-teacher-}\underline{\text{like}'}} \\ = \lambda W[\text{each-boy}'(\lambda y[\text{think}'(\text{no-teacher}'(\lambda x[\text{like}'(W(y)(x))(x))](y)))]$$

To allow for a case like (28) which involved a paycheck pronoun, we need only assume that the pronoun can denote not only the identity function over individuals and the identity function over functions of type $\langle e, e \rangle$, but it can also denote the identity function over functions of type $\langle e, \langle e, e \rangle \rangle$, the identity function over functions of type $\langle e, \langle e, t \rangle \rangle$, and so forth. Since there is no limit to the complexity of these functions then it follows that here too, as in the standard approach to pronouns, there are an infinite number of pronoun meanings. But this nonetheless avoids the objection raised earlier as regards the standard approach - for here each such pronoun makes a distinct contribution to the meaning.

In Jacobson (1991; to appear) I give several arguments for this revision of the Cooper/Engdahl/G&S analysis, which I'll briefly discuss here. The first is that the Engdahl/G&S analysis of functional gaps is incompatible with most accounts of extraction within Categorical Grammar. While there are a variety of such accounts (see, e.g., Steedman, 1987, Oehrle, 1990 for two), most have in common the idea that a gap does not correspond to a trace or anything else - it is nothing more than the result of some expected argument being postponed. It is thus not clear under these types of accounts how it could have the kind of complex meaning that it must have in the Engdahl/G&S analysis. Notice that under my revision, the gap in a functional question is like any other gap - it is just a "missing" argument. The reason it corresponds to a function is simply because the constituent missing it has type-shifted to expect a function in that position.

Secondly, this revision generalizes a proposal made in Chierchia (1990) to account for certain unexpected inferences of the type shown in (44):

- (44) a. Every Englishman believes whatever every Frenchman believes.
 b. Every Frenchman_i believes that he_i should drink lots of wine.
 c. Therefore, every Englishman_j believes that he_j should drink lots of wine.

As pointed out by Chierchia (among others), the conclusion in (44c) should not be inferable from (44a) and (44b) under the standard account in which the embedded sentence in (44b) denotes a proposition relative to some assignment function assigning values to the variables. Clearly, however, this is a valid inference. Chierchia's solution to this rests on three things. First, he assumes that there are two *believes* listed in the lexicon. The ordinary *believe* is a relation between individuals and propositions. The second, which I'll call *believe*₂ - denotes a relation between individuals and properties such that *x* stands in the *believe*₂ relation to a property *P* just in case *x* ordinary believes the proposition *P*(*x*). Second, Chierchia assumes that in some cases an embedded *S* can denote a property - and this is what happens in (44b). Basically, what Chierchia assumes is that the embedded *S* here starts out denoting the open proposition *x* should drink wine, and the *x* gets bound by a lambda such that this open proposition is mapped into a property. Thus the key claim in Chierchia's analysis is that in all three sentences we have an instance of his *believe*₂.

Now notice that all these mechanics are subsumed under the more general proposals here. Chierchia's *believe*₂ is nothing more than *z(believe)*. The difference is that Chierchia takes this to be a fact about a few lexical items, while the claim here is that this kind of type-shift process is perfectly general. This, then, allows for unexpected inference cases like (44) to be treated in just the same way that we treat functional questions and paycheck pronouns. Moreover, notice that under the no essential variables approach it follows immediately that the embedded *S* in (44b) denotes a property, and so we don't need any special rule mapping an open proposition into a property.

A related advantage to this approach concerns a sentence like (45) (Groenendijk and Stokhof, 1983):

- (45) Every Englishman admires someone the most. It's his mother.

This seems to clearly be the same basic phenomenon as is found in functional questions. However, to directly extend the complex meaning analysis of pronouns and gaps to a case like (45), we would have to say that *someone* here translates as an existentially quantified function of type $\langle e, e \rangle$ applied to a variable over individuals. If we assume that quantified NPs raise at LF and leave a trace then this is perhaps not too surprising: the raised NP *someone* can be translated into a generalized quantifier over functions while the trace translates as a variable over functions applied to a variable over individuals. If, on the other hand, one wishes to interpret quantified NPs *in situ* (as is consistent with the program of direct interpretation), then it's not clear how to get the right result here. Even if one can devise an appropriate meaning for *someone*, this treatment seems much less natural for the case of NPs. After all, Cooper's original idea is that pronouns translate as variables, so it makes some sense to have them translate as sequences of variables. Engdahl's further idea is that gaps are like pronouns, so they too can have these complex meanings. However, there's much less motivation for saying that an NP like *someone* translates as a generalized quantifier over functions which are applied to some variable over individuals.

Under the present analysis, (45) is perfectly straightforward, modulo the completely independent question of how to treat wide scope quantifiers in object position. With regard

to this, suppose that one analysis of a sentence like (46) under the reading in which *some book* has wide scope is as follows:

(46) Every man read some book.

Following Partee and Rooth (1983), assume that *read* is listed in its lowest type $\langle e, \langle et \rangle \rangle$. Assume further that this can compose with the subject first, and then the meaning of *every man read* occurs as argument of the object. With this (and probably with any other mechanism for wide scope objects) the analysis of (45) is straightforward, provided that we allow *someone* to be polymorphic - it can denote not only a generalized quantifier built from individuals, but also one built from functions of type $\langle e, e \rangle$. Thus it can have the meaning $\lambda T[\exists f(T(f))]$ (for T a variable over properties of functions (i.e., a variable of type $\langle \langle e, e \rangle, t \rangle$). Then (45) can be derived by letting $z(\text{admire}')$ compose with the subject, and this then occurs as argument of the object, exactly as in the case of (46). This will ultimately yield the "skolemized" meaning: $\exists f(\forall x(\text{Englishman}'(x) \rightarrow \text{loves}'(f(x))(x)))$.

5. Variable Binding in General

As should be clear from the above remarks, we now have all of the pieces to account for variable-binding in general; variable-binding will simply be the result of z applying to some function in such a way that one of its higher arguments "binds" a missing position in one of its lower arguments. To illustrate, consider again (15). As discussed above, the meaning of *(that) Mary said that he lost* is the property $\lambda y[\text{say}'(\text{lost}'(y))(m)]$. This was arrived at by taking *he* to denote the identity function on individuals, composing that with *lost'*, composing the meaning of *he lost* with *say'*, and in turn composing that with the type-lifted meaning of *Mary*. However, we left the analysis of this sentence without showing how it is that the pronoun *he* eventually gets bound.

But having motivated the z operation, we are now in a position to finish the analysis. All that happens here, of course, is that *thinks'* type-shifts by z ; this will then ensure that the subject argument binds. This is illustrated below in (47) (again I will indicate a constituent which has undergone z by underlining it):

(47) $z(\text{think}') = \lambda P[\lambda x[\text{think}'(P(x))(x)]]$
 $\text{Mary-said-that-he-lost}' = \lambda y[\text{say}'(\text{lost}'(y))(m)]$
 $\text{thinks-that-Mary-said-that-he-lost}'$ = $\text{thinks}'(\text{that-Mary-said-that-he-lost}')$
 $= \lambda x[\text{think}'(\lambda y[\text{say}'(\text{lost}'(y))(m)](x))(x)]$
 $= \lambda x[\text{think}'(\text{say}'(\text{lost}'(x))(m))(x)]$

This property is then taken as argument of the meaning of the subject NP *every man*.

While space precludes a discussion of a variety of more complex cases, it is worth turning to a few cases involving more than one pronoun. First, consider a case where there are two pronouns each bound by a different NP:

(48) Every man_i thinks that no woman_j believes that he_i loves her_j.

As discussed earlier, we will assume here that a constituent containing two pronouns can correspond to two different functions - according to the order in which it wants its arguments. In other words, we assume that the combinatorics are such that the lowest embedded S in (48) can have either of the meanings shown in (49):

- (49) a. $\lambda x[\lambda y[\text{love}'(y)(x)]]$ b. $\lambda y[\lambda x[\text{love}'(y)(x)]]$

(Of course each variable in the representation in (49) will actually correspond to a variable containing gender information; x is a variable over male individuals while y is a variable over female individuals, but we are ignoring gender here.) As it turns out, the meaning in (49a) is the appropriate one for (48); (49b) is the appropriate meaning for (50):

- (50) No woman_i thinks that every man_j believes that he_j loves her_i.

(If the "wrong" meaning is chosen in either case, then we will simply get a gender clash; since (49a) actually denotes a function defined only for male individuals it will be inappropriate if it combines in the wrong way with *every woman*). The derivation of (48) is shown in (51); (52) fleshes this out in detail:

- (51) $\text{every-man}'(z(\text{think}')(\text{no-woman}' \circ (z(\text{believe}') \circ (49a))))$

- (52) $\text{believe}' = z(\text{believe}') = \lambda P[\lambda x[\text{believe}'(P(x))(x)]]$

$\text{believe-that-he-loves-her}' = \text{believe}' \circ \lambda z[\lambda y[\text{love}'(y)(z)]]$

$= \lambda P[\lambda x[\text{believe}'(P(x))(x)]] \circ \lambda z[\lambda y[\text{love}'(y)(z)]]$

$= \lambda v[\lambda P[\lambda x[\text{believe}'(P(x))(x)]](\lambda y[\text{love}'(y)(v)])(x)]$

$= \lambda v[\lambda x[\text{believe}'(\lambda y[\text{love}'(y)(v)](x))(x)]]$

$= \lambda v[\lambda x[\text{believe}'(\text{love}'(x)(v))(x)]]$

$\text{no-woman-believes-that-he-loves-her}' = \text{no-woman}' \circ \text{believes-that-he-loves-her}'$

$= \text{no-woman}' \circ \lambda v[\lambda x[\text{believe}'(\text{love}'(x)(v))(x)]]$

$= \lambda y[\text{no-woman}'(\lambda x[\text{believe}'(\text{love}'(x)(y))(x)]]]$

$\text{think}' = z(\text{think}') = \lambda P[\lambda z[\text{think}'(P(z))(z)]]$

$\text{think}'(\text{no-woman-believes-that-he-loves-her}')$

$= \lambda P[\lambda z[\text{think}'(P(z))(z)]](\lambda y[\text{no-woman}'(\lambda x[\text{believe}'(\text{love}'(x)(y))(x)]]])$

$= \lambda z[\text{think}'(\lambda y[\text{no-woman}'(\lambda x[\text{believe}'(\text{love}'(x)(y))(x)]])(z))(z)]$

$= \lambda z[\text{think}'(\text{no-woman}'(\lambda x[\text{believe}'(\text{love}'(x)(z))(x)]])(z)]$

$\text{every-man-thinks-that-no-woman-believes-that-he-loves-her}'$

$= \text{every-man}'(\lambda z[\text{think}'(\text{no-woman}'(\lambda x[\text{believe}'(\text{love}'(x)(z))(x)]])(z))]$

What is somewhat more interesting is the case where two pronouns are bound by the same NP. Now in some cases, one pronoun can bind another. Consider, for example, the case of two free pronouns as in (53):

- (53) He_i loves his_i mother.

Although both are free, we know that the subject can bind the object pronoun; this can be shown by the fact that this allows sloppy identity in VP Deletion contexts like (54):

- (54) He_i loves his_i mother, and so do I.

In fact, however, this is straightforward. In essence, *loves* type-shifts by z and then takes the meaning of *his mother* (which is just the mother-function) as argument; the subject should then bind. In this case, however, we do not have a normal individual or

generalized quantifier as subject, but rather the subject is the identity function over individuals. But, as is generally true when the subject is a pronoun, the meaning of the VP simply composes with the subject meaning. The effect, then, is that the subject will "bind" the other pronoun; this is illustrated in (55):

- (55) $\text{love-his-mother}' = z(\text{love}')(\text{mother}') = \lambda x[\text{love}'(\text{mother-of}'(x))(x)]$
 $\text{he-love-his-mother}' = \text{love-his-mother}' \circ \text{he}' = \lambda y[\text{love}'(\text{mother-of}'(y))(y)]$

This means that the same sort of thing can happen in a case like (56):

- (56) Every man_i said that he_i loves his_i mother.

Here *he* can bind the other pronoun; *say* shifts by *z*, and so *every man* in turn binds *he*.

However, it is not always the case that when there are two pronouns bound by a single NP this can be treated by having one bind the other. Consider, for example, (57):

- (57) Every man_i thinks that the woman who loves him_i hates his_i mother.

Here neither pronoun can be taken as binding the other; this can be evidenced by the fact that in neither case can a quantified NP be substituted into the relevant position so as to bind the other pronoun.

Cases like these may at first glance appear problematic for this account of variable binding. In fact, however, nothing new is needed for these. Recall that it was shown above that *z* must apply in the syntax to any expression of the relevant type, rather than being a lexical operation. But this means that it can also apply to its own output. As it turns out, if *thinks* undergoes two successive applications of *z* the two different variable positions will be "merged", and the subject will bind both of these.

To demonstrate this, we start with the premise that the embedded S *the woman who loves him hates his mother* can have the meaning shown in (58):

- (58) $\lambda x[\lambda y[\text{hates}'(\text{mother-of-}y)(\text{the-woman-who-loves-}x)]]$

(It can also have the meaning in which the variables are expected to be bound in the other order, but either meaning will yield the same final result here.) The meaning of the entire S can be composed as shown in (59); (60) fleshes this out:

- (59) $\text{every-man}'(z(z(\text{think}'))(\text{the-woman-who-loves-him-hates-his-mother}'))$

- (60) $z(z(\text{think}'))$ (of type $\langle\langle e, \langle e, t \rangle \rangle, \langle e, t \rangle \rangle$) = $z(\lambda P[\lambda w[\text{think}'(P(w))(w)]]]$
 $= \lambda R[\lambda v[\lambda P[\lambda w[\text{think}'(P(w))(w)]](R(v))(v)]]$ (for *R* a variable of type $\langle e, \langle e, t \rangle \rangle$)

$\text{thinks-that-the-woman-who-loves-him-hates-his-mother}'$
 $= z(z(\text{think}'))(\lambda x[\lambda y[\text{hates}'(\text{mother-of-}y)(\text{the-woman-who-loves-}x)]]]$
 $= \lambda R[\lambda v[\lambda P[\lambda w[\text{think}'(P(w))(w)]](R(v))(v)]](\lambda x[\lambda y[\text{hates}'(\text{mother-of-}y)(\text{the-woman-who-loves-}x)]]]$
 $= \lambda v[\lambda P[\lambda w[\text{think}'(P(w))(w)]](\lambda x[\lambda y[\text{hates}'(\text{mother-of-}y)(\text{the-woman-who-loves-}x)]](v))(v)]$
 $= \lambda v[\lambda P[\lambda w[\text{think}'(P(w))(w)]](\lambda y[\text{hates}'(\text{mother-of-}y)(\text{the-woman-who-loves-}v)]](v))(v)]$
 $= \lambda v[\lambda w[\text{think}'(\lambda y[\text{hates}'(\text{mother-of-}y)(\text{the-woman-who-loves-}v)]](w))(w)(v)]$
 $= \lambda v[\lambda w[\text{think}'(\text{hates}'(\text{mother-of-}w)(\text{the-woman-who-loves-}v))(w))(v)]$
 $= \lambda v[\text{think}'(\text{hates}'(\text{mother-of-}v)(\text{the-woman-who-loves-}v))(v)]$

This is especially interesting since in this case the account here makes a rather different prediction from the standard account of variable binding. Notice that in the sentence in (57) the two pronouns are not "merged" until we reach the clause at which variable-binding takes place. This means that if we have two unbound pronouns, as in (61), the two pronouns cannot correspond to the same variable:

(61) The woman who loves him hates his mother.

Of course they can "accidentally" pick out the same person (the meaning of (61) will be a function of type $\langle e, \langle e, t \rangle \rangle$ and so the hearer may apply it to two individuals to yield a proposition, and the two contextually salient individuals can happen to be the same). But they cannot correspond to the same variable in the way that the two pronouns in (53) can where one pronoun in essence binds the other. It is only in a case where a sentence like (61) is further embedded that these two pronouns can "merge". This is different than the standard account of variable binding. Since pronouns are literally just free variables, and since variables are meaningful objects, there is nothing in the standard account which keeps the two pronouns in (61) from translating as the same variable. And, since variables literally have meanings, they will then be the "same" just as the two pronouns in (53) are. It is, unfortunately, difficult to come up with definitive evidence for one approach or the other here, but this is a place where one might look to find evidence to choose between the No Essential Variables approach and an approach making use of variables.

6. Bach-Peters' Sentences Reconsidered

We are now in a position to explain the facts noted at the outset regarding Bach-Peters' sentences; their interaction with Weak Crossover is immediately accounted for without positing a level of Logical Form. Before considering the Weak Crossover cases, however, let us first return to a well-formed Bach-Peters' sentence such as (2):

(2) The man_i who loves her_j kissed his_i wife_j.

This is straightforward and can be directly interpreted without recourse to any level "unpacking" the paycheck pronoun. Informally, the meaning and semantic composition of this can be represented as in (62):

(62) the man who z -loves f z -kissed the wife-function (where f picks up the contextually salient function *wife'*)

Of course I have used a free variable f here, but this is for expository purposes only. The actual meaning will turn out as in (63); the interested reader can verify that the apparatus given above yields this meaning:

(63) $\lambda f[z(\text{kiss}')(\text{wife}')(\iota x[\text{man}'(x) \ \& \ z(\text{love}')(f)(x)]]]$

(It should be noted that (63) takes *wife'* to be a function of type $\langle e, e \rangle$. The exact treatment of such relational nouns and of the genitive NP in *his wife* is not crucial to anything here. However, for convenience I am assuming that the lexical meaning of a relational noun like *wife* is actually a two-place relation and is thus of type $\langle e, \langle e, t \rangle \rangle$. I am further assuming that any lexical item with this meaning can type-shift into a meaning of type $\langle e, e \rangle$; I will call this *wife*₂'. *wife*₂' maps an individual x into an

individual y just in case $wife_1'(x)$ yields a function which characterizes the singleton set $\{y\}$. Thus the item *wife'* in (63) is actually $wife_2'$). Thus (2) does not strictly speaking denote a proposition since it contains an "unbound" paycheck pronoun; it thus denotes a function from functions of type $\langle e, e \rangle$ to propositions, but it can be contextually applied to any salient function such as the wife-function.

However, it was noted in Sec. 2 that the paycheck pronoun in a Bach-Peters sentence behaved as if it had a fuller representation with respect to Weak Crossover, and thus, in contrast to (2), (9) is impossible:

(9) *The man_i who she_j loves kissed his_j wife_j.

At first glance, then, the interaction of paycheck pronouns with Weak Crossover appeared to support not only the complex meaning analysis of paycheck pronouns, but indeed a stronger version of it: a complex representation analysis. But in fact, the conclusion that the paycheck pronoun has a complex representation follows only if Weak Crossover itself is a constraint on representations. In the account of variable-binding here, however, it is a constraint on how meanings may be combined.

In essence, Weak Crossover is a consequence of the fact that the only variable-binding operation is the generalized z operation given in (29), and this will always bind a pronoun to a higher argument. I should point out that this is somewhat of a stipulation; it follows from nothing deep about this system that there is no backwards binding, for one could as well formulate a "backwards" version of z - call it s as given in (64): (note that this is essentially the S combinator of Curry and Feys (1958); my rationale for labelling the variable-binding operation z is precisely because it is a backwards version of S):

(64) for any function F of type $\langle e, \langle X, Y \rangle \rangle$, $s(F)$ is a function of type $\langle e, \langle \langle e, X \rangle, Y \rangle \rangle$,
 where $s(F) = \lambda x[\lambda g[F(x)(g(x))]]$ (for g a variable of type $\langle e, X \rangle$)
 (intuitively: if, for example, the mother-function s -loves x then x 's mother loves x)

The fact that the absence of s follows from nothing deep in this system may, in fact, not be a terrible result, since there are cases which exhibit Weak Crossover violations. However, I will leave open the question of whether this means that s sometimes applies. If s in fact never does, then hopefully this will eventually follow from some deeper principle.

It is thus the existence of z and the non-existence of s which gives rise to Weak Crossover effects. To demonstrate, we can contrast a well-formed relative clause like (65) with a Weak Crossover-violating one such as (66):

(65) the man_i who_j loves his_j mother
 (66) the man_i who_j his_j mother loves

Assume that the meaning of the relative pronoun *who* is such that it wants to combine with a property P and a property Q to give $\lambda x[P(x) \ \& \ Q(x)]$. We thus need only consider the derivation of *loves his mother* in (65) and *his mother loves* in (66). (65) is straightforward: *his* denotes the identity function which composes with *mother'* to yield *mother'* (again, I am assuming that *mother* has shifted to denote a function of type $\langle e, e \rangle$), and this occurs as argument of $z(\text{love}')$. Thus *loves his mother* denotes the property $\lambda x[\text{loves}'(\text{mother-of}(x))(x)]$ and this occurs as argument of the relative pronoun. (66), however, could not be derived without making use of the illicit operation s . If there were such an operation, then (66) could be derived in exactly the way relative clauses

with gaps are normally derived. To pick one possibility, let us follow the general line in Steedman (1987) according to which the subject NP is always type-lifted and composes with an incomplete VP. Then the meaning of the subject NP - which is *mother'* - can type-lift to be the generalized quantifier $\lambda T[T(\text{mother}')]$ (for T a variable of type $\langle\langle e, e \rangle, t \rangle$) and this composes with $s(\text{love}')$. This is illustrated in (67):

$$\begin{aligned} (67) \quad \text{lift}(\text{mother}') \circ s(\text{love}') &= \lambda T[T(\text{mother}')] \circ \lambda x[\lambda f[\text{love}'(x)(f(x))]] \\ &= \lambda x[\lambda T[T(\text{mother}')](\lambda f[\text{love}'(x)(f(x))])] = \lambda x[\lambda f[\text{love}'(x)(f(x))](\text{mother}')] \\ &= \lambda x[\text{love}'(x)(\text{mother}'(x))] \end{aligned}$$

Similar remarks hold for other Weak Crossover contrasts, such as those in (68):

- (68) a. Every man_i loves his_i mother.
b. His_i mother loves every man_i.

(68a) is of course straightforward; it simply involves applying $z(\text{love}')$ to the mother-function, and this occurs as argument of the subject. If s existed, then (68b) could also be derived by whatever mechanism is allowed for 'wide-scope objects in general. For example, it was noted above that objects can receive wide scope if a transitive verb first composes with a generalized quantifier subject, and then occur as argument of the object. Such a mechanism, then, combined with s , would allow for (68b). The derivation would proceed exactly as illustrated above in (67), and then the meaning of *his-mother-love* would be the argument of the object.

Given that Weak Crossover is a consequence of the fact that the only operation to effect variable binding is z and that this binds a pronoun to a higher argument, the interaction of Bach-Peters' sentences and Weak Crossover now follows without any need to "unpack" the paycheck pronoun into a complex representation or even a complex meaning. So, for example, consider again (9). To derive this in a fashion analogous to the way that the good sentence (2) was derived, we would need to make use of the illicit s operation; its meaning would have to be composed as illustrated very informally as in (69) or, somewhat more formally, as in (70):

- (69) the man who f $s(\text{loves})$ $z(\text{kissed})$ the wife function
(70) $\lambda f[z(\text{kiss}')(\text{wife}')(\lambda x[\text{man}'(x) \ \& \ s(\text{love}')(x)(f)])]$

Thus the impossibility of (9) as a Bach-Peters' sentence follows in the same way that any other Weak Crossover violation does - this would have to involve the "backwards binding" operation s . but since we have recast Weak Crossover as a constraint on how meanings are combined, the paycheck pronoun in a Bach-Peters' sentence needs no complex representation, and so no level of LF is needed to account for the facts.

The interaction of Weak Crossover and functional gaps is quite parallel. Thus May (1985), using somewhat different terminology, noted that while (71) can have the functional reading, (72) cannot:

- (71) Who does every Englishman love?
(72) Who loves every Englishman?

Chierchia (1991) further showed that this reduced to a Weak Crossover violation, and used this fact to argue that the functional gap has a complex representation containing an instance of the variable bound by *every Englishman*. The details of Chierchia's

representations need not concern us here; suffice it to note that if the functional reading involves a trace in subject position which has a complex representation such as $t_{f(x)}$ then the object NP *every Englishman* (or, its trace after QR) is not in a position to bind the index of the subject trace. (It is, incidentally, rather complicated to formulate this all in terms of representations at some level, and this fact could be taken to suggest that the representational account is fundamentally incorrect and that the complications are artefacts of this account.) Under the present proposal, the impossibility of a functional reading for (72) is exactly parallel to account discussed above. The only way that this could have a functional reading is by invoking *s*. I will not give a detailed account of just how this reading could be derived with such an operation, as it involves further questions concerning the treatment of quantified NPs in object position and the treatment of subject "gaps". Suffice it to say that the mechanics proposed here do not allow for a functional reading for (72), which is exactly the result that we would like.

In sum, then, the No Essential Variables approach provides a straightforward account of the interaction of Bach-Peters' sentences with Weak Crossover. This account does not rely on unpacking the paycheck pronoun into a complex meaning and - even more importantly - it need not be unpacked into a complex representation. The facts surrounding Bach-Peters' sentences are thus quite compatible with the claim that surface structures directly receive a model-theoretic interpretation.

*This paper incorporates parts of an earlier paper "Functional Questions, Paycheck Pronouns, and Variable Binding" which was delivered at the ASL/LSA Conference, Santa Cruz, July, 1991. I am grateful for helpful comments from the audience there, and also to my class at the 1991 LSA summer institute. I also thank Sam Bayer, David Dowty, Mark Hepple, Hans Kamp, Barbara Partee, and Matthew Stone for helpful discussion on much of this. This research was supported by NSF Grant BNS-9014676.

References

- Chierchia, Gennaro, 1984, *Topics in the Syntax and Semantics of Infinitives and Gerunds*, Ph.D. Dissertation, University of Massachusetts. Distributed by GLSA, Linguistics Department, University of Massachusetts, Amherst.
- Chierchia, Gennaro, 1990, "Anaphora and Attitudes De Se", in R. Bartsch et al. (eds.), *Language in Action*, Dordrecht: Foris.
- Chierchia, Gennaro, 1991, "Functional WH and Weak Crossover", *Proceedings of the Tenth Annual West Coast Conference on Formal Linguistics*, Stanford: CSLI Publications, 1991.
- Cooper, Robin, 1979, "The Interpretation of Pronouns" in F. Heny and H. Schnelle (eds.), *Syntax and Semantics 10*, New York: Academic Press, 61-92.
- Curry, Haskell and Robert Feys, 1958. *Combinatory Logic*. Amsterdam: North-Holland Publishing Co.
- Engdahl, Elisabet, 1986, *Constituent Questions*. Dordrecht: D. Reidel Publishing Co.
- Ginzburg, Jonathan, 1992. *Questions, queries, and facts: a semantics and pragmatics for interrogatives*. Ph.D. Dissertation, Stanford.
- Groenendijk, Jeroen and Martin Stokhof, 1982, "Semantic Analysis of WH-Complements", *Linguistics and Philosophy* 5, 175-223.

- Groenendijk, Jeroen and Martin Stokhof, 1983. "Interrogative Quantifiers and Skolem Functions" in K. Ehlich and H. van Riemsdijk (eds.), *Connectedness in Sentence, Discourse and Text*, Tilburg Studies in Language and Literature 4, Tilburg University.
- Hepple, Mark, 1991, *The Grammar and Processing of Order and Dependency: A Categorical Approach*, Ph.D. Dissertation, University of Edinburgh, 1990.
- Jacobson, Pauline, 1977, *The Syntax of Crossing Coreference Sentences*. Ph.D. Dissertation, University of California, Berkeley. Published by Garland Publishing Inc., 1981; also distributed by Indiana University Linguistics Club, 1979.
- Jacobson, Pauline, 1991, "Functional Question, Paycheck Pronouns, and Variable Binding", presented at the ASL/LSA Conference, Linguistic Society of America Summer Institute, Santa Cruz, July, 1991.
- Jacobson, Pauline, to appear, "Raising Without Movement" in R. Larson et al. (eds.), *Control and Grammatical Theory*, Dordrecht: Kluwer.
- Karttunen, Lauri, 1971, "Definite Descriptions with Crossing Coreference", *Foundations of Language* 7, 157-182.
- Karttunen, Lauri, 1977, "The Syntax and Semantics of Questions", *Linguistics and Philosophy* 1, 3-44.
- Landman, Fred and Ieke Moerdijk, 1983, "Compositionality and the Analysis of Anaphora", *Linguistics and Philosophy* 6, 89-114.
- May, Robert, 1985, *Logical Form: Its Structure and Derivation*, Cambridge: MIT Press.
- Oehrle, Richard, 1990, "Categorical Frameworks, Coordination, and Extraction", in *Proceedings of the Ninth Annual West Coast Conference on Formal Linguistics*. Stanford: CSLI Publications, 411-425.
- Partee, Barbara and Mats Rooth, 1983, "Generalized Conjunction and Type Ambiguity", in R. Baurle, C. Schwarze, and A. von Stechow (eds.), *Meaning, Use and Interpretation of Language*, Berlin: de Gruyter, 361-383.
- Quine, W. V., 1966, "Variables Explained Away", in *Selected Logic Papers*, New York: Random House, 227-235.
- Rohrbacher, Bernhard, 1991, "Crossing Coreference Sentences: No Evidence for Pronominalization", to appear in *Proceedings of the Eastern States Conference on Linguistics*, 1991.
- Steedman, Mark, 1987. "Combinatory Grammars and Parasitic Gaps", *Natural Language and Linguistic Theory* 5, 403-440.
- Szabolcsi, Anna, 1987, "Bound Variables in Syntax: Are There Any?" in J. Groenendijk et al. (eds.), *Proceedings of the 6th Amsterdam Colloquium*, ITLI.
- Szabolcsi, Anna, 1991, "Combinatory Categorical Grammar and Projection from the Lexicon", in I. Sag and A. Szabolcsi (eds.), *Lexical Matters*. Stanford: CSLI Publications.

On Properties of the Zielonka-Lambek calculus

Theo M.V. Janssen

Dept. Mathematics and Computer Science - University of Amsterdam

Abstract

We will argue that it is interesting to use the Zielonka calculus as a method to investigate categorial grammar (the Zielonka calculus is equivalent to the Lambek calculus). Using the Zielonka calculus we immediately find a new invariant: the $/-$ invariant. A new notation for category names will be introduced. This notation used with the Zielonka calculus gives us two other invariants: the category-balance and the $/$ -balance. Furthermore some new proofs for known results and some new results are obtained.

1. The Zielonka calculus

We first present the Zielonka calculus in the notation where $X \setminus Y$ expects an X from the left. It is described by 2 application rules, 4 lifting rules and two rule scheme's.

A rule $\alpha \rightarrow \beta$ should be read as follows. If we have an expression of category α then that expression is also of category β .

$$A/: X/Y \ Y \rightarrow X$$

$$A\setminus: Y \ YX \rightarrow X$$

$$Z1: X \rightarrow Y/(X \setminus Y)$$

$$Z2: X \rightarrow (Y/X) \setminus Y$$

$$Z3: X/Y \rightarrow (X/Z)/(Y/Z)$$

$$Z4: YX \rightarrow (Z \setminus Y)(ZX)$$

The rule scheme's are as follows

$$S1: \text{If } X \rightarrow Y \text{ then } (X/Z) \rightarrow (Y/Z)$$

$$S2: \text{If } X \rightarrow Y \text{ then } (ZX) \rightarrow (ZY)$$

$$S3: \text{If } X \rightarrow Y \text{ then } (Z/Y) \rightarrow (Z/X)$$

$$S4: \text{If } X \rightarrow Y \text{ then } (YZ) \rightarrow (XZ)$$

The rule schemes in fact say that if $X \rightarrow Y$, then this category change is correct in any positive position, and the reversed change in any negative position. We will give category symbols a + superscript in positive positions, and a - in negative positions.

2. Count Invariants

In this section two invariants of the Zielonka-Lambek calculus are proved. The first is well known, the proof given here is new : it follows immediately from the rules in the Zielonka notation. The second invariant is from the author. Finally we investigate the relation between the invariants.

Definition

$C\text{-Count}(X) = \#(\text{occurrences of } C^+ \text{ in } X) - \#(\text{occurrences of } C^- \text{ in } X).$

Theorem If $X \ Y \Rightarrow^* Z$ then for all C : $C\text{-count}(X^+) + C\text{-Count}(Y^+) = C\text{-count}(Z^+).$

Proof. The equality holds for the application rules. The other rules introduce or remove basic categories in pair with opposite signs.

Definition

$\text{slash-Count}(X) = \#(\text{occurrences of } /^+ \text{ and } \setminus^+ \text{ in } X) - \#(\text{occurrences of } /^- \text{ and } \setminus^- \text{ in } X).$

Note that the directionality of the slashes is not taken into consideration.

Theorem invariant of slash-count

If $X Y \Rightarrow^* Z$ then $\text{slash-count}(X^+) + \text{slash-count}(Y^+) = \text{slash-count}(Z^+) + 1$

Proof The equality holds for the application rules. The other rules introduce or remove slashes in pairs, where the elements of the pairs have opposite signs.

Theorem $\sum_C \text{C-count}(X) = \text{slash-count}(X) + 1$.

Proof (M. Trautwein)

By induction. The equality holds for basic categories. For compound we have:

$$\sum_C \text{C-count}(X/Y) = \sum_C \text{C-count}(X) - \sum_C \text{C-count}(Y) = \text{slash-count}(X) + \text{slash-count}(Y) + 2 = \text{slash-count}(X/Y) + 1$$

Theorem If for all C : $\text{C-count}(X^+) + \text{C-count}(Y^+) = \text{C-count}(Z^+)$, then $\text{slash-count}(X^+) + \text{slash-count}(Y^+) = \text{slash-count}(Z^+) + 1$.

Proof sketch. Application of the previous theorem.

Remark. Van Benthem (1991) p. 73 mentions a case where the slash-count rejects a sequent that would pass the category count.

3. The Notation

Starting point is the notation that A/B expect a B from the right, and $A \setminus B$ expects an A from the left. Then we mark occurrences of categories and slashes in positive positions with $+$, and in negative positions with $-$.

Furthermore in negative positions the order is reversed.

Definition

Let α and β be compound category symbols. Then the new notation is defined by :

$$\begin{aligned} (\alpha/\beta)^+ &= (\alpha)^+ /^+ (\beta)^- & (\alpha/\beta)^- &= (\beta)^+ /^- (\alpha)^- \\ (\beta \setminus \alpha)^+ &= (\beta)^- \setminus^+ (\alpha)^+ & (\beta \setminus \alpha)^- &= (\alpha)^- \setminus^- (\beta)^+ \end{aligned}$$

$(\alpha)^+ = \alpha^+$, $(\alpha)^- = \alpha^-$ if α is a basic category.

Examples in old notation

$C \setminus E$

$A/(B/C)$

$A/((B/D)/(C/E))$

the same ones in new notation

$C^- \setminus^+ E^+$

$A^+ /^+ (C^+ /^- B^-)$

$A^+ /^+ ((C^+ /^+ E^-) /^- (D^+ /^- B^-))$

There is some redundancy in this notation (the directionality of $/$ can be calculated from the signs).

Using the new notation, the application rules become

A1: $X^+ /^+ Y^- \rightarrow X^+$

A2: $Y^+ Y^- \setminus^+ X^+ \rightarrow X^+$

The rule Z1..Z4 become:

E1. $X^+ \rightarrow (Y^+ /^+ (Y^- \setminus^+ X^+))$

E2. $X^+ \rightarrow ((X^+ /^- Y^-) \setminus^+ Y^+)$

E3. $X^+ /^+ Y^- \rightarrow (X^+ /^+ Z^-) /^+ (Z^+ /^- Y^-)$

E4. $Y^- \setminus^+ X^+ \rightarrow (Y^- \setminus^+ Z^+) \setminus^+ (Z^- \setminus^+ X^+)$

We will not read these rules as in the original system (changing the category of an expression), but as as rewriting rules for category names. Then these rules cover the

effect obtained by the rule schemes: in all positive positions in category names these changes can be performed. The rules are labelled E1 etc. because they expand the category name. Note that all expansion rules have the form that $/^+$ is replaced by some continuous string of the form ww^R with a $/^+$. The other effect of the schemes is obtained by adding rewriting rules for negative positions. They are formed by replacing a rule $X^+ \rightarrow Y^+$ by $Y^- \rightarrow X^-$.

$$R1. ((X \setminus^+ Y^+)/Y^-) \rightarrow X^-$$

$$R2. (Y \setminus^-(Y^+/X^-)) \rightarrow X^-$$

$$R3. (Y^+/Z^-)/(Z^+/X^-) \rightarrow Y^+ / X^-$$

$$R4. (X \setminus^-(Z^+)) \setminus^-(Z^-/Y^+) \rightarrow X^- \setminus^+ Y^+$$

The rules are labelled R1, .. because they reduce the length of the category name.

The effects of application of a rule is local, instead of the global effects that arise with rule schemes Z3 and Z4. Some continuous substring of the of the form ww^R with a $/^+$ is replaced. Using the new notation, we have formulated the Zielonka calculus by means of a finite set of rewriting rule schemes (in spite of the fact that no finite axiomatization is possible for the Zielonka calculus). In the sequel we will use $\Rightarrow$ and $\Rightarrow^*$ for rewriting category names in the same way as they are used in formal language theory.

4 Balance

In this section we will consider two invariants of the Zielonka-Lambek calculus. The first one is known (Roorda 1991), but here gets a new proof. The other is an immediate consequence of the $/$ -count.

Definition Balance

A string s of signed symbols is called balanced if 1. or 2. holds (w is a balanced sequence, σ an arbitrary symbol).

$$1. s = \sigma^+ \sigma \text{ or } s = \sigma \sigma^+.$$

$$2. s = \sigma^+ w \sigma, s = w \sigma^+ \sigma, s = \sigma^+ \sigma w, s = \sigma w \sigma^+, s = w \sigma \sigma^+, s = \sigma \sigma^+ w$$

A string s forms a **category balance** if the string obtained by deleting all other symbols than categories is balanced. A **slash-balance** is defined analogously.

Informally, a sequence is balanced if each occurrence of σ^+ can be connected with an occurrence by means of lines in the lower halfplane such that no intersection arises.

Examples: $A^+ B^- B^+ A^-$ is balanced, whereas $A^+ B^- A^- B^+$ is not.

Theorem (Roorda balance)

If $X \Rightarrow^* Z$, then the basic symbols in $X^+ Z^-$ form a category balance.

If $X Y \Rightarrow^* Z$ then the basic category symbols in $X^+ Y^+ Z^-$ form a category balance.

Proof sketch

1) If $U/V \Rightarrow_{A/} Z$ then $U^+ V^- V^+ U^-$ forms a category balance. This holds U, V being basic symbols, and for U, V being compound symbols.

2) If $X \Rightarrow^* X'$ and $X'^+ Y^+ Z^-$ forms a category balance, then also $X^+ Y^+ Z^-$ forms a category balance.

In case $X \Rightarrow^*_R X'$ this is trivial: the extra symbols within X can be linked internally.

In case $X \Rightarrow^* E X$, then we proof 2) with induction to the length of the expansion. Suppose $X = X_1 X_2$ and $X_1 X_2 \Rightarrow^* E X_1 C C X_2$ and we have a balance for $(X_1 C C X_2)^+ Y^+ Z^-$. Then we can make a balance for $(X_1 X_2)^+ Y^+ Z^-$: each of the two C's is connected with some other C's somewhere in the string. Connect these two other occurrences and remove the substring CC. Then we have a balance again. In a picture:

$(X_1 C C X_2)^+ Y^+ Z^-$ becomes $(X_1 X_2)^+ Y^+ Z^-$

here the arrows point to appropriate symbols within the compound category names X and Y

If the expansion rule introduces a longer string, it can be removed in this way symbol for symbol, (inside out). Then a balance for $(X_1 X_2)^+ Y^+ Z^-$ is obtained.

3) For $Y \Rightarrow^* Y'$ we proceed in exactly the same way.

4) For $X \Rightarrow^* Z$, we proceed analogously. For $X \Rightarrow X$ we have the balance $X^+ X^-$. If $X \Rightarrow X' \Rightarrow^* Z$ and $X'^+ Z^-$ is balanced, then also $X^+ Z^-$ is balanced as can be proven in the same way as in 3)

Remark. Here we found the balance invariant as a natural consequence of the balance in the rules. In Roorda 1991 the occurrences of signs, and the order of the basic categories, can only be understood if one knows about Girards theory of proof nets.

Theorem (/ -balance)

If $X \Rightarrow^* Z$ then the slashes in $X^+ Z^-$ form a slash balance.

If $X Y \Rightarrow^* X' Y' \Rightarrow_{A/} Z$ then the slashes in $X^+ Y^- / Z^-$ form a slash balance.

If $X Y \Rightarrow^* X' Y' \Rightarrow_{A\backslash} Z$ then the slashes in $\backslash X^- Y^+ Z^-$ form a slash balance.

Proof (Aarts)

We already know that the in the string the number of +slashes equals the number of --slashes. Any such string can be balanced. We prove the result easily with induction. In a string of one +slash and one - slash there obviously is a link. If the string is longer, there is at least one pair of a consecutive +slash and - slash. These two have to be linked, and when these are left out, the remainder of the string can be balanced.

5. Non-Associative Zielonka calculus

The non-associative Zielonka calculus (NAZ) is obtained when the rules Z3 and Z4 are omitted. In our notation this calculus consists of

$$A1: X^+ / Y^- Y^+ \rightarrow X^+$$

$$A2: Y^+ Y^- \backslash X^+ \rightarrow X^+$$

$$E1: X^+ \rightarrow (Y^+ / (Y^- \backslash X^+))$$

$$E2: X^+ \rightarrow ((X^+ / Y^-) \backslash Y^+)$$

$$R1: ((X^- \backslash Y^+) / Y^-) \rightarrow X^-$$

$$R2: (Y^- \backslash (Y^+ / X^-)) \rightarrow X^-$$

Normal form (Buzkowski 1986).

If $X Y \Rightarrow^* Z$ in NAZ then there is a derivation in which first only reduction rules are applied to X and Y, then application is performed, and then the expansion rules are

applied.

Application

Based upon this normal form a parser for the non-associative Zielonka calculus is made by C. Trautwein (1991). The procedure which performs the reductions is a procedure with backtracking, and seems to take exponential time. Below we will show that recognizing a possible reduction is polynomial, but that the number of reduced forms of a category is not limited by a polynomial.

Theorem

A substring of a category name to which a reduction rule can be applied, can be found in $O(n^2)$.

Proof sketch

We have to find a substring of the form $(Y^+ / (Y^- \text{ or } / Y^-) \backslash Y^+)$. Remind that Y^- denotes the string Y in reversed order, with changed signs. So the substrings are of the form ww^R , and such a pattern can be described by a contextfree grammar. They can be recognized in $O(n^2)$.

Remark This observation also holds for reductions in the associative Zielonka calculus. In the original notation for categories the pattern for reductions would be ww , and that is not context free.

Observation. There is no polynomial limiting the number of reduced formula's.

Consider $C_1 = (B \backslash A(1)) \backslash A(2) \backslash \dots \backslash A(n) / A(n) \dots A(2) A(1)$. So C_1 has n occurrences of A . Then $A^{(n)}$ can be removed, and then $A^{(n-1)}$, etc. So there are $O(n)$ reduced formulas. Let $C_2 = (C_1^+ / (C_1^+ / B^-))$. Here all combinations of reductions in the two occurrences of C_1 are possible, so $O(n^2)$ reducts. And $C_3 = (C_2^+ / (C_2^+ / B^-))$ has $O(n^4)$, etc.

Given two categories X and Y , one may ask whether they are conjoinable, i.e. whether there is a Z such that $X \Rightarrow^* Z$ and $Y \Rightarrow^* Z$. In the Lambek-Zielonka calculus, the decidability of this question is unknown (v. Benthem 1991, p. 76). Below we show the decidability for the nonassociative Zielonka calculus.

Theorem.

If there is a Z such that $X \Rightarrow^* Z$ and $Y \Rightarrow^* Z$, then there is a Z' such that $|Z'| \leq |X| + |Y|$ and $X \Rightarrow^* Z'$ and $Y \Rightarrow^* Z'$.

Proof sketch

Suppose that there such a Z . Consider a derivation in normal form, so we have

$X \Rightarrow^*_R X' \Rightarrow^*_E Z$. All occurrences of basic category symbols in X' can be connected with an occurrence in X . Each occurrence of a basic symbol in Z is either connected with an occurrence in X' (and thus in X), or is introduced by an expansion rule. Since no reduction rules are applied after X' , each occurrence in Z introduced by an expansion rule has a counterpart in X which is introduced by the same rule (but occurs with different sign). The same story holds for derivations starting from Y . If in both derivations an

occurrence of a basic category symbol (and its counterpart) is due to an expansion rule, then we can remove in both reductions the introduction of this symbol. We end up with a category name in which all occurrences of category symbols are connected with occurrences in X or Y .

Remark The decision procedure based upon this theorem will be non-polynomial because of the number of reduced formulas.

6. Conclusion

The new notation transformed the Zielonka rules to rules involving a small part of the category name. In this way their effect are easier to describe and their properties are easier to discover.

Acknowledgements.

I thank Erik Aarts, Johan van Benthem, Michael Moortgat, and Mart Trautwein for their remarks and suggestions during my work on this subject.

References

- Benthem van, J. , 1991,
 'Language in action: categories, lambdas and dynamic logic', North Holland, Amsterdam.
- Buzkowski, W. 1986,
 'Generative capacity of Nonassociative Lambek calculus, Bull. Pol. Acad. of Sc., math, vol 34, pp. 507-516.
- Roorda, D., 1991,
 'Resource Logics: Proof theoretical investigations', Unpublished Ph. D. thesis, University of Amsterdam,
- Trautwein, K., 1991,
 'Een parseeralgoritme voor de niet associatieve Lambek calculus, en de plaats ervan in de wereld', unpublished master thesis, univ. of Amsterdam.

Language Invariants

Edward L. Keenan & Edward P. Stabler
UCLA

In this paper we present a conception of language structure and language invariant which is new in the linguistic literature, though it is based on standard algebraic methods. This conception enables us to characterize language invariants over languages which themselves are structurally quite distinct; it suggests a new understanding of the role of grammatical categories and grammatical morphemes in language, and it provides several new concepts in terms of which we formulate hypotheses concerning the form of human languages. We focus here on linguistic invariants. Keenan (1987, 1991) provides evidence that the syntactic factors constraining the distribution of anaphors in different languages are structurally different. The approach to language invariants formalized here is an attempt to consider the nature of a theory of language which accepts this variation at face value.

The paper is presented in three parts. §1 contains our basic ideas. §2 illustrates these ideas with two simple artificial languages. We present a language independent semantic definition of anaphor and show that these simple languages identify their anaphors differently. In each language, however, the set of syntactic contexts in which anaphors occur is a structural invariant, as is the structural relation that must hold between anaphors and antecedents. §3 presents some mathematical results concerning our formal framework.

1. A relational conception of structure and invariance. Informally, we think of a language as given by a lexicon *LEX* which is a set of categorized expressions, a set *F* of generating functions, and a set Φ of filters. The language is the set of expressions which are built from *LEX* by recursively applying the functions and which satisfy the conditions given by the filters.

A standard view identifies the structure of a language, its strong generative capacity, with the set of derivation trees, where the trees record the ways expressions of the language are built. That is, the trees indicate which lexical items were used, which functions applied, and what their values were (Montague, 1970). Given a language to be studied, this view leads one to ask "For each expression σ of the language, what is the structure of σ ?" By contrast, our proposal is based on the relation "has the same structure as." Given a language to be studied, we are led to ask "When do expressions σ, τ have the same structure?" With this conception of structure, it turns out that a given set of expressions can be defined in various ways that differ in the derivation trees assigned to the strings but not in structure.

More precisely, given a language system *L* with lexicon *LEX*, generating functions $\mathcal{F}$, and filters Φ , we consider the functions from expressions to expressions which preserve *LEX*, $\mathcal{F}$, and Φ (in a sense of "preserve" specified below). These functions, which we call **homs** or **structure (preserving) maps**, are just the functions which respect the derivation trees for the expressions: for *h* a hom and σ an expression, the derivation trees for σ and $h(\sigma)$ are isomorphic.¹ Expressions σ and τ have the same structure, they are (syntactically)

isomorphic, $\sigma \cong \tau$, iff each can be mapped to the other by a hom. A set D of expressions is a (syntactic) invariant of L iff whenever $\sigma \cong \tau$ then $\sigma \in D$ iff $\tau \in D$. Such sets D cannot distinguish among expressions with the same syntactic structure. In a phrase,

Structure is what the structure preserving functions preserve.

Thus the invariants of L are those sets that can be "identified" in terms of the grammar of L . It will be useful to both specialize and generalize this notion. Specializing first, an expression σ is identified by the grammar if $\{\sigma\}$ is a syntactic invariant. That is, the only expression with the same structure as σ is σ itself. In a phrase,

Grammatical morphemes are fixed points of syntactic isomorphisms.

Generalizing, a set D is said to be K -invariant, for K a set of expressions, if L identifies the members of D in terms of the members of K . Formally, a hom h preserves a set K of expressions iff for all $k \in K$, $h(k) \in K$. Expressions σ, τ are isomorphic mod K , $\sigma \cong_K \tau$, iff each can be mapped to the other by a hom which preserves K . And a set D is called K -invariant iff $\sigma \cong_K \tau$ implies $\sigma \in D$ iff $\tau \in D$. The invariants of L as defined above are just the K -invariant sets for $K = \emptyset$.

We are particularly interested in K -invariant sets where K is a finite subset of the lexicon LEX . Since even quite idiosyncratic properties of a finite set of lexical items can be learned, we shall say that a set D is **lexically identifiable** iff it is K -invariant for some finite subset K of LEX . What this means is that the grammar uniquely identifies D once certain lexical items are identified. The lexical items themselves may not be syntactically identifiable, though they may be semantically identifiable. For example, in the artificial language "Little English" given below, the set NP_{ana} of anaphorically interpreted expressions is lexically identifiable. NP_{ana} is infinite, but its membership is grammatically determined once its (finitely many) lexical members are given. The lexical members are identified by the semantic interpreting function, but they are not identifiable on purely syntactic grounds.

1.1. Language systems, structure maps, and invariants. Using $[A \rightarrow B]$ for the set of partial functions from A to B , and using $\mapsto$ for total functions, we define a **language system** L formally as a 5-tuple $\langle CAT, V, LEX, \mathcal{F}, \Phi \rangle$ where CAT and V are non-empty sets whose elements are called categories and vocabulary items respectively, and

$$\begin{aligned} LEX &\subseteq V \times CAT, \\ \mathcal{F} &\subseteq \bigcup_{n \in \mathbb{N}} [(V^* \times CAT)^n \rightarrow (V^* \times CAT)], \text{ and} \\ \Phi &\subseteq [(V^* \times CAT) \mapsto \{0, 1\}]. \end{aligned}$$

Note that lexical items (elements of LEX) are pairs $\langle s, C \rangle$, where $s \in V^*$, the set of strings of vocabulary items, and $C \in CAT$. Where $\sigma = \langle s, C \rangle$ is such a pair, we write σ_1 for the first coordinate, s , and σ_2 for the second coordinate, C . When $K \subseteq V^* \times CAT$, we write K_1 for $\{s \in V^* \mid \exists C \in CAT, \langle s, C \rangle \in K\}$.

We define $PH_L = \{\sigma \in CL[LEX, \mathcal{F}] \mid \text{for all } \phi \in \Phi, \phi(\sigma) = 1\}$. Often we omit subscripts L when no confusion results. Here $CL[LEX, \mathcal{F}]$ is the closure of LEX with

respect to $\mathcal{F}$, that is, the intersection of all subsets of $V^* \times CAT$ which include LEX and are closed with respect to all $F \in \mathcal{F}$. We let $PH_C = \{\sigma \in PH \mid \sigma_2 = C\}$. An expression in $CL[LEX, \mathcal{F}]$ passes the filters if each $\phi \in \Phi$ maps it to 1. If $\sigma = \langle s, C \rangle \in PH$, we say that σ is an **expression of category C** and that s is a **string of category C** . The language (weakly) generated by L is just the set PH_1 , that is, the set $\{s \in V^* \mid \exists C \in CAT, \langle s, C \rangle \in PH\}$.

A **structure preserving map** for L , a “hom,” is a total function h from $V^* \times CAT$ to $V^* \times CAT$ such that

1. h preserves LEX ,
i.e., if $\sigma \in LEX$ then so is $h(\sigma)$, and
2. h preserves $\mathcal{F}$,
i.e., if $F \in \mathcal{F}$ maps σ to τ then $h(\sigma) \in Dom(F)$ and F maps $h(\sigma)$ to $h(\tau)$ – that is, h commutes with F , and
3. h preserves the filters Φ ,
i.e., if $\phi \in \Phi$ maps σ to some value (1 or 0), then ϕ maps $h(\sigma)$ to that same value.

Then the set of structure maps is:

$$HOM_L = \{h \in [(V^* \times CAT) \mapsto (V^* \times CAT)] \mid h \text{ preserves } LEX, \mathcal{F}, \Phi\}.$$

For $\sigma, \tau \in PH$, $\sigma \cong \tau$ iff there exist $h, k \in HOM_L$ such that $h(\sigma) = \tau$ and $k(\tau) = \sigma$. For $\sigma, \tau \in PH^n$, we write $\sigma \cong \tau$ to mean there are $h, k \in HOM_L$ such that for all $1 \leq i \leq n$, $h(\sigma_i) = \tau_i$ and $k(\tau_i) = \sigma_i$. Subset R of PH^n is a **(syntactic) invariant** iff $\sigma \cong \tau$ implies $\sigma \in R$ iff $\tau \in R$. Equivalently, R is an invariant iff R is closed under isomorphism: $\sigma \in R$ implies $\tau \in R$ for all τ such that $\sigma \cong \tau$. An expression σ is invariant iff $\{\sigma\}$ is. We write INV^n for the set of syntactically invariant subsets of PH^n .

Generalizing, we say that for $K \subseteq PH$, expressions σ, τ are **K -isomorphic ($\cong_K$)** iff each is the image of the other under a hom which preserves K . Subset D of PH^n is **K -invariant** iff, whenever $\sigma \cong_K \tau$, then $\sigma \in D$ iff $\tau \in D$. Write INV_K^n for the set of K -invariant subsets of PH^n . Note that $INV^n = INV_\emptyset^n$ and that $K \subseteq K'$ implies $INV_K^n \subseteq INV_{K'}^n$.

2. Some simple languages.

2.1. Little English. As a first example, we consider a simple language defined with one filter, *CASE*, and a set of (categorized) expressions which, to guide the reader, we exhibit first in a standard context free format:

$$\begin{array}{ll}
 \text{(Categorial Rules)} & S \rightarrow NP \ P1 \\
 & P1 \rightarrow P2 \ NP \\
 & C \rightarrow \text{both } C \text{ and } C \quad \text{for all } C \in \{S, NP, P1, P2\} \\
 & C \rightarrow \text{either } C \text{ or } C \quad \text{for all } C \in \{S, NP, P1, P2\} \\
 & C \rightarrow \text{neither } C \text{ nor } C \quad \text{for all } C \in \{S, NP, P1, P2\} \\
 \text{(Lexical Rules)} & NP \rightarrow \text{john} \mid \text{bill} \mid \text{he} \mid \text{him} \mid \text{himself} \\
 & P1 \rightarrow \text{laughed} \mid \text{cried} \\
 & P2 \rightarrow \text{praised} \mid \text{criticized}
 \end{array}$$

The CASE filter will apply to the expressions generated to guarantee the correct distribution of "nominative" NPs like *he* and *neither he nor john*, as well as "accusative" NPs like *him*, *himself*, *both himself and john*.

In our formalism, Little English = $\langle CAT, V, LEX, \mathcal{F}, \Phi \rangle$ where

$$CAT = \{NP, S, P1, P2, CONJ\}$$

$$V = \{john, bill, he, him, himself, laughed, cried, praised, criticized, both, and, either, or, neither, nor\}$$

$$LEX = (\{john, bill, he, him, himself\} \times \{NP\}) \cup (\{laughed, cried\} \times \{P1\}) \cup (\{praised, criticized\} \times \{P2\}) \cup (\{and, or, nor\} \times \{CONJ\})$$

$$\mathcal{F} = \{F1, F2, AND, OR, NOR\}$$

$$\Phi = \{CASE\}$$

where,

$$F1(\langle s, NP \rangle, \langle t, P1 \rangle) = \langle s \ t, S \rangle$$

$$F2(\langle s, NP \rangle, \langle t, P2 \rangle) = \langle t \ s, P1 \rangle$$

$$\text{for } C \in \{NP, P2, P1, S\},$$

$$AND(\langle and, CONJ \rangle, \langle s, C \rangle, \langle s', C \rangle) = \langle both \ s \ and \ s', C \rangle$$

$$OR(\langle or, CONJ \rangle, \langle s, C \rangle, \langle s', C \rangle) = \langle either \ s \ or \ s', C \rangle$$

$$NOR(\langle nor, CONJ \rangle, \langle s, C \rangle, \langle s', C \rangle) = \langle neither \ s \ nor \ s', C \rangle$$

$$\text{for } \delta \in LEX, CASE(\delta) = 1,$$

$$\text{for } \delta = G(\eta, \sigma, \tau), \text{ any } G \in \{AND, OR, NOR\}, CASE(\delta) = CASE(\sigma) \wedge CASE(\tau),$$

$$\text{for } \delta = F1(\sigma, \tau), CASE(\delta) = 1 \text{ iff } \sigma \in NP_{nom},$$

$$\text{for } \delta = F2(\sigma, \tau), CASE(\delta) = 1 \text{ iff } \sigma \in NP_{acc},$$

and where,

$$NP_{nom} = CL[\{\langle john, NP \rangle, \langle bill, NP \rangle, \langle he, NP \rangle\}, \{AND, OR, NOR\}]$$

$$NP_{acc} = CL[\{\langle john, NP \rangle, \langle bill, NP \rangle, \langle him, NP \rangle, \langle himself, NP \rangle\}, \{AND, OR, NOR\}].$$

Note that this definition of $F1$ is intended to mean that the domain of $F1$ is the set of pairs $\langle \sigma, \tau \rangle$ where $\sigma \in (V^* \times \{NP\})$ and $\tau \in (V^* \times \{P1\})$, and where the action of $F1$ is as indicated. The other function definitions are to be read similarly.

Little English contains just finitely many lexical expressions, such as $\langle bill, NP \rangle$. The functions $\mathcal{F}$ allow us to generate infinitely many other expressions, such as $\langle both \ bill \ and \ john, NP \rangle$ and $\langle bill \ both \ praised \ john \ and \ laughed, S \rangle$. On the other hand, $\delta = \langle himself \ criticized \ john, S \rangle \notin PH$, since $CASE(\delta) = 0$. For the same reason *both himself and bill criticized john* is not a string of any category. Thus in this

particularly simple language the “correct” distribution of “anaphors” is guaranteed by the *CASE* filter. Also note that $\langle \text{both he and him}, NP \rangle \in PH_{NP}$ but it is not in NP_{nom} or NP_{acc} , so no *S* containing it passes the *CASE* filter.

Note finally that NP_{nom} and NP_{acc} are not categories; neither is a PH_C for any $C \in CAT$. In this sense, case is covert. Later we study Little Latin in which case is overt – present in the category system and morphologically explicit. Observe now some invariants of Little English:

Theorem 2.1. In Little English, every $h \in HOM$ preserves NP_{nom} , NP_{acc} , and PH_C for all $C \in CAT$.

So all structure maps for Little English map NP_{nom} to NP_{nom} , PH_{P_1} to PH_{P_1} , ... (For proofs we refer the reader to Keenan and Stabler, 1992.)

Corollary 2.2. Each of the sets mentioned in Theorem 2.1 is an invariant in Little English.

In fact Corollary 2.2 is an instance of the following more general theorem:

Theorem 2.3. For any language system, every set K of tuples of expressions which is preserved by all homs is invariant.

Our concern with the languages we study here is to determine their invariants. In this example the categories (i.e. the PH_C) are invariants, as are certain sets defined by the filters. But many other sets are invariant. A basic **universal invariant** is given by the relation *occurs in* defined as follows: Let L be a language system and $\beta, \delta \in PH$. Then, β occurs in δ iff either $\beta = \delta$ or for some $F \in \mathcal{F}$ and some $\sigma_1, \dots, \sigma_n \in CL[LEX, \mathcal{F}]$, $\delta = F(\sigma_1, \dots, \sigma_n)$ and β occurs in one of the σ_i .²

Theorem 2.4. For all L , $\{ \langle \beta, \delta \rangle \in PH_L^2 \mid \beta \text{ occurs in } \delta \}$ is an invariant of L .

In general it is not easy to determine whether a particular linguistic property is an invariant of a given language. E.g. the PH_C are not guaranteed to be invariant: language systems whose generating functions are defined with great generality may allow homs to map *NPs* to *Ss* and vice versa. The following theorem is useful in this regard (see also §3):

Theorem 2.5. For all language systems L , all n , and all $K \subseteq PH^n$, PH^n and $\emptyset$ are K -invariants and INV_K^n is closed with respect to arbitrary intersections, unions, and complements (relative to PH^n). So each INV_K^n is a complete (and in fact atomic) boolean algebra.

So Theorems 2.3 and 2.5 jointly tell us that $NP_{nom} - NP_{acc}$ is an invariant: if s is an *NP* which is nominative but not accusative then anything s is isomorphic to also has that property. Little English presents infinitely many such *NPs*: *he*, *both he and john*, *either both he and john or both he and bill*, ... Moreover, since *LEX* is preserved by all homs and thus is invariant, we have by Theorem 2.5 that $LEX \cap (NP_{nom} - NP_{acc}) = \{ \langle he, NP \rangle \}$ is invariant. In other words, $\langle he, NP \rangle$ is a fixed point in the grammar of this language. Similarly the property of being accusative but not nominative is invariant. For further operations which build K -invariant sets from others, see §3.

More challenging are sets that are semantically defined. Typically such sets are not invariants; they are not “coded” in the syntax. For example, in real English there is no

reason to think that sentences (1a,b) are structurally distinct, and thus each could be the image of the other under a hom (e.g., one that extended a map from *LEX* to *LEX* which interchanges *five* and *ten* and sends everything else to itself):

(1a) At least five but not more than ten students left early.

(1b) At least ten but not more than five students left early.

But (1b) is always false, whereas the truth of (1a) depends on how the world is. So the set of logically false sentences of English is not a grammatical invariant; it is not coded in the syntax of English. Similarly the set of pairs of sentences $\langle \sigma, \tau \rangle$ such that σ entails τ is not a structural invariant. We may have $\sigma \cong \sigma'$ and $\tau \cong \tau'$ when σ entails τ while σ' does not entail τ' . By contrast, the basic distributional properties of anaphors are generally coded in the syntax of languages even though anaphoricity is a semantic property. To see this we first give a properly semantic definition of anaphor.

In general a model (situation) $\mathcal{M}$ for any of the languages we discuss is standardly given by a pair $\langle E, m \rangle$, where E is a nonempty set of objects about which we think of ourselves as speaking, and m is a function which assigns denotations of an appropriate sort to lexical items. An interpretation $|\cdot|^\mathcal{M}$ of L relative to $\mathcal{M}$ is a function which extends m by assigning denotations to the syntactically complex expressions as a function of how they are built up.

For Little English, given a universe E , we think of *P1s* as denoting subsets of E , which are properties. We write simply P for $P(E)$, the power set of E . *P2s* denote binary relations over E (subsets of $E \times E$). We write R for the set of binary relations over E . *Ss* denote in $2 = \{True, False\}$. Finally, *NPs* denote functions that map $n+1$ -ary relations to n -ary ones, for all n . Thus they are (possibly partial) functions which map properties to truth values and binary relations to properties.

As regards interpretative constraints on lexical items, we require:

- (i) that $m(himself, NP) = SELF$, where $SELF$ is the function mapping each binary relation R to $\{a \mid \langle a, a \rangle \in R\}$, and
- (ii) when d is *john*, *bill*, *he*, or *him*, there is $b \in E$ such that $m(d, NP) = I_b$, the individual generated by b . (By definition I_b maps a property p to *True* iff $b \in p$ and I_b maps each binary relation R to $\{a \mid \langle a, b \rangle \in R\}$).

One verifies that for $C = NP, P2, P1$, or S , the denotation set for C (relative to E) is always a set with a boolean structure. The function $m(and, CONJ)$ maps any pair $\langle d, d' \rangle$ of possible C denotations to $d \wedge d'$, their boolean meet (greatest lower bound). And the function $m(or, CONJ)$ maps $\langle d, d' \rangle$ to $d \vee d'$, their boolean join (least upper bound). And $m(nor, CONJ)$ maps $\langle d, d' \rangle$ to $\neg(d \vee d')$, the complement of $d \vee d'$.

An interpretation $|\cdot|^\mathcal{M}$ of L relative to $\mathcal{M} = \langle E, m \rangle$ is a function which extends m to the complex expressions satisfying:

$$\begin{aligned} |F(\sigma, \tau)|^\mathcal{M} &= |\sigma|^\mathcal{M}(|\tau|^\mathcal{M}) && \text{for all } F \in \{F1, F2\}, \text{ and} \\ |G(\eta, \sigma, \tau)|^\mathcal{M} &= m(\eta)(|\sigma|^\mathcal{M}, |\tau|^\mathcal{M}) && \text{for all } G \in \{AND, OR, NOR\}. \end{aligned}$$

We can now define: for $\sigma, \tau \in PH_S$, σ entails τ iff for all models $\mathcal{M}$, if $|\sigma|^\mathcal{M} = True$ then $|\tau|^\mathcal{M} = True$. The purpose of this definition is simply to show that our model structure is rich enough to support an entailment relation of a standard sort.

Now, leading up to a semantic definition of anaphor, consider first *NPs X* in real English which may occur in (2a,b):

- (2a) John criticized X (b) Bill praised X

For some choices of X , such as *Sam* or *most but not all of the teachers in Boston*, the two S s above must have the same truth value if the objects John criticized are exactly those which Bill praised. Such NPs will be interpreted by functions F which satisfy the following invariance condition (Keenan, 1987), where aR is the property $\{b \mid aRb\}$:

[A] $aR = bS \Rightarrow (a \in F(R) \text{ iff } b \in F(S)).$

However, substituting expressions like *himself*, *both himself and Sam*, *everyone but himself* for X in (2a,b), we see that the two S s may have different truth values even if the objects John criticized are exactly those that Bill praised. For example, if those objects are just Sam, Bill, and Mary then *john criticized himself* is false but *bill criticized himself* is true. Semantically then, whether these NPs put an object b in the set they associate with a relation R depends both on what objects b bears R to and on what object b is. The functions F which interpret these NPs then only satisfy the following weaker invariance condition (Keenan, 1987):

[B] $aR = aS \Rightarrow (a \in F(R) \text{ iff } a \in F(S)).$

We now define: an expression X , of whatever category and complexity, is a **proper anaphor** iff its interpretation in all models satisfies [B], and in some models fails [A].

Note that this definition is properly semantic: whether an expression is an anaphor depends on how it is semantically interpreted. Of course the definition would have to be generalized for larger languages than we consider here, but it suffices to illustrate how semantically defined properties and relations may or may not be coded in the syntax.

Corollary 2.6. In Little English, letting $NP_{ana} = \{\sigma \in PH \mid \sigma \text{ is a proper anaphor}\}$,

- NP_{ana} is infinite. It includes, e.g., $\langle \text{himself}, NP \rangle$, $\langle \text{both himself and John}, NP \rangle$, $\langle \text{either both himself and John or both himself and Bill}, NP \rangle$, ...
- $\langle \text{himself}, NP \rangle$ is the only lexical item in NP_{ana} .
- NP_{ana} is a proper subset of PH_{NP} , and does not include, e.g., $\langle \text{him}, NP \rangle$, $\langle \text{both him and John}, NP \rangle$, ...
- Not all NPs containing *himself* lie in NP_{ana} . For example, the following are not in the set: $\langle \text{both himself and neither himself nor bill}, NP \rangle$, $\langle \text{both either himself or john and neither bill nor himself}, NP \rangle$. In fact, it appears that a purely syntactic definition of NP_{ana} would be quite complex. Nevertheless:

Theorem 2.7. In Little English, NP_{ana} is K -invariant for $K = \{\langle \textit{himself}, NP \rangle\}$.

Theorem 2.7 says that the set of proper anaphors of Little English is structurally identifiable once its lexical elements are given. So we can identify anaphors just by knowing their structure and where *himself* occurs. Note also that NP_{ana} is not an invariant. In this language, a hom may interchange $\langle him, NP \rangle$ and $\langle himself, NP \rangle$, so NP_{ana} is not closed under isomorphism. Notice that in real English, in which agreement is enforced between anaphors and antecedents, no hom could interchange *him* and *himself*, as shown by the sentences (3a,b):

- (3a) John and Bill like him (b) * John and Bill like himself

Consequently, we expect that in real English NP_{ana} is fully invariant, that is, K -invariant for $K = \emptyset$.

We might emphasize here that our notion of K -invariance has enabled us to say in a rigorous way what we mean when we say informally that the anaphors of Little English are structurally identifiable from their lexical elements. Note that the functions which build NP_{ana} from $\{\langle \textit{himself}, NP \rangle\}$ are not generating functions of L .

Knowing now what the proper anaphors of Little English are, we may observe that they only occur only as arguments of $P2$ s to form $P1$ s. So we can identify the possible locations of anaphors in an expression in familiar structural terms: for expressions β, δ in Little English β is a possible locus of an anaphor in δ , β PLA δ , iff for some $P2, \eta$, $F2(\beta, \eta)$ occurs in δ . Of course this definition does not require that β itself be an anaphor. Rather, the idea is that those occurrences of β which combined with $P2$ s to form $P1$ s can be replaced by anaphors preserving grammaticality (i.e., membership in PH). For example, omitting category designations for readability, *john* is a possible locus of an anaphor in *bill criticized john* since $F2(\langle \textit{john}, NP \rangle, \langle \textit{criticized}, P2 \rangle) = \langle \textit{criticized john}, P1 \rangle$ occurs in $\langle \textit{bill criticized john}, S \rangle$. We note:

Theorem 2.8. In Little English, PLA is a syntactic invariant.

In the same vein we observe that σ is a structurally possible antecedent for τ in δ iff for some ν , $F1(\sigma, F2(\tau, \nu))$ occurs in δ . And again we have:

Theorem 2.9. $\{\langle \sigma, \tau, \eta \rangle \mid \sigma \text{ is a structurally possible antecedent for } \tau \text{ in } \eta\}$ is a syntactic invariant in Little English.

Thus, as in current versions of Binding Theory, Little English grammatically identifies the Anaphor-Antecedent relation in terms of constituent structure.

2.2. Little Latin. We turn now to a second language, Little Latin, in which anaphors are identified in terms of overt case marking, not constituency. Anaphora paradigms which motivate our syntax here are found in, e.g., Hindi, Armenian, and Korean (Keenan, 1988). In Hindi (4a,b), we can flip the order of the subject NP and the object reflexive easily, but we cannot change the case marking (5a,b).

- | | |
|---|--|
| (4a) Ram-ne aapne-ko mara
Ram-Subj self-obj hit
'Ram hit himself' | (b) Aapne-ko Ram-ne mara
self-obj Ram-subj hit |
| (5a) *Ram-ko aapne-ne mara
Ram-obj self-subj hit | (b) *Aapne-ne Ram-ko mara
self-subj Ram-obj hit |

Little Latin is here defined to be like Little English in that it has basically the same "content" words with the same meanings, it builds intransitive and transitive S s from one and two place predicates combined with the appropriate number of arguments, and complex expressions in essentially all categories can be built using *and*, *or* and *nor*. In addition, it has just one lexical anaphor. But in contrast to Little English, Little Latin is verb-final, the relative order of arguments of $P2$ s is free, and grammatical case is overt. NPs combine with case markers to form KPs and it is the latter which function as arguments of predicates. An "accusative KP ," KPa , combines with a $P2$ to form a $P1n$, i.e. a $P1$ which combines with

a nominative *KP*, a *KP_n*, to form an *S*. Equally, nominative *KPs* may combine directly with *P2s* to form *P1a's*, the sort of *P1* which combines with an accusative *KP* to make an *S*. So the category system and generating functions are slightly more complex than Little English. On the other hand, no CASE filter is needed. Formally:

Little Latin = $\langle CAT, V, LEX, \mathcal{F}, \Phi \rangle$ where,

$$CAT = \{NP, K, KP_a, KP_n, P2, P1_a, P1_n, S, CONJ\}$$

$$V = \{john, bill, pron, self, laughed, cried, praised, criticized, \\ -nom, -acc, both, and, either, or, neither, nor\}$$

$$LEX = (\{john, bill, pron, self\} \times \{NP\}) \cup \\ (\{laughed, cried\} \times \{P1_n\}) \cup \\ (\{praised, criticized\} \times \{P2\}) \cup \\ (\{-nom, -acc\} \times \{K\}) \cup \\ (\{and, or, nor\} \times \{CONJ\})$$

$$\mathcal{F} = \{CASEMARK, F1, F2, AND, OR, NOR\}$$

$$\Phi = \emptyset$$

where,

$$CASEMARK(\langle d, K \rangle, \langle s, NP \rangle) = \begin{cases} \langle sd, KP_n \rangle & \text{if } d = -nom \text{ and } s \neq self \\ \langle sd, KP_a \rangle & \text{if } d = -acc \end{cases}$$

$$F1(\langle s, C \rangle, \langle t, C' \rangle) = \langle s \ t, S \rangle \quad \text{if } C = KP_a, C' = P1_a, \text{ or} \\ C = KP_n, C' = P1_n$$

$$F2(\langle s, C \rangle, \langle t, P2 \rangle) = \begin{cases} \langle s \ t, P1_n \rangle & \text{if } C = KP_a \\ \langle s \ t, P1_a \rangle & \text{if } C = KP_n \end{cases}$$

for $C \in \{KP_a, KP_n, P2, P1_a, P1_n, S\}$,

$$AND(\langle and, CONJ \rangle, \langle s, C \rangle, \langle s', C' \rangle) = \langle both \ s \text{ and } s', C \rangle$$

$$OR(\langle or, CONJ \rangle, \langle s, C \rangle, \langle s', C' \rangle) = \langle either \ s \text{ or } s', C \rangle$$

$$NOR(\langle nor, CONJ \rangle, \langle s, C \rangle, \langle s', C' \rangle) = \langle neither \ s \text{ nor } s', C \rangle.$$

It is useful to observe the following:

1. No generating functions yield expressions of category *NP* or *K* or *CONJ*; so in that sense these categories are closed. All you can do with an *NP* is combine it with a casemarker *K* = *-nom* or *-acc* to yield a *KP_a* or *KP_n*.
2. We cannot conjoin *KP_a*'s with *KP_n*'s, since *KP_a* and *KP_n* are different categories. So, e.g. *both john-acc and bill-nom* is not a string of category *C* for any $C \in CAT$.
3. Note that the following strings of category *S* are generated (with the bracketting indicated):

- a. *[john-nom [self-acc praised]]*
- b. *[self-acc [john-nom praised]]*

Moreover these expressions will be assigned the same semantic interpretation (see below). And since $\langle self-acc, KP_a \rangle$ will be interpreted as an anaphor we see that in Little Latin anaphors may asymmetrically c-command their antecedents.

4. By contrast, *john-acc self-nom praised* is not a string of category *S* in this language, since *self-nom* is not a string of category *KPn*. Thus Little Latin enforces case relations between anaphors and their antecedents, but not constituency relations as in Little English.

Finally we observe, as in Little English:

Theorem 2.10. In Little Latin, for each $C \in CAT$, PH_C is a syntactic invariant.

We consider now the semantics for Little Latin. Like Little English, *john*, *bill* and *pron* denote individuals, I_b 's; *self* denotes *SELF*, and, or and nor have their standard boolean meanings; $P2$ s denote binary relations over E , $P1n$'s properties, S 's truth values, and NPs denote (possibly partial) functions taking properties to truth values and binary relations to properties. Moreover, as we have only one case distinction, KPa 's denote the same as the NP they are built from, while $m(-acc)$ sends any NP denotation (there are at most four in any model) to itself.

What is semantically new here are *-nom*, KPn 's, and $P1a$'s. KPn 's, like NPs , map properties to $2 = \{True, False\}$, but unlike NPs they map binary relations to $[[R \rightarrow P] \rightarrow 2]$, the set of functions whose range is 2 and whose domain is the set of functions from binary relations to properties. So when a KPn combines with a $P2$ it is still (semantically) looking for something which maps $P2$ denotations to properties. Syntactically, of course, what a KPn forms when it combines with a $P2$ is a $P1a$, so $P1a$'s denote elements $[[R \rightarrow P] \rightarrow 2]$. Finally, $m(-nom, K)$ takes individuals (which are NP denotations other than *SELF*) as follows:

$$\begin{aligned} m(-nom, K)(I_b)(p) &= True \text{ iff } b \in p & \text{all } p \in P, \\ m(-nom, K)(I_b)(R)(G) &= I_b(G(R)) & \text{all } R \in R, \text{ all } G \in [R \rightarrow P] \end{aligned}$$

We define interpretations relative to a model as follows. For each model $\mathcal{M} = \langle E, m \rangle$, $|\cdot|^\mathcal{M}$ is that function which extends m satisfying:

$$\begin{aligned} |F(\langle s, C \rangle, \langle t, C' \rangle)|^\mathcal{M} &= \begin{cases} |(s, C)|^\mathcal{M}(|(t, C')|^\mathcal{M}) & \text{if (a) } F = CASEMARK \text{ or } F = F2, \text{ or} \\ & \text{(b) } F = F1, C = KPn, \text{ and } C' = P1n \\ |(t, C')|^\mathcal{M}(|(s, C)|^\mathcal{M}) & \text{if (c) } F = F1, C = KPa, \text{ and } C' = P1a \end{cases} \\ |G(\eta, \sigma, \tau)|^\mathcal{M} &= m(\eta)(|\sigma|^\mathcal{M}, |\tau|^\mathcal{M}) & \text{for } G \in \{AND, OR, NOR\}. \end{aligned}$$

Notice that in these last definitions we have labelled some parts (a), (b), and (c). These clauses are relevant to establishing that the following two *S*s are logically equivalent, that is, they receive the same truth value in all models:

$$\begin{aligned} &\langle self-acc john-nom praised, S \rangle, \\ &\langle john-nom self-acc praised, S \rangle. \end{aligned}$$

This is easy to prove. Let $\mathcal{M}$ be an arbitrary model. Then, omitting category markings on strings for readability, we have:

$$\begin{aligned}
& |self-acc\ john-nom\ praised|^{\mathcal{M}} \\
&= |john-nom\ praised|^{\mathcal{M}}(|self-acc|^{\mathcal{M}}) && \text{by (c)} \\
&= (|john-nom|^{\mathcal{M}}(|praised|^{\mathcal{M}}))(|self-acc|^{\mathcal{M}}) && \text{by (a)} \\
&= m(-nom)(|john-nom|^{\mathcal{M}})(|praised|^{\mathcal{M}})(|self-acc|^{\mathcal{M}}) && \text{by (a)} \\
&= |john-nom|^{\mathcal{M}}(|self-acc|^{\mathcal{M}}(|praised|^{\mathcal{M}})) && \text{def } m(-nom) \\
&= |john-nom|^{\mathcal{M}}(|self-acc\ praised|^{\mathcal{M}}) && \text{by (a)} \\
&= |john-nom\ self-acc\ praised|^{\mathcal{M}} && \text{by (a)}
\end{aligned}$$

Clearly then, anaphors and their antecedents in Little Latin do not satisfy the sort of “hierarchical” relations (e.g., c-command) that their counterparts do in Little English. Rather, anaphors are identified in terms of their case marking properties, and the anaphor-antecedent relation is identified in terms of the case relation of the two *KPs* in a simple sentence:

Theorem 2.11. The set KP_{ana} of *KPa*’s interpreted as proper anaphors is similar to NP_{ana} in Little English in that,

- a. KP_{ana} is infinite,
- b. the only non-coordinate member of KP_{ana} is $\langle self-acc, KP_a \rangle$, and
- c. KP_{ana} is a proper subset of PH_{KP_a} , lacking, e.g., $\langle pron-acc, KP_a \rangle$, $\langle both\ john-acc\ and\ bill-acc, KP_a \rangle$,...
- d. Not all *KPa*’s containing *self-acc* lie in KP_{ana} . E.g., $\langle both\ self-acc\ and\ neither\ self-acc\ nor\ bill-acc, NP \rangle \notin KP_{ana}$,...

Theorem 2.12. In Little Latin, KP_{ana} is invariant, that is, *K*-invariant for $K = \emptyset$.

We observe that an expression β occurring in an expression δ can be replaced by an anaphor preserving grammaticality iff β is a phrase of category *KPa*. More explicitly: $\beta \text{ PLA } \delta$ iff β occurs in δ and $\beta \in PH_{KP_a}$.

Theorem 2.13. In Little Latin, $\{(\beta, \delta) \mid \beta \text{ is a possible location of an anaphor in } \delta\}$ is an invariant.

Equally in Little Latin we observe that an expression σ is a **structurally possible antecedent** for τ in δ iff $\sigma \in PH_{KP_n}$ and for some η , either $F1(\sigma, F2(\tau, \eta))$ occurs in δ or $F1(\tau, (F2(\sigma, \eta)))$ occurs in δ . And we show:

Theorem 2.14. $\{(\sigma, \tau, \eta) \mid \sigma \text{ is a structurally possible antecedent of } \tau \text{ in } \eta\}$ is an invariant in Little Latin.

To conclude this section, we have exhibited two simple languages in which the possible locations of anaphors and also the structural anaphor-antecedent relation are syntactic invariants. This is so even though the structural means the two languages use to identify anaphors and antecedents are quite different. In one language anaphors are always lower in the derivation tree than their antecedents (and hence c-commanded by them). In the other, anaphors may be higher than their antecedents but antecedents must be marked nominative

whereas anaphors must be marked accusative. We see then a natural sense in which the identification of anaphors and antecedents may be a structural invariant of languages without being structurally realized in the same way in all languages. We propose as a universal of natural language:

Universal: In all natural languages L ,

- a. The set of anaphorically interpreted expressions is lexically identifiable, and
- b. The relations "is a possible location of an anaphor in" and "is a structurally possible antecedent of" are syntactic invariants.

We turn now to a more formal study of the sorts of grammars we have been using.

3. Formal properties of language systems and some open questions. Our primary concern in this paper has been with the invariants of a language system. Our first set of theorems below provide some information regarding how to determine such invariants. Ultimately of course to determine the invariants for a particular language system L we determine the set of structure preserving functions HOM_L , and so a number of our results concern these functions. (We omit the subscript L when no confusion will result.)

3.1. Invariants. Recall first the following definitions:

Definition:

- a. For $\sigma, \tau \in PH$, $\sigma \cong \tau$ iff $\exists h, k \in HOM$, $h(\sigma) = \tau$ and $k(\tau) = \sigma$.
When $\sigma, \tau \in PH^n$, we write $\sigma \cong \tau$ to mean there are homs h, k such that for all $1 \leq i \leq n$, $h(\sigma_i) = \tau_i$ and $k(\tau_i) = \sigma_i$.
- b. For $K \subseteq PH$, $\sigma \cong_K \tau$ iff there are homs h, h' which preserve K such that $h(\sigma) = \tau$ and $h'(\tau) = \sigma$. We let $HOM^K \subseteq HOM$ be the set of homs which preserve K ; the homs which preserve K will be called a K -homs.
- c. For $R \subseteq PH^n$ and $K \subseteq PH$, we say that R is K -invariant iff for all n -tuples σ, τ , $\sigma \cong_K \tau \Rightarrow (\sigma \in R \text{ iff } \tau \in R)$. Equivalently: R is K -invariant iff R is closed under K -isomorphisms, $\sigma \in R \Rightarrow \tau \in R$ for all τ such that $\sigma \cong_K \tau$. R is called invariant (simpliciter) when R is K -invariant for $K = \emptyset$.

We now observe some elementary properties concerning invariants.

Theorem 3.1. $\cong_K$ is an equivalence relation.

Theorem 3.2.

- a. For $K, D \subseteq PH$, if D is preserved by all K -homs then D is K -invariant. The converse may fail, and in particular may fail when $K = \emptyset$.
- b. The collection of subsets of PH^n preserved by all K -homs is closed under arbitrary intersections and unions, and includes PH^n and $\emptyset$ as members. So it is a complete lattice. But it is not in general closed under complements.
- c. If $\{A_i \mid i \in I\}$ is an indexed family of sets each of which is closed under all K -homs then their product $\prod_{i \in I} A_i$ is closed under all K -homs. And if each A_i is K -invariant then $\prod_{i \in I} A_i$ is K -invariant.
- d. If A is closed under all K -homs and $\mathcal{G} \subseteq \mathcal{F}$ then the closure of A with respect to $\mathcal{G}$, $CL[A, \mathcal{G}]$, is closed under all K -homs. We may everywhere replace 'closure under all K -homs' here by ' K -invariant' if all $g \in \mathcal{G}$ are one to one.

Part b of the previous theorem contrasts the sets preserved by all K -homs with our earlier result, Theorem 2.5, about the sets of $\cong_K$ elements. We repeat Theorem 2.5 here for convenience, elaborating just slightly:

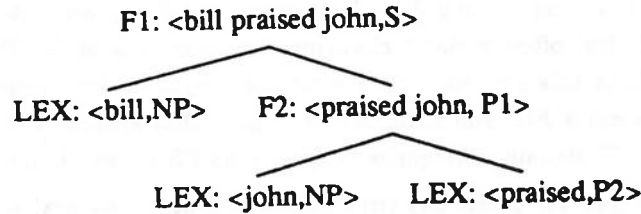
Theorem 3.3. For all language systems L , all n , and all $K \subseteq PH^n$, PH^n and $\emptyset$ are K -invariants and INV_K^n is closed with respect to arbitrary intersections, unions, and complements (relative to PH^n). So each INV_K^n is a complete, atomic boolean algebra. The atoms of each INV_K^n are the $[\sigma]_K = \{\tau \mid \sigma \cong_K \tau\}$ for each $\sigma \in PH^n$.

Theorem 3.4. If $K \subseteq PH$ then $HOM^K \subseteq HOM^\emptyset$ and $INV_\emptyset^n \subseteq INV_K^n$.

Theorem 3.5. (Universal Invariants)

- a. For all language systems L , the following sets are preserved by all homs and are thus invariant (Thm 3.2a), in fact K -invariant for all K (Thm 3.4):
 1. $Dom(F)$, all $F \in \mathcal{F}$
 2. $Ran(F)$, all $F \in \mathcal{F}$
 3. $\emptyset, LEX, CL[LEX, \mathcal{F}], PH$
 4. $\{(\beta, \delta) \mid \beta, \delta \in PH \text{ and } \beta \text{ occurs in } \delta\}$
- b. For each n -tuple σ , $[\sigma]_K = \{\tau \mid \tau \cong_K \sigma\}$ is an (atomic) K -invariant of arity n . All other elements of INV_K^n are unions of these.

3.2. The structure preserving maps. Given an arbitrary language system $L = \langle CAT, V, LEX, \mathcal{F}, \Phi \rangle$, we consider first a reassuring observation about the structure maps (the "homs"). They are precisely the functions from $V^* \times CAT$ to $V^* \times CAT$ which preserve derivation trees for expressions. E.g., the following depicts a derivation tree for $\langle \text{bill praised john}, S \rangle$ in Little English:



In general, a derivation tree T for L is a rooted, finite, ordered, labelled tree satisfying:

- 1) the label of each leaf is a pair $\langle LEX, \sigma \rangle$, where $\sigma \in LEX$,
- 2) labels of internal nodes n are pairs $\langle F, \sigma \rangle$ such that $F \in \mathcal{F}$ and, where $n_1, \dots, n_k$ is the (maximal) sequence of nodes that n immediately dominates, $\langle label(n_1)_2, \dots, label(n_k)_2 \rangle \in Dom(F)$ and $\sigma = F(label(n_1)_2, \dots, label(n_k)_2)$, and
- 3) the root node n is such that $label(n)_2$ passes all the filters.

Theorem 3.6. Let $h: (V^* \times CAT) \mapsto (V^* \times CAT)$. Write t_h for that function from derivation trees to trees which replaces the second coordinate of labels by their values under h .

Then $h \in HOM_L$ iff for all derivation trees T of L , $t_h(T)$ is a derivation tree of L .

Thus the homs are exactly the functions that preserve all of the derivations of expressions in L .

There is a close, inverse relation between the homs on the one hand and the generating functions and filters, on the other. The fewer the functions and filters, the easier it is to preserve them so the more homs we have. This inverse relation is given by Theorem

3.8 below. The limit cases are in Theorem 3.9. First we note the following useful and algebraically expected fact:

Theorem 3.7.

- a. HOM_L is closed under composition, and
- b. $id_{V^* \times CAT} \in HOM_L$, where $id_{V^* \times CAT}$ is the identity map from $V^* \times CAT$ to $V^* \times CAT$.

Theorem 3.8. Let $L = \langle CAT, V, LEX, \mathcal{F}, \Phi \rangle$ and $L' = \langle CAT, V, LEX, \mathcal{F}', \Phi' \rangle$.

- a. If $\mathcal{F} \subseteq \mathcal{F}'$ and $\Phi = \Phi'$, then $PH_L \subseteq PH_{L'}$ and $HOM_{L'} \subseteq HOM_L$.
- b. If $\mathcal{F} = \mathcal{F}'$ and $\Phi \subseteq \Phi'$, then $PH_{L'} \subseteq PH_L$ and $HOM_L \subseteq HOM_{L'}$.

Only the identity map on $V^* \times CAT$ preserves structure if all functions of the right type are generating functions or all functions of the right type are filters:

Theorem 3.9. Let $L = \langle CAT, V, LEX, \mathcal{F}, \Phi \rangle$, $LEX \neq \emptyset$.

- a. If $\Phi = [(V^* \times CAT) \mapsto \{0, 1\}]$, then $PH_L = \emptyset$ and $HOM_L = \{id_{V^* \times CAT}\}$.
- b. If $\mathcal{F} = \bigcup_{n \geq 0} [(V^* \times CAT)^n \rightarrow (V^* \times CAT)]$, and $\Phi = \emptyset$, then $PH_L = (V^* \times CAT)$ and $HOM_L = \{id_{V^* \times CAT}\}$.

These last two theorems underscore both similarities and differences between filters and generating functions. They are similar in that an augmentation of either decreases the homs, since in each case there is more for a hom to preserve. But the decrease they induce is not of the same type: an increase in $\mathcal{F}$ holding Φ constant increases PH_L , whereas an increase in Φ holding $\mathcal{F}$ constant decreases PH_L .

This inverse relation between homs and functions/filters is highly non-linear. Below we exhibit many ways of augmenting $\mathcal{F}$ without changing HOM_L and often without changing PH_L either, and thus often without changing the invariants of L . We use the following notation throughout this section: For L a language system with generating functions $\mathcal{F}$, categorized expressions PH , and homs HOM , $L_{\mathcal{F}'}$ is that system which results from L by replacing $\mathcal{F}$ with $\mathcal{F}'$, defining categorized expressions $PH_{\mathcal{F}'}$, and homs $HOM_{\mathcal{F}'}$.

Our first example is linguistically trivial but algebraically natural and technically useful. Namely, we can add the identity function on $V^* \times CAT$ to $\mathcal{F}$ without changing the homs or PH . More generally, let us write $pr_{j/n}$ for the projection function with domain $(V^* \times CAT)^n$ which sends each n -tuple in its domain to its j th coordinate. Write $CL[\mathcal{F}, PR]$ for $\mathcal{F} \cup \{pr_{j/n} \mid 1 \leq j \leq n\}$. Then,

Theorem 3.10. $PH = PH_{CL[\mathcal{F}, PR]}$ and $HOM = HOM_{CL[\mathcal{F}, PR]}$

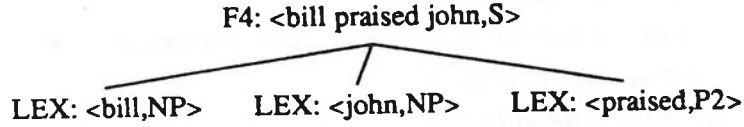
A linguistically more interesting closure condition is composition. Adding to $\mathcal{F}$ a function that is the composition of others in $\mathcal{F}$ does not change HOM . In fact, the (typically infinite) closure of $\mathcal{F}$ under the following very general form of composition preserves HOM :

Definition: Given $\langle CAT, V, LEX, \mathcal{F}, \Phi \rangle$, $\mathcal{F}$ is closed under **generalized composition** (GC) iff for all $m, k \geq 1$, if g of arity m is in $\mathcal{F}$ and $h_1, \dots, h_m$ with common domain $K \subseteq (V^* \times CAT)^k$ satisfying $\langle h_1(\sigma), \dots, h_m(\sigma) \rangle \in Dom(g)$ for all $\sigma \in K$ are in $\mathcal{F}$, then $f: K \rightarrow (V^* \times CAT)$ is in $\mathcal{F}$, where for all $\sigma \in K$, $f(\sigma) = g(h_1(\sigma), \dots, h_m(\sigma))$.

Write $CL[\mathcal{F}, GC]$ for the closure of $\mathcal{F}$ under generalized composition.

Theorem 3.11. $HOM = HOM_{CL[\mathcal{F}, GC]}$ and $PH = PH_{CL[\mathcal{F}, GC]}$.

We observe that closing a system under generalized composition may modify its strong generative capacity by adding many more derivation trees. For example, recalling the grammar of Little English, one way we may compose $F1$, which builds S s from NP s and $P1$ s, with $F2$, which builds $P1$ s from NP s and $P2$ s, yields a function $F4$ which directly maps triples consisting of two NP s and a $P2$ to a sentence, as indicated in the following tree:



Using $F4$, the string *Bill praised John* now has a flat derivation tree that it did not have in the original system.³

As McKenzie et al. (1987) remark in a more general algebraic setting, in studying a set $\mathcal{F}$ of functions it is often more revealing to work with its clone, its closure under projections and generalized composition, than with $\mathcal{F}$ itself. This is reflected in our linguistic setting by:

Corollary 3.12. For L and L' are language systems differing at most in their sets $\mathcal{F}$ and $\mathcal{F}'$ of generating functions, if $\text{clone}(\mathcal{F}) = \text{clone}(\mathcal{F}')$ then $PH_L = PH_{L'}$, $HOM_L = HOM_{L'}$, and, for all $n \in \mathbb{N}$, $K \subseteq PH_L = PH_{L'}$, they have the same INV_K^n .

For example, the set $\{ \langle \sigma, \tau, \eta \rangle \mid \sigma \text{ is a structurally possible antecedent for } \tau \text{ in } \eta \}$ which is invariant in Little English will also be invariant in any language system that has the same clone. This continues to hold when we generalize the notion "clone" to include any of the hom-preserving closure operations.

On the other hand, one way our language systems differ from certain algebraic systems is that we cannot close them under the addition of constant functions without radically changing the homs:

Theorem 3.13. For any $n \geq 1$, $h \in HOM_L$ preserves all constant functions of arity n iff $h = id_{V^* \times CAT}$.

We also cannot in general close our generating functions under λ -abstraction, as defined below, without changing the homs:

Definition: For each $F \in \mathcal{F}$ with arity $n \geq 1$, for each $1 \leq i \leq n$ and for each $(n-1)$ -tuple $\sigma = \langle \sigma_1, \dots, \sigma_{n-1} \rangle$ such that for some τ , $\langle \sigma_1, \dots, \sigma_{i-1}, \tau, \sigma_i, \dots, \sigma_{n-1} \rangle \in \text{Dom}(F)$, we define $\lambda(i, F, \sigma): (V^* \times CAT) \rightarrow (V^* \times CAT)$ as follows:

$$\begin{aligned}
 \text{Dom}(\lambda(i, F, \sigma)) &= \{ \tau \mid \langle \sigma_1, \dots, \sigma_{i-1}, \tau, \sigma_i, \dots, \sigma_{n-1} \rangle \in \text{Dom}(F) \}, \text{ and} \\
 \lambda(i, F, \sigma)(\tau) &= F(\sigma_1, \dots, \sigma_{i-1}, \tau, \sigma_i, \dots, \sigma_{n-1}).
 \end{aligned}$$

Theorem 3.14. If we close $\mathcal{F}$ under λ -abstraction in the sense defined above, we do not in general preserve HOM_L .

The interest of this fact is highlighted by the observation that for all $\sigma \in PH_L$, there is an n -ary $g \in \text{clone}(\mathcal{F})$ and some $\tau_1, \dots, \tau_n \in LEX$ such that $\sigma = g(\tau_1, \dots, \tau_n)$.

There are still other ways to extend $\mathcal{F}$ without changing HOM_L . If we add a function that identifies two arguments of an already present generating function, this does not change

HOM_L :

Definition: $\mathcal{F}$ is closed under identification of arguments (ID) iff for all i, j, n such that $1 \leq i < j \leq n$, for all n -ary functions F , if $F \in \mathcal{F}$ then $(n-1)$ -ary $F_{i,j} \in \mathcal{F}$ where $F_{i,j}(\sigma_1, \dots, \sigma_{n-1}) = F(\sigma_1, \dots, \sigma_{j-1}, \sigma_i, \sigma_j, \dots, \sigma_{n-1})$ if $\langle \sigma_1, \dots, \sigma_{j-1}, \sigma_i, \sigma_j, \dots, \sigma_{n-1} \rangle \in Dom(F)$, and $F_{i,j}(\sigma_1, \dots, \sigma_{n-1})$ is undefined otherwise.

Theorem 3.15. $HOM = HOM_{[\mathcal{F}, ID]}$ and $PH = PH_{[\mathcal{F}, ID]}$.

Adding the union or "unification" of compatible functions does not change HOM either:

Definition: $F, G \in \mathcal{F}$ are compatible iff

- (i) F and G are of the same arity, and
- (ii) for all $\sigma \in Dom(F) \cap Dom(G)$, $F(\sigma) = G(\sigma)$.

Definition: For compatible F, G , the unification

$$(F \vee G)(\sigma) \begin{cases} = F(\sigma) & \text{if } \sigma \in Dom(F) \\ = G(\sigma) & \text{if } \sigma \in Dom(G) \\ \text{undefined otherwise} \end{cases}$$

Theorem 3.16. $HOM = HOM_{[\mathcal{F}, \vee]}$ and $PH = PH_{[\mathcal{F}, \vee]}$.

Finally, of more algebraic interest, call a function $g: (V^* \times CAT) \mapsto (V^* \times CAT)$ central if it is one to one and commutes with all $h \in HOM_L$. For each $F \in \mathcal{F}$, such a g naturally induces a function g_F whose domain is $\{g(\sigma) \mid \sigma \in Dom(F)\}$ and whose action at any $\tau \in Dom(g_F)$ is given by $g_F(\tau) = g(F(g^{-1}(\tau)))$. Let $*$ be the binary operation mapping each pair $\langle g, F \rangle$ to g_F . Then:

Theorem 3.17. $HOM = HOM_{CL[\mathcal{F}, *]}$.

(We do not know if $PH = PH_{CL[\mathcal{F}, *]}$.) For bijective g we have a stronger result: any $F \in \mathcal{F}$ can be replaced by g_F without changing the homs. This follows since if g is bijective, one shows easily that g^{-1} is central (and bijective) and that $*(g^{-1}, g_F) = F$. So closing $\mathcal{F} - \{F\} \cup \{g_F\}$ under $*$ puts F back and does not change HOM_L . The idea here is that, for g bijective, we could have chosen g_F instead of F as a generating function without altering the structure of L .

3.3. Factors and categories. We turn now to a somewhat deeper question concerning homs for natural languages. Namely, how much of the language do we have to see to determine whether two expressions are isomorphic or whether one can be mapped to the other by a hom? Naively speaking it would be surprising if natural languages presented homs that could only be distinguished from non-homs on expressions consisting of more than a thousand lexical items. We expect then the behavior of homs in natural languages should be learnable just by considering a "small" sample of "small" expressions. Language systems that satisfy this intuition in a particularly strong way are ones in which:

- a. LEX is finite and
- b. homs are "projected" from their behavior on lexical items. That is, each hom extends a function from LEX to LEX .

We show here these conditions are met by a broad class of (formal) languages which include those we have developed to model certain aspects of natural languages. These are languages

whose generating functions “factor” and are “shuffling” – notions we define below. For language systems that meet these conditions Theorems 3.18 and 3.19 give us significant information concerning how a hom is determined once its behavior on the lexical items is given.

Many natural language systems have generating functions whose action on the strings is independent of their action on the categories. We make this idea precise as follows:

Definition: A function $f \in [(V^* \times CAT)^n \rightarrow (V^* \times CAT)]$ has **factors** iff there are functions $f_1 \in [(V^*)^n \rightarrow V^*]$ and $f_2 \in [CAT^n \rightarrow CAT]$ such that for all $\langle \langle s_1, C_1 \rangle, \dots, \langle s_n, C_n \rangle \rangle \in Dom(f)$,

$$f(\langle \langle s_1, C_1 \rangle, \dots, \langle s_n, C_n \rangle \rangle) = \langle f_1(s_1, \dots, s_n), f_2(C_1, \dots, C_n) \rangle.$$

In this case, we call f_1 the **first factor** of f , and f_2 the **second factor** of f . A set $\mathcal{F}$ has **factors** iff every $F \in \mathcal{F}$ does, in which case we let $\mathcal{F}_1 = \{F_1 \mid F \in \mathcal{F}\}$ and $\mathcal{F}_2 = \{F_2 \mid F \in \mathcal{F}\}$. And $L = \langle CAT, V, LEX, \mathcal{F}, \Phi \rangle$ has **factors** iff $\mathcal{F}$ does.

One role that categories play in the definition of languages is to delimit the domains of the generating functions and filters. One simple way that this could be done deserves to be singled out:

Definition: A function $f \in [(V^* \times CAT)^n \rightarrow (V^* \times CAT)]$ is **category closed** just in case $\langle \langle s_1, C_1 \rangle, \dots, \langle s_n, C_n \rangle \rangle \in Dom(f)$ implies that for all s'_i such that $\langle s'_i, C_i \rangle \in PH$, $\langle \langle s'_1, C_1 \rangle, \dots, \langle s'_n, C_n \rangle \rangle \in Dom(f)$.

The set of functions $\mathcal{F}$ is **category closed** iff every $F \in \mathcal{F}$ is.

A filter ϕ is **category closed** just in case $\phi(s, C) = \phi(s', C')$ whenever $C = C'$.

The set Φ of filters is **category closed** iff every $\phi \in \Phi$ is.

And $L = \langle CAT, V, LEX, \mathcal{F}, \Phi \rangle$ is **category closed** iff $\mathcal{F}$ and Φ are.

The natural grammars for Sentential Logic and First Order Languages are category closed, and all context free languages have grammars meeting this condition. But neither Little English nor Little Latin is category closed.

3.4. Shuffling languages. We consider here language systems in which the generating functions can permute, delete or copy their arguments, interleaving them with categoryless strings (elements of $(V - LEX_1)^*$). Little English and Little Latin both fall into this class of “shuffling languages.”

Definition: $g: (V^*)^n \times \mathbb{N} \rightarrow V^*$ is a **selection function** iff for every $1 \leq i$ there is a $1 \leq j \leq n$ such that for every $\langle \langle s_1, \dots, s_n \rangle, i \rangle \in Dom(g)$, $g(\langle \langle s_1, \dots, s_n \rangle, i \rangle) = s_j$.

Definition: Given $V, LEX_1 \subseteq V$, $f: (V^*)^n \rightarrow V^*$ is a **shuffling function** iff there is a selection function g and some $\alpha_0, \dots, \alpha_m \in (V - LEX_1)^*$ such that for every $s_1, \dots, s_n \in Dom(f)$, $f(s_1, \dots, s_n) = \alpha_0 g(s_1, \dots, s_n, 1) \alpha_1 \dots \alpha_{m-1} g(s_1, \dots, s_n, m) \alpha_m$.

A generating function $F: (V^* \times CAT)^n \rightarrow (V^* \times CAT)$ is **shuffling** iff it has factors and F_1 is shuffling. A language system with generating functions $\mathcal{F}$ is **shuffling** iff every member of $\mathcal{F}$ is.

Notice that shuffling functions can drop or repeat elements. We allow, for example, $m = 0$, or $m = 1000n$.

We now provide a partial characterization of the structure preserving functions for shuffling languages. They are all functions with factors whose action on strings is determined by their action on lexical items in a particularly simple way.

Definition: Given any function $f:V \mapsto V^*$, the **string homomorphism** determined by f is the function $f^*:V^* \mapsto V^*$ such that:

$$f^*(s) = \begin{cases} f(s) & \text{if } s \in V \\ \epsilon & \text{if } s = \epsilon \quad (\epsilon \text{ is the empty string}) \\ f(s_1) \dots f(s_n) & \text{if } s = s_1 \dots s_n, \text{ where } s_1, \dots, s_n \in V. \end{cases}$$

Theorem 3.18. Let $L = \langle CAT, V, LEX, \mathcal{F}, \Phi \rangle$ be shuffling and lexically unambiguous (i.e., $\langle s, C \rangle, \langle s, C' \rangle \in LEX \Rightarrow C = C'$). Then for all $h \in HOM_L$ there is $\phi:V \rightarrow V$ such that

- $\phi(s) = s$, all $s \in (V - LEX_1)$, and
- for all $\langle s, C \rangle \in PH$, $h(s, C)_1 = \phi^*(s)$.

Notice that the unambiguity condition does not rule out massive structural homonymy. It only eliminates the case where a given lexical item has two categories. (This unambiguity condition can be generalized slightly, but then requires a more complex formulation.)

When we consider shuffling language systems that are category closed as well, every category-preserving permutation of lexical items extends to a structure preserving function. We say a function h is **category preserving** iff $h(\sigma)_2 = \sigma_2$, for all $\sigma \in PH$.

Theorem 3.19. If L is lexically unambiguous, shuffling, and category closed then every category preserving map $h:LEX \mapsto LEX$ extends to an element of HOM_L .

3.5. Comparing languages. We conclude with an observation concerning the structural similarity of different languages. We have been using "homs" in a "local," language internal, sense. But this notion applies standardly to different languages of the same type. We say that L and L' are of the same type iff the filters of L correspond bijectively to those for L' and the set of generating functions $\mathcal{F}$ corresponds bijectively to $\mathcal{F}'$ in such a way that F and its image F' have the same arity. In such a case then a (total) function h from $(V^* \times CAT)$ to $(V'^* \times CAT')$ is a **homomorphism (hom)** from L into L' iff:

- $h|_{LEX:LEX \mapsto LEX'}$,
- for all n -tuples $\sigma \in Dom(F)$, $h(\sigma) \in Dom(F')$ and $h(F(\sigma)) = F'(h(\sigma))$, and
- for all filters ϕ , $\phi(\sigma) = \phi'(h(\sigma))$.

Definition: L and L' are **structurally equivalent** iff they are of the same type and there is a hom from L into L' and a hom from L' into L .

For example, suppose we took a language like Little English, removed say $\langle john, NP \rangle$ and added in $\langle sam, NP \rangle$ and $\langle frank, NP \rangle$. While the lexicons of the resulting languages have different sizes, and the homs of the two languages are distinct (since they have different domains), the two systems are intuitively structurally equivalent, and they are formally equivalent given the definition above. The map h from LEX to LEX' which sends $\langle john, NP \rangle$

to $\langle sam, NP \rangle$ and everything else to itself extends to a category preserving hom from L to L' . And the map k from LEX' to LEX sending both $\langle sam, NP \rangle$ and $\langle frank, NP \rangle$ to $\langle john, NP \rangle$ and everything else to itself also extends to a hom from L' to L .

Note however that Little English and Little Latin are not structurally equivalent – they are not even of the same similarity type. In this sense then these Little languages are not structurally comparable. But this does not prevent us from showing that in both the Anaphor-Antecedent relation is a structural invariant.⁴

Finally, the approach to language structure we have taken here is rather clearly in its infancy. A great many questions have arisen in the course of our work which we have not yet been able to answer. Here are a few:

1. Characterize in some enlightening way the universally invariant sets. (Cf. Theorem 3.5.)
2. In simple semantically interpreted formal languages, expressions and sets of expressions which are syntactically invariant are also semantically invariant in the sense of being closed under automorphisms of the semantic primitives (Keenan, 1987; van Benthem, 1991). Is this true of natural languages?
3. Are there revealing conditions on language systems as we have defined them which characterize known classes of languages – e.g. the context free ones, the context sensitive ones, etc.?
4. Just how much structural information do the homs give us? To what extent can we determine the degree of structural ambiguity of an L by studying its homs?
5. For an arbitrary language system L , is there a structurally equivalent system L' such that
 - a. L' has just one category?
 - b. L' has no filters?
 - c. L' is category closed?
6. Consider the following possible language universal:

Universal: All natural languages have grammars which factor.

Is this claim trivial in the sense that for any language system L there is an L' with factors such that $HOM_L = HOM_{L'}$?

Notes:

1. Technically, these functions h are "endomorphisms" (homomorphisms from the language into itself), but we prefer the term "hom" or "homomorphism," for its familiarity. Contrast the homomorphisms of §3.5.
2. More formally, we define the "occurs in" relation OCC as follows: $\beta OCC \delta$ iff for some n , $\beta OCC^n \delta$, where $\beta OCC^0 \delta$ iff $\beta = \delta$, and for all n , $\beta OCC^{n+1} \delta$ iff either $\beta OCC^n \delta$, or for some k -ary $F \in \mathcal{F}$ and some $\sigma_1, \dots, \sigma_k \in CL[LEX, \mathcal{F}]$, $\langle \sigma_1, \dots, \sigma_k \rangle \in Dom(F)$, $\delta = F(\sigma)$ and $\beta OCC^n \sigma_i$ for some $1 \leq i \leq k$.
3. Letting $\mathcal{F} = \{F1, F2, AND, OR, NOR\}$ be as defined for Little English, consider the composed function $F3: (V^* \times CAT)^3 \rightarrow (V^* \times CAT)$ defined so that $F3(\sigma, \tau, \eta) = F2(pr_{2/3}(\sigma, \tau, \eta), pr_{3/3}(\sigma, \tau, \eta))$. Since $F3$ will be in the closure of $\mathcal{F}$ under projection and composition, so will the following:

$$F4(\sigma, \tau, \eta) = F1(pr_{1/3}(\sigma, \tau, \eta), F3(\sigma, \tau, \eta)) = F1(\sigma, F2(\tau, \eta)).$$
4. We would like a stronger result here. For example, the systems which result from Little English and Little Latin by closing their generating functions under the HOM preserving operations of section 3.2 do not lie in a mutually homomorphic relation.

References:

- van Benthem, J. (1991) *Language in Action*. NY: North-Holland.
- Keenan, E.L. (1987) Semantic case theory. In J. Groenendijk and M. Stokhof, eds., *Procs. Sixth Amsterdam Colloquium*. Reprinted in R. Bartsch et al., eds., *Semantics and Contextual Expression*. Dordrecht: Foris, 1989.
- Keenan, E.L. (1988) On semantics and the binding theory. In J. Hawkins (ed), *Explaining Language Universals*. Oxford: Basil Blackwell.
- Keenan, E.L. (1991) Anaphora invariants and language universals. In D. Bates, ed., *Procs. WCCFL X*. Stanford, California: Center for the Study of Language and Information.
- Keenan, E.L. and Stabler, E.P. (1992) Language structure and invariants. Forthcoming UCLA Center for Cognitive Science Technical Report.
- McKenzie, R.N., McNulty, G.F., Taylor, W.F. (1987) *Algebras, Lattices, Varieties: Volume I*. Monterey, California: Wadsworth & Brooks/Cole.
- Montague, R. (1970) English as a formal language. Reprinted in R.H. Thomason, ed., *Formal Philosophy: Selected Papers of Richard Montague*. New Haven: Yale University Press, 1974.

Comparatives and Nested Quantification *

Jean-Yves Lerner and Manfred Pinkal
Universität des Saarlandes, Saarbrücken

1. Introduction

The paper is concerned with the relation between comparative constructions and quantifier terms. In particular, it discusses

- the status of the comparative complement as a degree quantifier
- the behavior of quantifier terms occurring in comparative complements
- attributive comparative constructions, in particular, the interaction of the comparative operator and the quantifier term in which it occurs

In our analysis, we make use of quantifying into quantifier terms, which we take to be a standard pattern of semantic composition in natural languages. This operation of "Nested Quantification" leads to an analysis that relates syntax and semantics in a rather direct and natural way: It makes quantifier scoping a much more local phenomenon than is usually assumed, and provides an analysis that assigns standard denotations to those surface constituents which form "natural semantic units".

We will restrict our discussion to clausal comparatives like (1) and (2).

- (1) John is taller than Bill is
- (2) John read a more interesting book than Bill had recommended to him

In principle, the argument carries over to phrasal comparative constructions like (3) and (4). We will, however, not comment on the special problems connected with phrasal and non-constituent comparatives like (3) and (4), in this paper.

- (3) John is taller than Bill
- (4) John recommended a more interesting book to Mary than Bill to Sue.

For the sake of easy readability we use extensional type logic as semantic representation formalism. Also, the analysis refers to the syntactic framework of GB, esp., it uses LF representations to indicate scope relations, and discusses the compatibility of the analysis with syntactic constraints formulated in the terminology of the Principles-and-Parameters approach. In its essential part, however, the argument is independent of any specific syntactic or semantic framework.

2. Some Basic Assumptions

As a prerequisite for our analysis, we make some basic assumptions about the semantic analysis of comparative constructions.

First, the arguments of the comparative relation are degrees, rather than standard objects, as is clearly shown by the interdimensional comparative in Sentence (5).

(5) The door is higher than it is wide

Example (5) also shows that we have to consider degrees on specific scales, rather than truth degrees or satisfaction degrees of the vague one-place predicates denoted by the respective adjectives' positive form: (5) will be true also in the case that the door satisfies the predicate of being high to a pretty low degree and is very wide at the same time, if only the spatial distance establishing its height exceeds its the distance establishing its width. Also, it is difficult to see how a comparative semantics working with truth degrees would handle differential comparative constructions like (6).

(6) John is three inches taller than Bill.

Furthermore, comparative constructions relate individual degrees rather than intervals, as, e.g., proposed in [Seuren 1978]. Arguments for the latter as well as the former statements are given in [Stechow 1984] and [Pinkal 1989]. In our ontology, we assume a set of degrees distinct from the set of standard individuals, and refer to it with the type symbol "d".

A semantic account of adjective constructions has to relate the semantics of comparatives to the semantic of the positive. The positive can be used in predicative or attributive position, and there are basically two predicative uses available: It can be used as a vague one-place predicate, as in (7), and as a measure expression, relating an individual and a degree, as in (8).

(7) John is (very) tall

(8) John is 6 ft. tall

From our considerations on the nature of the comparative relation it should be clear that the relevant use of the positive is the relational one. In order to cover both predicative and attributive use, we assume that the positive adjective has type $\langle\langle e, t \rangle, \langle d, \langle e, t \rangle \rangle\rangle$: It takes a one-place predicate (the common-noun denotation), and yields a relation between individuals and degrees, as first proposed in [Hellan 1981]. For predicative constructions, we assume the existence of a contextually specified parameter Q_0 that fills the predicate position. Thus, we obtain (9) and (10) as representations of the attributive* and the predicative use of *tall*, respectively.

(9) $\lambda Q \lambda d \lambda x \text{ tall}' (Q)(x, d)$

(10) $\lambda d \lambda x \text{ tall}' (Q_0)(x, d)$

The latter decision has been made for technical reasons, only. We could as well have assumed a second, type $\langle d, \langle e, t \rangle \rangle$, reading of the adjective. Note, however that one cannot do away with the more complex version (9), in general. In many cases, the adjective + noun construction cannot generally be interpreted as the semantic conjunction or intersection of both expressions; in some cases, the property of non-intersectiveness carries over to the comparative: John and Bill may both be football players and pianists, but the fact that John is a better footballplayer than Bill does not necessarily entail that John is a better pianist, as well.

3. Comparative Complements and Quantifiers

3.1 The Over-all Semantics of the Complement

As mentioned above, we restrict the discussion to clausal comparatives and take the comparative complement to denote a degree term, according to our basic considerations. The question what specific kind of degree term it is that the complement denotes has been answered in different ways.

(11) John is taller than Bill is.

Existing semantic theories of the comparative would interpret the complement clause in (11) as "the degree d such that Bill is d -tall" [Russell 1905], "the maximum degree d such that Bill is d -tall" ([v.Stechow 1984]), "every degree d such that Bill is d -tall" ([Cresswell 1976]).

The most straightforward account of the comparative complement seems to be the classical Russellian analysis as a definite description. The alternative proposals are motivated by the fact that certain operators like the connective *or* and the modal auxiliary *can* are interpreted in a non-standard way, when occurring in the comparative complement, as well as the observation that the comparative licences the occurrence of negative polarity items like *any* and *ever*. According to [Ladusow 1979], it must be assumed that the comparative establishes a downward entailing context. Sentences (12) to (15) illustrate these statements.

(12) John is taller than Bill or Fred are.

(13) John is richer than a professor can be.

(14) John is richer than Bill, Alan, or any student is.

(15) John is richer than Bill will ever be.

Sentence (12) has the preferred reading that John is taller than Bill and taller than Fred; (13) asserts that John is rich to a degree that is greater than any degree of wealth whatsoever that can be achieved by a professor. (14) and (15) are examples containing negative polarity items, which are usually taken to be narrow-scope

existentials, but receive a universal interpretation when occurring in the complement. Correct meaning assignments for all these examples can only be obtained, if the respective operators do not take scope over the degree term, and if the degree term is not a simple definite description, since this would require equal height of Bill and Fred, a constant degree of wealth for professors, etc. We rather must assume one of the alternative interpretations mentioned above. We take the universal degree term variant to be the correct one. A detailed argument for our decision is given in [Pinkal 1989]. We will just indicate the main points of the argument, here. In advance, we will point out that our choice actually leads to the right results.

We assume that the interpretation of the complement works on an underlying structure that can be approximately paraphrased by something as "than Bill is d tall" for a sentence like (11). Let the comparative complement be a universal degree term which binds the second argument of the comparison relation, and let the restriction of the universal quantifier be given by the complement clause. We obtain interpretations for (11)-(15) that can be paraphrased by (16)-(20).

- (16) John is taller than every degree d such that Bill is d tall
- (17) John is taller than every degree d such that Bill or Fred is d rich.
- (18) John is richer than every degree d such that a professor can be d rich.
- (19) John is richer than every degree d such that Bill, Alan, or some student is d rich.
- (20) John is richer than every degree d such that, for some time $t > t_0$, Bill will be d rich at t .

Note also that the restriction of a universal quantifier is downward entailing which explains the occurrence of negative polarity items.

Let us turn to the question, why we decided in favour of the universal degree term analysis, instead of v.Stechow's "maximum degree description" proposal, which leads to equivalent results for the cases discussed so far. We will give a semantic argument, and consider a plausibility argument which draws upon the syntactic structure of complement sentences.

As to the semantic argument, we restrict ourselves to one example, which however seems to provide compelling evidence.

- (21) 2 is greater than any negative rational number is

(21) is obviously true and has just that intuitive content which follows from our analysis. On v.Stechow's account, the degree term would lack a semantic value, since no maximum exists for the negative rational numbers.

The second argument runs as follows: The closest syntactic relatives to comparative complement clauses appear to be free relative clauses like (22).

- (22) I accept what(ever) you accept

Actually, it seems to be reasonable to assume for comparative complements the syntactic structure indicated in (23), which parallels the structure of free relatives given in (24), the *wh* element being a phonetically empty relative pronoun referring to degrees.

(23) John is taller than [_{CP} (*wh*)_d [_{IP} Bill is (*d*-tall)]]

(24) I accept [_{CP} *what*_x [_{IP} you accept *x*]]

The preferred reading of free relative clauses is the universal one ("everything that you accept" in example (22)). Therefore, the universal reading for the complement is not a stipulation to be made for comparatives, but is naturally obtained by a semantic generalization over a class of related syntactic constructions.

We will conclude this section by a first attempt to provide a precise description of the semantics for comparative complements. As shown in (25), we interpret the *wh* element in (23) as a universal degree determiner, form a degree predicate by abstraction with variable *d* over the open proposition denoted by the complement IP, and obtain the bottom line by functional application.

(25) $wh \Rightarrow \lambda D \lambda D' \forall d (D(d) \rightarrow D'(d))$

Bill is *d*-tall $\Rightarrow \lambda d [\text{tall}'(Q_0)(b^*, d)]$

wh_d Bill is *d*-tall $\Rightarrow \lambda D \forall d (\text{tall}'(Q_0)(b^*, d) \rightarrow D(d))$

3.2 Distributive interpretation of quantifiers in comparative complements

While the phenomena examined in the last section have been widely discussed, little attention has been paid to a problem connected with the much more common distributive interpretation of term phrases in comparative complements. The *or*-complement sentence (12), *e.g.*, has a second reading stating that John is taller than at least one of the two other guys. Apart from that special case, most types of quantifier phrases allow only a distributive reading, which cannot be obtained by a straightforward application of the analysis argued for in the last section.

(26) John is taller than every student is.

(27) John made a better proposal to Bill than he made to most students.

(28) $\lambda D \forall d (\forall x (\text{student}'(x) \rightarrow \text{tall}'(Q_0)(x, d)) \rightarrow D(d))$

Consider, *e.g.*, Sentences (26) and (27). If we apply the universal degree term analysis directly to the complement of (26), we get (28) as its denotation, which can be paraphrased as "every degree *d* such that every student is *d* tall". According to this interpretation, (26) makes a substantial claim only in the case that all

students are of identical height. Obviously, this is a much too weak and intuitively completely implausible analysis. The result for (27) is similar: We obtain "every degree d such that John made a d -good proposal to most students" as denotation of the complement, the restriction of which is non-empty only in the case that John made to most students proposals of identical quality. Note that the problem arises not only with the universal degree term analysis, but with any of the three proposals mentioned above. The resulting truth conditions are different, but all of them are pragmatically deviant for the cases under consideration. Obviously, the terms *every student* and *most student* cannot be part of the restriction of the complement's degree term, but must take scope over it, in order to obtain an adequate interpretation for (26) and (27).

This raises a syntactic problem. The next available proposition outside the complement (in syntactic terms: the next IP to which the raised quantifier can be adjoined) is that of the matrix clause. Comparative complements should however behave as scope islands, as relative clauses clearly do: The assumption of a wide-scope reading seems to violate syntactic constraints. Moreover, if we adopt the basic assumptions of GB, we have to account for ECP violations in certain cases, as, e.g., in Sentence (26). The plausible interpretation (29) seems to be available only by the assumption of the syntactically highly implausible underlying structure (30).

(29) $\forall x (\text{student}'(x) \rightarrow \forall d (\text{tall}'(Q_0)(x,d) \rightarrow \text{tall}'(Q_0)(j^*, d)))$

(30) $[_{IP} [_{NP} \text{every student}]_i [_{IP} \text{John is taller than} [_{CP} \text{wh}_d [_{IP} \text{ti is } d\text{-tall}]]]]$

3.3 Nested Quantification

A solution to this problem is provided by the technique of quantifying into noun phrases, which has been proposed in [Rooth 1985] for independent, purely semantic reasons. Rooth points out that a rule of quantifying into NP must be added to the standard quantification rules of Montague grammar in order to obtain adequate interpretations of nested NP constructions occurring in the argument position of intensional verbs, like (31).

(31) For her term project, Mary needs every book by some Norwegian.

The most natural reading for (31) is the one, where the scope of *some Norwegian* is inferior to the intensional verb *need* (which excludes application at the level of the matrix sentence), but superior to *every book* (which excludes *in-situ* application). This makes a rule of "Quantifying in NP" necessary, which Rooth formulates in roughly the following way (α' and β' are expressions of type $\langle e, t \rangle$, t and x_i and Q are variables of type e and $\langle e, t \rangle$, respectively):

- (32) If α is a NP indexed by i that translates to α' , and β is a NP that translates to β' , then the result of quantifying α in β is
 $\lambda Q \alpha' [\lambda x_i \beta'(Q)]$

We will refer to this rule as the "Rule of Nested Quantification". As can be seen from (32), the rule is a special case of Functional Composition. If we apply (32) to (33), we obtain the required representation (34).

- (33) [NP [NP some Norwegian]_i [NP every book by x_i]]

- (34) $\lambda Q \lambda P \exists x (\text{Norwegian}'(x) \wedge P(x))$
 $[\lambda x_i \lambda R \forall y (\text{book}'(y) \wedge \text{by}'(y, x_i) \rightarrow R(y))(Q)]$
 $\Rightarrow \lambda Q \exists x (\text{Norwegian}'(x) \wedge \forall y (\text{book}'(y) \wedge \text{by}'(y, x) \rightarrow Q(y)))$

The nested quantification rule provides a solution for the problem mentioned in the last Section. Quantifiers occurring within comparative complements need not be raised to the level of the matrix sentence, but can be immediately quantified into the degree term which is denoted by the comparative complement. For sentence (26), this means that (35), the denotation of *every student*, takes (36), the denotation of *wh_d x is d-tall*, as argument and yields the intuitively correct interpretation (37), which in turn leads to the adequate interpretation (29) for the matrix sentence.

- (35) $\lambda P \forall x (\text{student}'(x) \rightarrow P(x))$
(36) $\lambda D \forall d (\text{tall}'(Q_0)(x, d) \rightarrow D(d))$
(37) $\lambda D \lambda P \forall x (\text{student}'(x) \rightarrow P(x)) [\lambda x_i \lambda D' \forall d (\text{tall}'(Q_0)(x_i, d) \rightarrow D'(d))(D)]$
 $\Rightarrow \lambda D \forall x (\text{student}'(x) \rightarrow \forall d (\text{tall}'(Q_0)(x, d) \rightarrow D(d)))$

The solution given here in rather unspecific terms also works out fine on the LF account, within the framework of GB theory. The LF representation for the reading of (26) under discussion is indicated in (38).

- (38) John is taller than [CP wh_d [IP [NP every student]_i [IP t_i is d-tall]]]

The NP *every student* is adjoined to the complement IP, by QR. Its trace t_i is properly governed. The raised NP may take scope over the wh element, however, according to May's Scope Principle (cf. [May 1985]): The two occurrences of IP in the adjunction structure form segments of the same category. A category α dominates a category β only if every segment of α dominates β . Thus, the lowest maximal projection dominating the adjoined NP is not IP, but CP. The NP and the wh element are in the same c-command domain, which means that *every student* can take scope over the wh element, and therefore leads to the interpretation (37). Note that the operation of nested quantification can be iterated without difficulties

for complements containing several quantifier terms. For more details (cf. [Lerner 1991]).

The LF configuration also allows *wh* taking scope over *every student*. In this case we obtain for the complement the same representation that results from a plain *in situ* interpretation, i.e., (28). As discussed in the last section, this is a highly implausible weak reading, and it seems reasonable to explain its non-availability by a Gricean argument.

4. The Global Structure of Comparative Constructions

4.1 The Semantics of the Comparative Operator

First, let us make the assumption of Section 2 more specific that comparatives basically specify a relation between two degrees.

(39) John is taller than six feet

(40) John is tall to a degree greater than 6 ft

(41) John is taller than Bill is

(42) John is tall to a degree greater than every degree *d* such that Bill is *d*-tall

(40) appears to be a natural paraphrase of the respective comparative statement (39), in terms of the relational version of the adjective's positive form, and the same holds for (41) and (42) respectively, given the universal degree term semantics for the complement. In both cases, the semantics of the comparative operator can be roughly paraphrased by "a degree greater than whatever the complement specifies". Technically, it forms an indefinite degree description together with the complement, which binds the degree position of the positive. (For a similar proposal cf. [Heim 1985])

(43) $\lambda P \lambda D [\exists d' [P(\lambda d [d' > d]) \wedge D(d')]]$

In (43), the formal representation for the comparative operator is given, with P being a variable of type $\langle\langle d, t \rangle, t \rangle$. It looks somewhat complicated, but is a straightforward type-logical translation of the paraphrase given above. We will call the element resulting from the application of the comparative operator to the complement the "comparative term".

Using (43) as the representation for the comparative operator and $\lambda D' D'(6ft)$ as translation of the measure NP *six feet*, we obtain (44) as representation of the comparative term of (39), (45) as the interpretation of the AP *taller than six feet* (by functional composition of the comparative term and the two-place adjective denotation), and (46) as representation of (39) itself.

$$(44) \lambda P \lambda D [\exists d' [P(\lambda d [d' > d]) \wedge D(d')]] (\lambda D' D'(6ft))$$

$$\Rightarrow \lambda D [\exists d' [d' > 6ft \wedge D(d')]]$$

$$(45) \lambda x \exists d' [d' > 6ft \wedge \text{tall}'(Q_o)(x, d')]$$

$$(46) \exists d' [d' > 6ft \wedge \text{tall}'(Q_o)(j^*, d')]$$

Correspondingly, the representations of the comparative term of (41), its AP, and (41) itself work out as (48), (49), and (50), respectively, if we assume (47) to be the representation for the complement.

$$(47) \lambda D' \forall d [\text{tall}'(Q_o)(b^*, d) \rightarrow D'(d)]$$

$$(48) \lambda D \exists d' [D(d') \wedge \forall d [\text{tall}'(Q_o)(b^*, d) \rightarrow d' > d]]$$

$$(49) \lambda x \exists d' [\text{tall}'(Q_o)(x, d') \wedge \forall d [\text{tall}'(Q_o)(b^*, d) \rightarrow d' > d]]$$

$$(50) \exists d' [\text{tall}'(Q_o)(j^*, d') \wedge \forall d [\text{tall}'(Q_o)(b^*, d) \rightarrow d' > d]]$$

4.2 The Scope of the Comparative Term in Attributive Constructions

The analysis of the last section seems to carry over straightforwardly to attributive comparative constructions like (51).

(51) John wrote a better paper than Bill had written.

We might take the comparative construction to be a modifier of the head noun of the NP in which it occurs, forming an ordinary predicate denotation together with it, which acts as the restriction of the indefinite term phrase. Thus, the most plausible syntactic structure for the object NP of (51) - the "comparative NP", as we will call it in the following - seems to be the one indicated by (52).

$$(52) [NP \text{ a } [N' [N' \text{ better paper}] [\text{than Bill had written}]]]$$

We will outline the semantic analysis in somewhat more detail, now: According to our analysis of predicative constructions, we take something like "wh_d Bill had written a d good paper" as the basis for the interpretation of the complement. In the denotation of *d-good paper* in the complement as well as the lower N' in the matrix NP, the adjective denotation $\lambda Q \lambda d \lambda x \text{ good}'(Q)(x, d)$ combines with the head noun, yielding (53).

$$(53) \lambda d \lambda x \text{ good}'(\text{paper}')(x, d)$$

Using the analyses for comparative complement and comparative operator from the previous sections, we receive (54) for the complement, and (55) for the comparative term.

(54) $\lambda D' \forall d [\exists y [\text{good}'(\text{paper}')(y, d) \wedge \text{write}'(b^*, y)] \rightarrow D'(d)]$

(55) $\lambda D \exists d' [D(d') \wedge \forall d [\exists y [\text{good}'(\text{paper}')(y, d) \wedge \text{write}'(b^*, y)] \rightarrow d' > d]]$

(55) can be paraphrased by "a degree greater than every degree d such that Bill had written a d -good paper" (disregarding tense, which is irrelevant for the problem under discussion). Note that the universal degree term analysis of the complement yields the most plausible interpretation for those cases where Bill had written more than one paper.

Next, we would combine the comparative term (55) with (53), the result being (56), which provides the restriction predicate of the comparative NP.

(56) $\lambda x \exists d' [\text{good}'(\text{paper}')(x, d') \wedge \forall d [\exists y [\text{good}'(\text{paper}')(y, d) \wedge \text{write}'(b^*, y)] \rightarrow d' > d]]$

This straightforward and plausibly-looking analysis is however ruled out by the fact that the comparative imposes a constraint on the determiner position of the comparative NP, as can be seen from (57) and (58), which are clearly unacceptable.

(57) *John wrote every better paper than Bill had written.

(58) *John wrote the better paper than Bill had written.

Apparently, comparative NPs must be indefinite. This constraint cannot be accounted for, if the comparative term is given scope only over N' and accordingly becomes part of the meaning of the predicate denotation, as in (56). It follows that the comparative operator must have the NP in its scope, and that the NP adjunction structure (59) is the most plausible syntactic analysis, rather than (52). We will call the lower NP in (59) the "host NP" of the comparative construction.

(59) $[\text{NP}[\text{NP a } [N' \text{ better paper}]] [\text{than Bill had written}]]$

In a standard account of natural language quantification, it is the immediate consequence of excluding N' as a scoping possibility that the comparative operator must take scope over the matrix sentence, as a whole. This assumption is implicit, e.g., in Cresswells analysis, who takes comparative constructions to denote a relation between a degree predicate denoted by the matrix sentence ($\lambda d [\text{John wrote a } d \text{ good paper}]$, in (51)), and a degree predicate denoted by the complement ($\lambda d [\text{Bill had written a } d \text{ good paper}]$).

The nested quantification technique, however, makes a third analysis available. As the complement clause is adjoined to its host NP, we can obtain an analysis of the whole comparative NP by quantifying the comparative term into the host NP. This

account is more local than the last-mentioned alternative, and provides a semantic representation for the comparative NP, which Cresswell's account does not. Since comparative NPs like that of Sentence (51) apparently have an independent natural semantic content, the nested quantification analysis appears to be the appropriate one.

Translating the indefinite article in the usual way as existential quantifier and taking (53) to be the semantic representation of the N' of the host NP, we receive (60) as translation for the host NP itself, where d_1 is the degree variable to be bound by the comparative term.

$$(60) \lambda Q[\exists x[\text{good}'(\text{paper}')(x, d_1) \wedge Q(x)]]$$

Application of the comparative term (55) to (60) by means of the Nested Quantification Rule gives (61) as denotation for the comparative NP *a better paper than Bill had written.*, where R is a variable of type $\langle e, t \rangle$. (62) is the representation for Sentence (51).

$$(61) \lambda R [\lambda D \exists d'[D(d') \wedge \forall d[\exists y[\text{good}'(\text{paper}')(y, d) \wedge \text{write}'(b^*, y)] \rightarrow d' > d]] \\ [\lambda d_1 [\lambda Q[\exists x[\text{good}'(\text{paper}')(x, d_1) \wedge Q(x)]] (R)]]]$$

$$\Rightarrow \lambda R \exists d'[\exists x[\text{good}'(\text{paper}')(x, d') \wedge R(x)] \\ \wedge \forall d[\exists y[\text{good}'(\text{paper}')(y, d) \wedge \text{write}'(b^*, y)] \rightarrow d' > d]]$$

$$(62) \exists d'[\exists x[\text{good}'(\text{paper}')(x, d') \wedge \text{write}'(j^*, x)] \\ \wedge \forall d[\exists y[\text{good}'(\text{paper}')(y, d) \wedge \text{write}'(b^*, y)] \rightarrow d' > d]]$$

(62) is equivalent to the analysis that would have been obtained on the N' modifier account, which was discussed above, since the existential degree quantifier and the existential standard quantifier can switch their positions, without changing the truth conditions. This is different, however, if the comparative NP has a universal or a definite determiner, as in (57) and (58). If the comparative term is given scope below the determiner, (57) is assigned a representation that can be paraphrased by (63), which expresses a perfectly acceptable piece of information.

(63) John wrote every paper that is better than any paper written by Bill

The problem is that this reading is simply not available, for Sentence (57). On the alternative account, we obtain (64) as representation for *every better paper than Bill had written.*

$$(64) \lambda Q \exists d'[\forall x[\text{good}'(\text{paper}')(x, d') \rightarrow Q(x)] \\ \wedge \forall d[\exists y[\text{good}'(\text{paper}')(y, d) \wedge \text{write}'(b^*, y) \rightarrow d' > d]]]$$

(64) implies that (57) is vacuously true if there is at least one degree of goodness which is so high that no paper possesses it. Obviously, this reading is

pragmatically deviant. The crucial aspect of (64) is that the degree variable *d'* is existentially bound outside the scope of the first universal quantifier. There are similar arguments which account for the semantic oddity of (58). Thus, the analysis provides a natural explanation for the puzzling indefiniteness constraint on comparative NPs.

4.3 The Structure of Comparative NPs

The analysis presented so far provides appropriate interpretations for comparative NPs and several kinds of sentences containing comparatives. It is not very specific, however, about the question how the semantics is related to the syntactic surface structure, particularly in the case of attributive constructions. We have made use of the notions of "comparative operator", "comparative term", and "host NP", all of which lack a straightforward surface-syntactic correlate: In Example (51), the host NP is taken to be the phrase *a better paper*, but without the comparative information. The comparative term is made up from the complement (*than*) *Bill had written* (the internal structure of which has also been left open), and the comparative operator. The semantic role of the particle *than* as well as the question of the syntactic or lexical location of the comparative operator have been left completely open, so far.

Let us start with the last problem, which is the most basic one. One possible solution is to consider *than* the bearer of the comparative operator. The comparative morphem in the AP is semantically empty and has merely the syntactic function to license the complement *than* clause. The advantage of this analysis is that it leads to a very simple compositional semantics for the comparative NP. The comparative operator takes its syntactic sister, the complement clause, as argument, and the resulting comparative term can be applied *in situ* to the host NP, since it has already the right scope position, according to our analysis. The problem with *than* as comparative operator is the plain fact that comparatives can occur without an explicit complement, as in (65): We only refer to the anaphoric reading of complementless comparatives, here, where a comparison degree is provided by the context of utterance, and exclude the so-called "absolute comparative" from our discussion.

(65) John wrote a better paper

One possible way out is the assumption that the comparative morphem comes in two versions: a semantically empty one which requires an explicit complement, and a second one for complementless constructions, where it bears the comparative operator. The only alternative to this slightly artificial stipulation is to assume that the comparative morphem is the bearer of the comparative operator, in general, whereas the particle *than* is semantically empty (say, an identity operator). This explains why complementless constructions are possible, at all. At the same time, however, the problem arises how to bring comparative operator and complement together, and how to get the operator into its proper scope position. We try to give the best of an answer we can to these questions.

The comparative operator is a quantifier term lacking an argument (*i.e.*, the degree term provided by the complement), according to our semantic analysis. We assume that it gets into its scope position by a mechanism of raising or delayed application, as it is well-established for ordinary quantifier terms, in different versions. The question why it must be raised, in the case of an explicit complement, and why it must be raised to some certain level, can be answered in different ways. One might take the *than* clause to be a scope indicator, as, *e.g.*, certain negation particles in Romance languages are. One also may assume that raising is necessary because the operator is lacking an argument, and that the comparative term is applied as soon as it is saturated, *i.e.*, on the NP level. We do not claim that one of these answers is the correct and satisfactory one. However, we would like to point out that it is problems of a basic and substantial kind that we try to explain. They do not arise because of any specific theoretical decision, but must be coped with by any syntactic and semantic theory of the comparative.

There is clear evidence that there can be no uniform scope assignment is possible for the comparative operator in attributive constructions. It is the comparative operator taking scope over the determiner of the comparative phrase that accounts for the indefiniteness constraint, and the question which scope it takes depends on the presence as well as on the position of the complement. Examples (66) and (67), which are the complementless counterparts of (57) and (58), show that the scope situation is different, when no overt complement is present. (66) and (67) are perfectly acceptable on the anaphoric reading, in an appropriate context of utterance.

(66) John knows every better paper.

(67) John knows the better paper.

Whichever theory one decides to adopt, it has to account for the fact that the restriction on the determiner completely disappears, when no complement is present. This however can only be the case, when the comparative operator takes scope below the NP, *i.e.*, if it applies *in situ*, at the N' level. Our tentative explanations are that no scope indicator is present that forces raising, or, alternatively, that saturation takes place already at the surface position: There is no need for the operator to pick up its argument somewhere else in the syntax tree, an anaphoric mechanism provides a contextually specified degree variable, d_0 , or $\lambda D D(d_0)$, in the type-raised version. Anyway, the completed term has the form given in (68), it is applied *in situ*, yielding (69) as the denotation for the N' *better paper*, and (70) gives the natural denotation for the comparative NP, a paraphrase of which is "every paper that is better than d_0 ".

(68) $\lambda D [\exists d' [d' > d_0 \wedge D(d')]]$

(69) $\lambda x \exists d' [\text{good}'(\text{paper}')(x, d') \wedge d' > d_0]$

(70) $\lambda P \forall x [\exists d' [\text{good}'(\text{paper}')(x, d') \wedge d' > d_0] \rightarrow P(x)]$

We conclude the discussion of this aspect by an example which demonstrates that not only presence or absence of an overt complement is relevant for the scope of

the comparative operator, but also the syntactic position of the complement. The example is taken from German, since corresponding English examples are not available. In general, German comparative constructions behave pretty much like English ones, under the aspects discussed in this paper.

- (71) jeder mehr auf seinen Mercedes als auf seine Intelligenz stolze Mann
every more of his Mercedes than of his intelligence proud man

In this kind of construction, the complement is embedded in the AP, therefore the comparative operator does not take scope over the determiner in the comparative NP, and, accordingly, the construction is ok with the universal determiner.

5. Residual Issues

In this section, we will shortly comment on three problems of different status that are in need to be treated, but were not discussed in the paper, for different reasons.

First, we argued in the last section that the same kind of a non-local scoping mechanism applies to the comparative operator, as is responsible for NP scope assignment. However, the usual mechanisms would be predicted not to apply at all, in this case, because it does violate syntactic constraints: The prenominal AP is a scope island (as well as an island for wh extraction). Thus, if we take the operator to be attached to the comparative morphem, we have to assume a special kind of raising that is not subject to the usual constraints. But the problem is not one of the specific treatment that we propose. If you take the *than* to be the bearer of the operator, instead of the comparative morphem, you have to explain the fact that the operator can bind the variable position inside the AP. Even if you take the position that the semantic analysis should be completely revised, you are left with a similar problem outside semantics, *i.e.*, the fact that a grammatical feature embedded in the AP has to be passed up to the NP, in order to license the comparative complement. There is a problem, but it is not one of a special theory, but rather a very general and fundamental one. There are syntactic and semantic dependencies that violate any locality conditions which have been formulated, so far, but are nevertheless bound to some kind of local domain.

Second, our semantic analysis of comparatives are based on versions of the complement that are given in (72) and (73). We made use of structures such as:

- (72) John is taller than [(wh)_d [IP Bill is (d-tall)]]

- (73) John wrote [NP [NP a [N' better paper]]
[XP than [CP [C (wh)_d] [IP Bill wrote [NP (a d-good paper)]]]]]

The part of the complement written in parentheses, the "degree phrase", which can be a AP or NP, is never phonetically realized. We should say a word about the kind of mechanism that makes the missing parts available.

We assume that the semantics of the missing AP/NP is provided by an anaphoric mechanism. Note that in German, and similarly in French, the degree phrase position is filled by an overt pronoun.

(74) Hans ist größer, als Peter es ist.

(75) Hans is taller than Peter (it) is _.

(76) Hans hat ein interessanteres Buch gelesen, als Peter ihm eines empfohlen hat

(77) Hans read a more interesting book than Peter recommended (one) to him.

In predicative constructions such as (74) and (75) the anaphora refers to the AP of the matrix clause. In attributive constructions, it is the N' of the host NP rather than the NP itself that serves as the antecedent, which can be seen from examples like (78).

(78) John wrote at least one better paper than Bill had written _.

We take the gap to be filled in by a standard indefinite (existential) term the restriction of which is provided by the N' anaphora. The German data support this view, since the anaphoric mechanism in German complements is to a great extent identical in morphological and semantic properties with ordinary One Anaphora. The internal syntactic structure we assume for the complements of (75) and (77) are given in (79) and (80), respectively.

(79) John is taller than [(wh)_d [IP Bill is pro_{Adj}]]

(80) John wrote [NP [NP a [N' better paper]]
[XP than [CP [C (wh_d)] [IP Bill had written [NP (a) pro_{N'}]]]]

The N' content to be copied by the anaphoric mechanism, in (63), is the predicate $\lambda x[\text{good}'(\text{paper})(x,d)]$. Note that there is no problem with the comparative form of the antecedent under the assumption that anaphora essentially refers to the level of semantic representation (or to the level of LF, cf. [Sag 1976] and [Williams 1977]). For independent reasons, we have assumed before that the comparative operator is raised to the NP level, leaving the positive as the AP denotation. There is a problem, however with the unbound variable *d* in the anaphoric predicate: In order to obtain the right interpretation for the complement it has to be bound by the *wh* element in the head of the complement clause. We have an idea how to treat the problem, but we will postpone its presentation to a subsequent publication.

Finally, a problem arises with our analysis of quantifiers in comparative complements, in Section 3. (It was pointed out to us by Irene Heim, in personal communication.) We used the technique of Nested Quantification in order to show that wide scope readings for comparatives occurring in complements are possible without violation of island constraints. For ordinary headed relative clauses, the

theory predicts that wide scope readings are not available, since relative clauses do not denote terms. Free relatives, however, can be interpreted as denoting universal terms, and accordingly should allow application of nested quantification. But wide scope readings of quantifiers occurring in free relatives have a somewhat marked status. Our theory may overgenerate, in this respect.

6. Concluding Remarks

Our analysis relies heavily on the operation of Functional Composition, in particular, the special case of Nested Quantification. The main results that we have reported in this paper are:

- an analysis of the interaction between comparative operator, comparative complement and quantifier terms occurring in the complement that provides the appropriate scope readings and at the same time conforms with the relevant syntactic constraints
- a semantic account of attributive comparative constructions that explains the peculiar indefiniteness constraint on the determiner of the comparative NP as well as the fact that the constraint is dependent of the presence of an overt comparative complement
- a semantic interpretation of predicative and attributive comparatives and their complements that assigns a semantic representation to those surface constituents which have an independent content, according to speakers' intuitions.

Our analysis relies heavily on the operation of Functional Composition, and in particular on the special case of Nested Quantification. We do not consider this operation to be merely a technical device. There are various data that support the view that Nested Quantification is not a marked and non-standard operation that is forced in certain special circumstances. It rather appears to be as freely available as the standard Quantification into S rule, and it leads to a more natural and more local account of semantic composition, in comparative constructions as well as in other areas of natural language structure.

* The research reported in this paper was supported by the Deutsche Forschungsgemeinschaft (Project GuK), in the Special Program on Cognitive Linguistics.

References

- Cresswell, M.J.: 1976, The Semantics of Degree, in B. Partee (ed), *Montague Grammar*, Academic Press, New York, 261-292
- Heim, I.: 1985, *Notes on Comparative and Related matters*, unpublished manuscript, Austin, Texas
- Hellan, Lars : 1981, *Towards an Integrated Analysis of Comparatives*, Gunter Narr Verlag, Tübingen 0), 1-45
- Ladusow, Williams A: 1979, *Polarity Sensitivity as Inherent Scope Relations*, PhD diss. Austin, Texas
- Lerner, J.-Y.: 1991, *Quantorenanhebung bei Komparativkonstruktionen*, GuK-Report 18 , Universität des Saarlandes, to appear in *Linguistische Berichte*
- May, Robert: 1985, *Logical Form. Its Structure and Derivation*, The MIT Press., Cambridge/Mass.
- Pinkal, Manfred: 1989, Die Semantik von Satzkomparativen, *Zeitschrift für Sprachwissenschaft* 8, 206-256
- Rooth, Mats E.: 1985, *Association with Focus. Dissertation*, Department of Linguistics ,South College University of Massachusetts at Amherst
- Russell, B.: 1905, On Denoting, *Mind*, 14 (1905), pp.479-93
- Sag, Ivan A: 1976, *Deletion and Logical Form*, Ph.D. thesis, M.I.T., available from Indiana University Linguistics Club
- Seuren, Pieter M: 1978, The structure and selection of positive and negative graduable adjectives, *Proceeding of the 14th Regional Meeting of the Chicago Linguistic Society* , 336-346
- v. Stechow, Arnim: 1984, Comparing Semantic Theories of Comparison, *Journal of Semantics* 3, 1-77
- Williams, Edwin S: 1977, Discourse and Logical Form, *Linguistic Inquiry* 8, 101-139

Bare common nouns and their relation to the temporal constitution of events in Hungarian

MÁRTA MALECZKI

**Department of General and Applied Linguistics
Attila József University
Szeged, Hungary**

This paper supports with some data the widely accepted view that there is a close connection between the structured domains of objects and events. Data considered here are from Hungarian, thus the conclusions of these investigations certainly cannot carry over directly to English or other languages. However, I hope that the data presented here with their modell-theoretic explanation deserve some attention.

The main theoretical conclusion of the present investigations is that whenever the object of a verb refers cumulatively, the mapping relating it to the event structure has to be not only a homomorphism, as e.g. Dowty (1991) and Krifka (1989) assume, but has to be an injective homomorphism.

The paper is organized as follows:

In the first part the relevant Hungarian data and the main questions related to them are presented.

In the second section the claim that all Hungarian singular bare common nouns refer cumulatively will be supported by some convincing data. It is argued that the indefinite article and the plural morpheme put restrictions on the reference domain of singular bare common nouns in a compositional and monotonic way.

The third part is devoted to Hungarian verbal predicates and their denotations on the event-structure.

The fourth section sketches the main interdependences between the temporal constitution of events and the arguments participating in them.

1. The relevant linguistic data

1.1. Data to be examined, data to be neglected

In Hungarian the presence of bare singular common nouns in well-formed syntactic constructions is quite common, irrespectively whether they are count or mass.

- (1) Mari tea-t iszik.
Mary tea-acc. drinks
'Mary is drinking tea.'
- (2) Péter level-et ír.
Peter letter-acc. writes
'Peter is writing a letter/letters.'

In the examples above bare common nouns in the accusative contribute to the meaning of the sentence compositionally. Singular bare common nouns (CNs from now on) can also occur in the nominative with some intransitive verbs (see (3))¹, and even in the other cases² with other verbs (see (4), (5)). When the CN is in some oblique case, the meaning of it usually joins with the meaning of the verb non-compositionally (examples (4), (5)).

- (3) Vendég érkezett.
guest arrived
'A guest has arrived.'
- (4) Péter fej-be vágta Pált
Peter head-into cut Paul
'Peter hit Paul on the head'
- (5) Egy morzsa cigányút-ra ment.
a crumb gipsy-way-on went
'A crumb went down the wrong way.'

For the sake of clarity the data considered here will be restricted to neutral sentences with bare common nouns in the accusative. The intention with this restriction is to keep different things separately. It is the structure

of the nominal and verbal predicates and their relation that is to be examined here, and they may be blurred by interacting with focusing, not to mention the non-compositional contribution of the meaning of the parts to the meaning of the whole verb-phrase. Hopefully it will be possible to get some explanation also to the other interesting occurrences of bare common nouns on the basis of the present investigations.

1.2. Verbs with and without CN-argument³

It is a fact of Hungarian that although constructions with bare common nouns are very common, the class of verbs allowing for compositional CN-objects in neutral sentences is fairly restricted:

- (6) *Péter level-et meg-írt.
Peter letter-acc. PFX-wrote

- (7) *Mari koldus-t észrevett.
Mary beggar-acc. noticed

These verbs may take only NP-objects:

- (8) Péter megírt egy level-et.
Peter PFX-wrote a letter-acc.
'Peter has written a letter.'

- (9) Mari észrevett egy koldus-t.
Mary noticed a beggar-acc.
'Mary noticed a beggar.'

A syntactic solution to this problem may be based on the assumption that Hungarian has a position under V' that is possibly filled by one constituent, called verb carrier in Kálmán et al. (1985). Verbs like the ones above have already built this constituent into themselves: in (8) it is the prefix, in (9) the 'észre' part of the verb. Since the position under V' has already been filled by these elements, bare common nouns, which are also considered verb carriers, are ruled out. However, several questions remain open in such a descriptive solution; for instance, why are CNs verb carriers, what are the criteria of being a verb carrier, why can verbs take only one verb carrier, etc.

Syntactically, CN-V constructions⁴ belong to the class of verb carriers and form complex predicates with the verb. However, as Komlósy (1985) argues, there are two groups of complex predicates in Hungarian according to their origins. One group is constructed in the lexicon: prefixed verbs and idioms belong here. Members of the second group arise through syntactic processes, and predicates with compositional bare CN complements belong to this group. That bare CNs are some kind of modifiers and they are not really incorporated into the meaning of the verb can be illustrated by the fact that they may have adjectives:

- (2a) Péter hosszú level-et ír.
 Peter long letter-acc. writes
 'Peter is writing a long letter.'

Anyway, it is a fact of Hungarian that there is an asymmetry in the subcategorization possibilities of Hungarian verbs. While there are verbs that take NP-arguments, and exclude bare CN arguments (see (6), (7)), verbs allowing for a bare CN-argument can always take the same argument with the same thematic role with some determiner, as exemplified in (10) (cf. (2)):

- (10) Péter írt egy levelet.
 Pete wrote a letter

In categorial terms, if a verb has an $(S \backslash NP) \backslash CN$ category, it always has an $(S \backslash NP) / NP$ category, but not vice versa. As categories are given lexically, we may assume that some lexical property of the verb determines whether it may or may not take a bare common noun argument. On the basis of the connection between the nominal and verbal reference revealed in e.g. Bach (1981), Bach (1986), Dowty (1991), Krifka (1989), it may be expected that there is a feature shared by bare common nouns and verbs allowing for them. Consequently, if we examine the semantic characteristics of Hungarian CNs more closely, we may get some clue to the problem what lexical property of verbs may be blamed for prohibiting or allowing for them.

2. The algebraic structure of Hungarian nouns

The thorough investigation of the count/mass distinction and the distributive, collective and group-reference of nouns lead to the conclusion that the domain of objects is structured (e.g. Link 1983, Landman 1989). On a structured domain other referential properties of nouns and noun-phrases can also be explicated conveniently (Krifka 1989). The most important of these for the present purpose is the cumulative/quantized distinction.

2.1. Cumulative versus quantized reference

Let us assume with Landman (1989), Krifka (1989) that in our model the set of objects is structured as a complete join semilattice without bottom element.

An $\langle A, * \rangle$ pair containing a set A with an operation $*$ defined on A is called a semilattice iff the $*$ operation is idempotent, commutative and associative.

We may define a $+$ (join) operation having these properties and being interpreted in a partially ordered set A as the operation yielding for every $\{a, b\}$ subset of A the supremum of a and b : $a + b = \sup\{a, b\}$.

$\sup\{a, b\}$ is defined in a partially ordered set A as the lowest element of A among its elements higher than both a and b .

If the join operation $+$ is extended to every subset of A , the semilattice is complete. If there is no b in A such that b is equal to or lower than every element of A , the lattice contains no bottom element.

The extensions of nominal predicates, whether lexicalized or not, may be considered as sub-lattices of the object domain A , with the same properties: they are complete join semilattices, without bottom element (Krifka 1989).

The cumulative/quantized distinction may be roughly characterized as preserving the lattice structure versus selecting some part of it that does not form a semilattice. More exactly, a predicate refers cumulatively iff for every two a, b of its domain, $a + b$ is also in the domain of the predicate, in other words, the extension of the predicate is closed under the $+$ operation. This means that the predicate covers either the whole domain of objects it may denote, putting no restriction on the semilattice, or selects some part forming itself a complete join semilattice. Cumulative reference does not depend on whether the noun is count or mass. Countability can be represented by the atomicity of the lattice, so count/mass distinction has got to

do with the property whether the semilattice has minimal parts or not. This distinction is not interesting for the present purposes as count nouns may also refer cumulatively if they satisfy the definition given above. On the other hand, mass nouns may become quantized if their domains are restricted by some measure function as described in Krifka (1989).

Thus the quantized/cumulative distinction is not bound to the count/mass distinction in any way. For the sake of simplicity atomic domains will be assumed from now on.

2.2. The denotation of Hungarian nouns and noun phrases

There are pieces of evidence that singular bare common nouns refer cumulatively in Hungarian, contrary to their singular form and irrespectively whether they are count or mass. Data supporting this assumption are of two kinds: one group of them is directly connected to the semantic intuition, and so to the use of language, while the other group may be labelled as language specific agreement phenomena.

1. The truth-conditions of sentences containing countable bare common noun objects are such that if there is anything in the extension of the object noun the predicate may be asserted of, the proposition will be true independently of the exact number of the things the noun denotes (assuming the other arguments of the verb are O.K.). This opinion is shared by every interviewed native speaker without the slightest doubt. Thus example (11) is true irrespectively whether Peter stole one dog, two dogs, etc., up to the number of all dogs in the universe.

- (11) Péter kutyá-t lopott.
 Peter dog-acc. stole
 'Peter stole dogs/a dog.'

These truth-conditions are sometimes clearly reflected by the context. For instance, there is a test in Krifka (1989), copied as (12), which shows that bare plurals in English may denote singular objects. This test is applicable to Hungarian exactly in the opposite way, i.e. if the use of singular and plural is reversed, as shown in (13).

- (12) – Do you have children?
 – Yes, I have one child./*No, I have (only) one child.

- (13) – Van gyerek-ed?
 is child-your
 'Do you have children?'
 – Igen, van három fiam./ *Nincs, de van három fiam.
 Yes, is three son-my / * No but is three son-my
 'Yes, I have three sons.' / * 'No, but I have three sons.'

2. There are some agreement phenomena in Hungarian, exemplified in (14) and (15), that remain unexplained without the assumption that bare CNs refer cumulatively.

One such phenomenon is that in every sentence with conjoined subjects the nominal predicate may be a singular CN, and the predicate refers to each member of the conjoined subject. There is no reading relating the predicate to only one member of the coordinated subject.

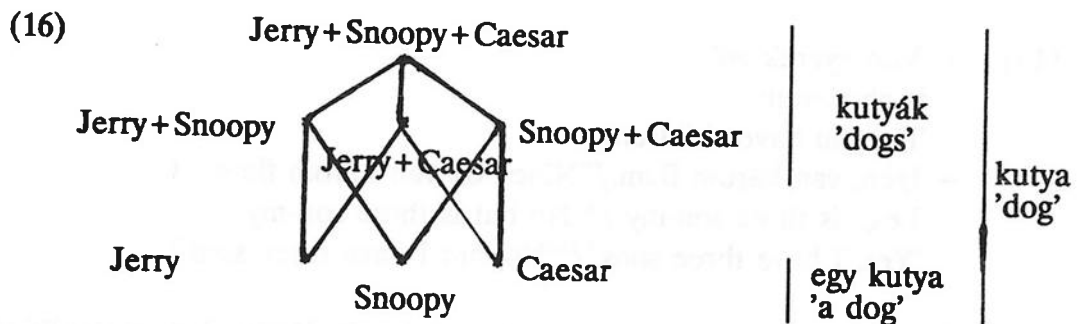
- (14) Alex és Bella macska.
 Alex and Bella cat
 'Alex and Bella are cats.'

The other agreement phenomenon that can be easily explained by assuming cumulative reference for singular common nouns is that nouns after numerals are obligatorily in the singular in Hungarian:

- (15) nyolc asztal, 365 tigris
 eight table 365 tiger

On the basis of these data we can reasonably assume that the extension of Hungarian nouns and noun-phrases may be given on an object-domain defined as in 2.1. in the following way. A complete join semilattice is assigned lexically to every singular common noun, so they get uniformly the type of extension commonly assigned to mass nouns or plural countable nouns in English. Plural morpheme and the indefinite article egy 'a/an' put restrictions on the domain of bare singulars selecting one part of the semilattice, thus dividing it into two disjoint parts.

Taking a universe with three dogs, the extension of kutya 'dog', kutyák 'dogs', egy kutya 'a dog' can be given as shown in (16).



As the reference of Hungarian bare plurals always implies the presence of more than one object, they are to be interpreted like the semilattice of their singular counterparts without the atoms. Thus the meaning of the plural morpheme in Hungarian can be described as subtracting the atoms from whatever domain it operates on:

$$(17) \text{Dom}(\text{plural}(X)) = \text{Dom}(X) \setminus \text{ATOMS}(X)$$

On the other hand, the atoms (without their sums) constitute the interpretation of singular nouns with the indefinite article egy 'a/an':

$$(18) \text{Dom}(\text{egy}(X)) = \text{ATOMS}(X)$$

So it is the meaning of the indefinite article which selects the singular elements not the singular form of the noun, while the plural morpheme selects the plural elements. In the example (16) above, *kutya* 'dog' has the whole K semilattice as its domain, the domain of *kutyák* 'dogs' is K minus the atoms of K, while the domain of *egy kutya* 'a dog' consists of the atoms of K.

By incorporating the atomicity into the meaning of the plural morpheme and the indefinite article in this way, two distributional properties of mass nouns, which have non-atomic semilattice interpretation, automatically follow. In non-atomic lattices we cannot separate atoms, so we cannot subtract them. As the indefinite article *egy* 'a/an' selects the atoms of a semilattice, while the plural morpheme the complement of the atoms, mass nouns neither get indefinite article, nor stand in plural, by the same reason. If it happens that a lexically mass noun is preceded by *egy* 'a/an' or followed by the plural morpheme, it is interpretable only as if it had atoms, thus it will have necessarily a countable meaning.

The interpretation given above for nouns and noun-phrases is both compositional and monotonic. It is compositional as a well-defined meaning is assigned both to the plural morpheme and the indefinite article, which interacts with the lexically given meaning of the noun in a computable fashion. Moreover, as the plural morpheme and the indefinite article add new information to the lexically available meaning of the common nouns, and this more specific information restricts the interpretation domain, the requirements of monotonicity are also met.

3. The structure of events⁵

It is widely accepted that there is another structured domain of entities disjoint from the set of objects: the domain of events (see e.g. Bach 1981, 1986, Dowty 1991, Krifka 1989, Link 1987). In the same way as objects, events also constitute a complete join semilattice without bottom element. Verbal predicates have their denotations in this domain, independently of whether they are single lexical units or composed of a verb and some other, lexical or phrasal expression(s).

On the event-lattice also exists the semantic counterpart of the quantized/cumulative reference, called here telic/atelic distinction.

3.1. The telic/atelic distinction and the Vendler-classes of verbal predicates

The intuition behind the present usage of words telic and atelic is the following.

When exploring the reference domain of verbal predicates, the topic is the inner structure of event time, which is to be carefully distinguished from speech time and reference time (see Reichenbach (1947)). The latter two, called external times by Kiefer (1982), will be constantly neglected in what follows, as telicity is connected solely to the structure of event time.

A telic predicate differs from an atelic one roughly in that the former has some change inside its event time, while the latter has not. This means that the tenseless truth-conditions of a proposition containing a telic predicate depend on at least two sharply distinguishable events, characterizable by von Wright's T ('and next') operator as shown in (19) (see Dowty 1979):

(19) $pT\sim p$ or $\sim pTp$

That atelic verbs do not have a semantic character based on changing is expressible by the T operator as in (20):

(20) pTp or $\sim pT\sim p$.

Featuring by these properties the four Vendler-classes, we may say that statives and activities are lexically specified as atelic (know, run) while accomplishments and achievements are telic (write a letter, win a race)⁶.

However, there is another feature playing role in the classification of verbal predicates. While the telic/atelic distinction is based on the presence or lack of change in the inner structure of the predicate, another distinction can be made on the basis of the course of events. Some predicates, called durative, denote an action that needs some stretch of time to be fulfilled, independently of the telicity (e.g. run - atelic, build a house - telic). The four Vendler-classes of predicates may be characterized by these features in the following way:

(21)	stative	activity	accomplishment	achievement
telic	—	—	+	+
durative	—	+	+	—
Examples:	know exist	run dig	build dig a hole	arrive realize

3.2. The denotation of verbal predicates

I will assume that atelic verbal predicates refer cumulatively, similarly to mass nouns in English and every common noun in Hungarian. The cumulative reference is meant here in the same way as above, so a verbal predicate refers cumulatively iff for every subset of events belonging to the extension of the predicate the join of these events is also there. Thus a cumulatively referring verbal predicate has a complete join semilattice interpretation on the semilattice of events, exactly as cumulatively referring nouns on the object lattice. The intuition behind this assumption can be made clear by some examples: two or three events of running may also form a running

(recall that external times are excluded now!), while two or three events of arriving cannot form one arriving.

As shown in (21), atelic predicates are statives and activities. However, there is a striking difference between the two with respect to their internal time-consuming properties. While activities are necessarily composed of smaller events, as they need some time to be fulfilled, states are timeless, as Bach's Gedankenexperiment shows: if we imagine a world with exactly one moment of time, it is possible to give the truth-conditions for a stative predicate in that world, but impossible to interpret predicates belonging to the other classes (Bach, 1981, p. 71.). In such a timeless world we can decide whether Peter loves Mary or knows a poem, but cannot answer the question whether he is running or painting a picture. This property of statives may be made explicit in the following way.

Statives as atelic predicates refer cumulatively, so they will get a semilattice interpretation. But since they have no inner event-structure, there is no need of more than one event representing their denotation. Consequently, the semilattices assigned to statives will contain only one element. In this way, although in a pathological fashion, they refer cumulatively, and there is nothing in their denotation structure indicating that they need time to be interpreted.

This kind of solution is not unknown on the object-domain: Landman (1989) argues that singular definite noun-phrases have exactly this sort of denotation.

The next problem is how to model the achievements. If the non-durative property is represented by some strict limitation of the number of the elements constituting the domain of the predicate, then achievements also have to meet this kind of restriction. But as we have seen in (19), their telic specification involves some change, and this is possible only if we have at least two distinct events. However, two events are enough for indicating change, so I assume that the restriction put on the domain of achievements is that they have exactly two events in their denotation. Keeping in mind that telicity is characterized here as in (19), the two events constituting the denotation of an achievement may be considered as relative complements in the event lattice. As these are not ordered with respect to each other, the denotation of achievements can never constitute a semilattice; thus cumulative reference will not be possible⁷.

Thus while the atelic property may be modelled by having a lattice structure, durativity may be modelled by having more than two elements in the extension of the predicate.

On the basis of the above reasoning, accomplishments, being telic, will not have lattice structure, but being durative, have more than two events in their extensions. As accomplishments have usually some complement or argument, their characterizing properties are expected to arise as a consequence of the interaction of verbal and nominal denotations. Thus the analysis of accomplishments leads us to our next topic.

4. The connection between the domains of objects and events

4.1. Data

Arriving at the point where it is possible to answer our main question raised in 1.2., first I will present data illustrating how bare common noun objects fit the four Vendler-classes of verbs.

1. Stative verbs

Statives do not normally have bare common noun objects in neutral sentences⁸. The * indicates that these sentences are unacceptable with neutral intonation:

(22) *Péter verset tud.
Peter poem knows

(23) *Péter lányt utál.
Peter girl hates

2. Activity verbs

Transitive activity verbs may have bare common noun objects:

(24) pénzt keres
money-acc. earns

(25) autót vezet
car-acc. drives

As we have seen above in the examples of (21), one and the same verb may be the head of a VP that is an activity or an accomplishment, depending on partly the presence or the absence of its object, partly on the structure of its object. That is also the case in Hungarian.

3. Accomplishment verbs

Similarly to English, a class of verbs being activity when used intransitively becomes accomplishment with an object. The difference is that Hungarian verbs belonging to this group (e. g. ír 'write', ás 'dig') may take countable CN or NP-objects equally well. The interesting thing is that when the argument is a bare CN, the reference of the whole predicate will be always atelic, in accordance with the observations of Verkuyl (1972) (cf. for Hungarian Kiefer 1982):

- (26) Albert vermet ás
 Albert pit dig
 'Albert is digging a pit'

4. Achievement verbs

Verbs belonging to this class typically cannot take bare common noun objects, as examples (6), (7) show, repeated here:

- (6) *Péter level-et meg-írt.
 Peter letter-acc. PFX-wrote
- (7) *Mari koldus-t észrevett.
 Mary beggar-acc. noticed

In Hungarian the most striking overt devices of telic/atelic specification beside the arguments are the verbal prefixes and other verb carriers. These usually specify predicates as telic. Since they mostly contribute to the meaning of the verb phrases non-compositionally, they are considered as constituting one lexical unit with the verb. So I take prefixed verbs as one word here, and I will not bother about how verbal prefixes change the original character of the verb (not denying with this decision that it is an interesting question).

4.2. Explaining data

Dowty (1991), Krifka (1989) assumes that the cumulative/quantized properties of arguments carry over to the verb phrase because the thematic roles relating arguments to the events are functions with special properties, more closely, they are structure preserving homomorphisms from objects to events.

Putting aside now the carefully defined distinctions thematic roles may have (see Krifka 1989, pp. 92-96), for the present purpose it seems enough to stipulate that these homomorphisms have to be injective (one-to-one mappings). Thus they will preserve not only the ordering but also the structure of their domain, embedding the denotation of the noun into the denotation of the verb.

However, the relation between objects and events may be a homomorphism only if both the domain and the range of the function is structured. This follows from the definition of homomorphism:

- (27) If h is a homomorphism mapping $\langle A, * \rangle$ to $\langle B, \# \rangle$, then for every a, b elements of A $h(a*b) = h(a)\#h(b)$.

When the domain or the range or both of them are unstructured, the injective homomorphism criterion naturally loses its sense. Moreover, the mapping becomes a relation that may or may not be a function, and while it is possible to relate unstructured denotations of nouns to some structured verbal denotation, this seems to be impossible vice versa (see later).

Thus we require that whenever the denotation of the noun is cumulative, the mapping relating it to the verb has to be an injective homomorphism (an embedding).

If we put together what has been said about the referential properties of Hungarian nominal and verbal predicates and the assumption that whenever a homomorphism connects them it has to be injective, we get the following explanation to the data presented above.

A. Statives

The denotation of a bare common noun is a normal semilattice with at least two elements in every linguistically interesting model, see 2.2., while the denotation of statives is a one-element lattice. Thus the function, which would map the denotation of a CN to the denotation of a stative, cannot be injective, although it may be a (bit deformed) homomorphism.

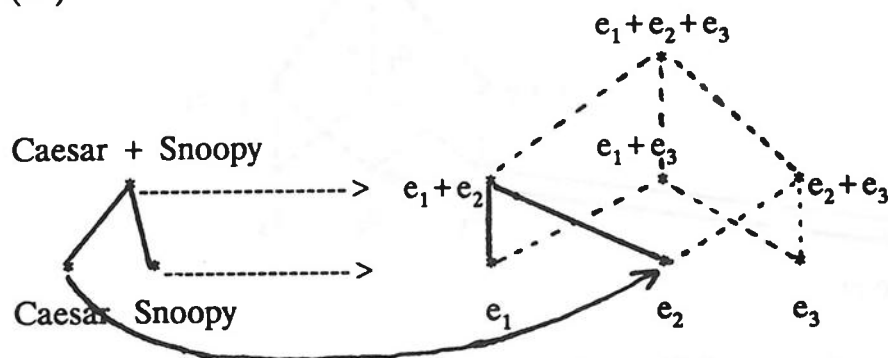
That is why sentences like (22), (23) are unacceptable.

B. Activities and accomplishments

As the denotations of activity verbs are normal semilattices, and so are that of bare common nouns, it is possible to have homomorphisms from CNs into activities. Requiring that these are injective, they embed the semilattices of nouns into the semilattices of verbs. Thus the extension of a verbal

predicate becomes specified by its object: a function mapping autó 'car' into the semilattice of vezet 'drive' will select those events from the denotation of the verb that are car-drivings. A function mapping kutya 'dog' into the semilattice of events of lop 'steal' will select those events from the semilattice of lop 'steal' that are dog-stealings. As the mapping is an injective homomorphism, it will yield a cumulatively referring set of events whenever its domain is cumulative. Thus a sublattice of lop 'steal' isomorphic to the semilattice of kutya 'dog' will be the denotation of kutyát lop 'dog-steal'. This is exemplified in (28) on a universe with two dogs and three stealing events:

(28)



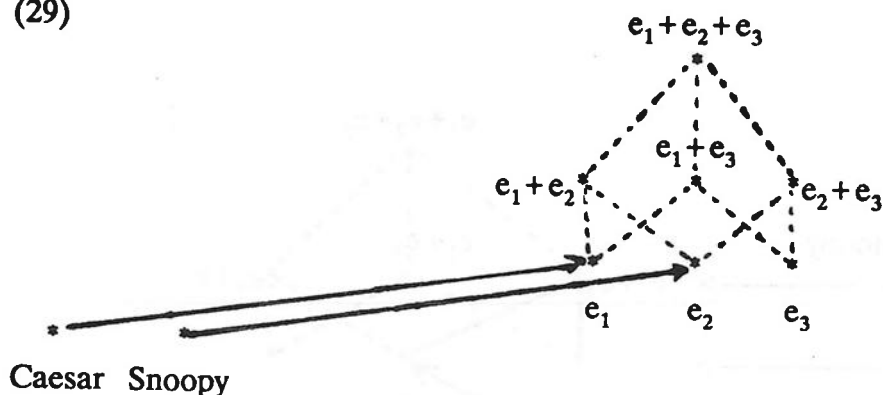
A prime defining feature of activities is that they refer cumulatively. If their cumulative reference becomes impossible because of the reference properties of the object, activities turn into accomplishments. Thus ír 'write' is a lexically cumulatively referring verb when used intransitively and it remains cumulatively referring when its object is a bare common noun (levelet ír 'letter writes'), but it becomes an accomplishment when its object is an indefinite NP (ír egy levelet 'write a letter').

Accomplishments have an object or something else restricting to something or directing in some way the activity described by the verb. This restriction may be lexically given, as in the case of some prefixed verbs (e.g. elpusztít 'destroy') or may take place with the aid of some other phrasal or lexical unit (e.g. el sétál v hová 'walk to NP', ír egy levelet 'write a letter'). In the latter case the cumulative or quantized reference of the predicate depends on the referential properties of the restricting unit and the mapping relating it to the event-structure.

When the denotation of the object is not structured while the denotation of the verb is a semilattice, we have theoretically at least three possibilities:

1. The relation is a function, which maps the denotation of the noun into the structured denotation of the verb in such a way that the range does not constitute a semilattice. In this case the predicate becomes telic, and turns into an accomplishment. For instance, the extension of *lop egy kutyát* 'steal a dog' will be the following on the same domains as above:

(29)



Caesar Snoopy

2. The other possibility is that the relation is a function which maps the denotation of the noun in such a way that the range is a structured part of the denotation of the verb. It seems, however, that in this case one and the same object is assigned to more than one events, thus we reach at the third possibility:

3. The relation is not a function but a one-many relation, where the range of the relation is structured.

This is the case with verbs like *read*, where the atelic character of the verb is transferred to the complex predicate, but the cumulatively referring event-structure has to be assigned to one book:

- (30) Péter olvas egy könyvet.
Peter reads a book
'Peter is reading a book.'

C. Achievements

Since achievement verbs have no lattice structure, as was argued in 3.2., a cumulatively referring argument cannot be embedded into their denotations. This explains the unacceptability of sentences like (6) or (7).

5. Summary

The crucial lexical property that characterizes the class of Hungarian verbs taking bare common noun objects is the cumulative reference.

Since the functions mapping nouns with structured denotations to events are injective homomorphisms, cumulatively referring (atelic) verbs may take CN objects while telic verbs may not.

Thus the answer to the question raised in 1.2. is that the lexical property shared by CN-taking verbs and bare common nouns is their cumulative reference, and the cumulative reference of the VP containing CN-object is guaranteed by requiring the injectivity of the homomorphism relating the object to the verb.

Notes

1. These verbs show the so-called definiteness effect, (DE) as demonstrated by Szabolcsi (1986). However, I do not want to deal with the whole class of Hungarian DE-verbs at present. As far as I can see, the verbs I examine here constitute a proper subset of DE-verbs: the transitive ones. The indefinite argument of transitive DE-verbs is always their object.
2. There is no agreement upon the exact number of nominal cases among Hungarian linguists. The supposed minimal number is seventeen.
3. In papers on Hungarian syntax bare common nouns are considered modifiers not arguments. However, my concern is semantic in nature, and I hope that this paper makes clear that bare common nouns are arguments from a semantic viewpoint, even if with special properties.
4. The syntactic position of the bare CN, irrespectively whether singular or plural, is the same as that of the other verb-carriers: it always precedes immediatly the verb in neutral sentences.
5. The use of the word 'event' is ambiguous here, as in the whole linguistic literature. Here either the denotation of a predicate is called event (cf. eventuality, Bach, 1986), or the basic entities constituting the reference domain of predicates. However, I hope that the context disambiguates this word whenever it is necessary. In this particular case both meanings are intended.
6. I am aware that statives are usually treated separately from "real" events, and it makes sense to characterize as telic or atelic only the latter ones but not the former. However, if atelicity is defined as in (20), then statives clearly belong to this class.
7. If the reader feels this reasoning counterintuitive, then (s)he can choose another way to prevent cumulative reference. If we stipulate that the ordering relation is strict in the denotation of achievements, the two events will not be closed under the join operation.
8. Incorporation of the object is sometimes possible, but these being idiomatic expressions fall from the scope of the present investigations.

References

- Bach, E.: 1981, On Time, Tense, and Aspect: An Essay in English Metaphysics, in P. Cole (ed), *Radical Pragmatics*, Academic Press, New York, pp. 63-81.
- Bach, E.: 1986, The Algebra of Events, *Linguistics and Philosophy* 9(1), 5-16.
- Dowty, D.: 1979, *Word Meaning and Montague Grammar*, Reidel, Dordrecht.
- Dowty, D.: 1991, Thematic proto-roles and argument selection, in *Language* 67(3), pp. 547-619.
- Kálmán, L. - Prószéky, G. - Nádasdy, Á. - Kálmán, C. Gy.: 1987, Hocus, Focus, and Verb Types in Hungarian Infinitive Constructions, in Abraham, W. - de Mey, S. (eds.), *Topic, Focus, and Configurationality*, Benjamins, Amsterdam, pp. 129-142.
- Kiefer, F.: 1982, The Aspectual System of Hungarian, in Kiefer (ed.), *Hungarian Linguistics*, John Benjamins, Amsterdam, pp. 293-329.
- Komlósy, A.: 1985, Predicate complementation, in Kenesei (ed.), *Approaches to Hungarian I* JATE, Szeged, 53-78.
- Krifka, M.: 1989, Nominal Reference, Temporal Constitution and Quantification in Event Semantics, in R. Bartsch, J. v. Benthem and van Emde Boas (eds.), *Semantics and Contextual Expression*, Foris, Dordrecht, pp. 75-115.
- Landman, F.: 1989, Groups, I, *Linguistics and Philosophy* 12(5), 559-605.
- Link, G.: 1983, The Logical Analysis of Plurals and Mass Terms: A Lattice-Theoretical Approach, in R. Bauerle, Chr. Schwarze & A. v. Stechow (eds.), *Meaning, Use and Interpretation of Language*, de Gruyter, Berlin, 302-323.
- Link, G.: 1987, Algebraic Semantics for Event Structures, in J. Groenendijk, M. Stokhof & F. Veltman (eds.), *Proceedings of the Sixth Amsterdam Colloquium*, ITLI, Amsterdam, 243-262.
- Reichenbach, H.: 1947, *Elements of Symbolic Logic*, University of California Press, Berkeley.
- Szabolcsi, A.: 1986, From the Definiteness Effect to Lexical Integrity, in Abraham, de Meij (eds), *Topic, Focus and Configurationality*, John Benjamins, Amsterdam, pp. 321-348.
- Verkuyl, H. J.: 1972, *On the Compositional Nature of the Aspects (Foundations of Language, Supplementary Series, Vol. 15)*, Reidel, Dordrecht.

FREE CHOICE PERMISSIONS AND ROSS'S PARADOX: Internal vs External Nondeterminism

J.-J.Ch. Meyer

Free University Amsterdam & University of Nijmegen

ABSTRACT

In this short note we show how the well-known paradox of free choice permissions in deontic logic can be understood in terms of nondeterministic actions. Free and "imposed" permissions can be treated adequately by distinguishing between two kinds of nondeterministic choice operators: one that expresses an *external* choice, which cannot be influenced by the actor, and one that expresses an *internal* choice, on which the actor *has* an influence. Similar notions of nondeterminism are well-known in computer science where they are used in the context of parallel execution of programs with the possibility of synchronisation. We also apply the insights gained by this approach in a discussion on the well-known paradox of Ross.

1. Introduction

What in the literature (cf. [Ka],[Hil]) is known as the problem of free choice permissions can be explained concisely as follows:

In the standard system of deontic logic [vW] and many other deontic systems it is a theorem that

$$P\alpha \supset P(\alpha \vee \beta) \quad (1)$$

where P is the permissibility operator. This conflicts with our intuitions concerning the nature of permissions as is shown by striking examples as

- (2) "You are allowed to mail this letter" implies "You are allowed to mail this letter or burn it".

and

- (3) "You may buy oranges" implies "You may buy oranges or (buy) bananas".

This problem cannot be resolved by just putting *two* operators P_W and P_S into the standard system(s), denoting *weak* and *strong* permissibility, respectively, and satisfying

$$P_W(\alpha \vee \beta) \equiv P_W\alpha \vee P_W\beta \quad (4)$$

$$P_S(\alpha \vee \beta) \equiv P_S\alpha \wedge P_S\beta \quad (5)$$

Doing this yields new and even more serious problems regarding $P_S(\alpha \wedge \beta)$: it then holds that

$$P_S\alpha \supset P_S(\alpha \wedge \beta), \quad (6)$$

since $P_S\alpha \equiv P_S((\alpha \wedge \beta) \vee (\alpha \wedge \neg \beta)) \equiv P_S(\alpha \wedge \beta) \wedge P_S(\alpha \wedge \neg \beta)$. And (6) is definitely not in agreement with our intuitions about permissions: the fact that you are allowed to mail the letter does certainly not imply that you are allowed to burn it as well!

Kamp [Ka] proposes a solution where a 'focus' or 'scope' operator F is used to keep track of the disjuncts that are permitted in a disjunctive permission. Thus Kamp can have expressions like $P_K(F\alpha \vee F\beta)$ and $P_KF(\alpha \vee \beta)$, where the permission operator P_K resembles Von Wright's P_S . The former expression entails $P_KF\alpha$, the latter does not. As Hilpinen remarks in [Hil], Kamp's focus operator F can be regarded as a *selection* operator which selects the class of possible ways that are permitted in order to establish the result given as its argument.

Part of the problem as I see it is due to the use of ordinary logical connectives inside the scope of the P -operator. In fact, this feature gives rise to many other deontic paradoxes as well (cf. [Ca], [Me]). So, to start with, we must distinguish between *actions* (or *practitions*) and *assertions*. We shall do this by introducing disjoint classes for these two entities. We use α, β (with possible marks such as indexes) for actions and ϕ, ψ (with possible marks) for assertions.

2. Internal and External Choices

Consider a so-called 'free' permission P_F as opposed to an 'imposed' permission P_I , say

$$P_F(\text{buy oranges OR buy bananas}) \quad (7)$$

as opposed to

$$P_I(\text{mail the letter OR burn it}). \quad (8)$$

Both (7) and (8) contain OR-actions in which there is a choice to be made. We may call these actions nondeterministic. We may argue that it is not so much a difference of the kind of permission that has to be taken into account as well as a permission of two different kinds of actions: the action appearing in (7) "buy oranges OR buy bananas" is to be interpreted as an action in which there is a choice to be made such that it is upto the actor to decide whether the first alternative "buy oranges" is chosen or the second one "buy bananas". The action in (8) "mail the letter OR burn it" also denotes a choice of actions that has to be made, but this decision is not to be influenced by the actor; here it is the "environment" of the actor that decides (or has decided) which alternative is chosen. We may call the 'free' choice "buy oranges OR buy bananas" an '*internally*' nondeterministic action: the actor himself decides what alternative is chosen. The 'imposed' choice "mail the letter OR burn it" may be called an '*externally*' nondeterministic action: the decision as to which alternative has to be taken is beyond the control of the actor; it is decided by the external environment.*

We may represent this by using two different symbols to indicate what kind of choice we are dealing with. We shall use the notation $\alpha + \beta$ for an *external* choice between the actions α and β and the notation $\alpha \oplus \beta$ for an *internal* choice between α and β .

Now we may replace the 'imposed' permission $P_I(\alpha \text{ OR } \beta)$ by a permission of an external choice: $P(\alpha + \beta)$ and the 'free' permission $P_F(\alpha \text{ OR } \beta)$ by a permission of an internal choice: $P(\alpha \oplus \beta)$.

3. Formal Semantics of Internal and External Choices

Although the preceding may sound very convincing and clear, we have not gained much if we cannot handle internal and external choices in a formal way and give rules how permissions of such actions must be evaluated formally. The basic idea we use here is the same as in [Me] where permission and deontic operators in general are interpreted in a dynamic logic framework (cf. [Ha]). For instance, for an action α to be forbidden we write equivalently that the performance of α leads to liability to some sanction. In a dynamic logic expression we may write this as $[\alpha]V$, where the propositional variable V stands for liability to sanction. The semantics of the expression $[\alpha]\phi$ for arbitrary assertion ϕ is the following: the performance of action α in the current state (world) leads to a state (world) in which ϕ holds.

The framework of dynamic logic provides us with a lot of calculation axioms, rules and theorems for expressions of the form $[\alpha]\phi$ (cf. [Ha] and [Me]). As a consequence of these facts concerning dynamic logic expressions together with the axiom that some action is permitted iff it is not forbidden (F), i.e.

$$P\alpha \equiv \neg F\alpha, \quad (9)$$

we obtain that $P\alpha$ is equivalent with the dynamic logic expression $\langle\alpha\rangle\neg V$, where $\langle\alpha\rangle$ is the dual of $[\alpha]$:

$$\langle\alpha\rangle\phi \equiv \neg[\alpha]\neg\phi \quad (10)$$

The informal meaning of $\langle\alpha\rangle\phi$ is: the performance of action α may/can lead to (a state/world in which) ϕ . In [Me] we already treated nondeterministic actions denoted there as $\alpha\cup\beta$. In our present terminology these actions are *external* choices, so here we would write $\alpha+\beta$. Also in [Me] we saw that for these nondeterministic actions the following property holds:

$$[\alpha+\beta]\phi \equiv [\alpha]\phi \wedge [\beta]\phi. \quad (11)$$

This has as consequences (see [Me]) that it holds that:

$$F(\alpha+\beta) \equiv F\alpha \wedge F\beta; \quad (12a)$$

$$P(\alpha + \beta) \equiv P\alpha \vee P\beta. \quad (12b)$$

This is indeed in accordance with our intuitions regarding prohibitions and permissions of 'imposed' (external) choices. But how about 'free' (internal) choices? The answer follows from observing the situation with external choices:

In [Me] we gave a formal semantics for external choices on the basis of sets: let the interpretations $\llbracket \alpha \rrbracket$ and $\llbracket \beta \rrbracket$ of actions α and β be sets of semantical entities that denote the performance of actions, then $\llbracket \alpha \cup \beta \rrbracket$ is the set-theoretic union of $\llbracket \alpha \rrbracket$ and $\llbracket \beta \rrbracket$:

$$\llbracket \alpha \cup \beta \rrbracket = \llbracket \alpha \rrbracket \cup \llbracket \beta \rrbracket. \quad (13)$$

(Notation as in [Me].)

Furthermore, the interpretation of $[\alpha]\phi$ is given in [Me] as: let σ be some state (world) and let $W_{\alpha, \sigma}$ denote the sets of states (worlds) that one gets into after the performance of α in σ ; then

$$\sigma \models [\alpha]\phi \text{ iff } W_{\alpha, \sigma} \models \phi, \quad (14)$$

where for a set W of states, $W \models \phi$ is defined by $\tau \models \phi$ for all $\tau \in W$. The set $W_{\alpha, \sigma}$ is determined by the set of action denotations $\llbracket \alpha \rrbracket$ and the state σ (cf. [Me]). We therefore write $\llbracket \alpha \rrbracket(\sigma)$ instead of $W_{\alpha, \sigma}$, viewing $\llbracket \alpha \rrbracket$ as a function from states to sets of states. Validity is defined as usual: $\models \phi$ iff $\sigma \models \phi$ for all states $\sigma \in \Sigma$, the universe of states.

Obviously we can copy this in the present setting, but we also have to define a variant for internal choices. We do this in an obvious manner:

We use two different types of sets of states (and of action denotations): $\vee$ -sets and $\wedge$ -sets, denoted as $W^\vee$ and $W^\wedge$, respectively. Both types of sets have their own set-theoretic operators. Moreover, we may nest these sets hierarchically and consider e.g. an $\vee$ -set of $\wedge$ -sets of $\vee$ -sets. Now we define:

$$\llbracket \alpha + \beta \rrbracket(\sigma) = \{\llbracket \alpha \rrbracket(\sigma)\} \cup^\wedge \{\llbracket \beta \rrbracket(\sigma)\} (= \{\llbracket \alpha \rrbracket(\sigma), \llbracket \beta \rrbracket(\sigma)\}^\wedge); \quad (15a)$$

$$\llbracket \alpha \oplus \beta \rrbracket(\sigma) = \{\llbracket \alpha \rrbracket(\sigma)\} \cup^\vee \{\llbracket \beta \rrbracket(\sigma)\} (= \{\llbracket \alpha \rrbracket(\sigma), \llbracket \beta \rrbracket(\sigma)\}^\vee). \quad (15b)$$

Furthermore, we maintain (14) where now $W_{\alpha,\sigma}$ is either of the $\vee$ or $\wedge$ type, and define:

$$W^\wedge \models \phi \text{ iff } \forall w \in W : w \models \phi; \quad (16a)$$

$$W^\vee \models \phi \text{ iff } \exists w \in W : w \models \phi. \quad (16b)$$

By these definitions we obtain the following results for prohibitions and permissions:

$$F(\alpha + \beta) \equiv F\alpha \wedge F\beta \quad (17a)$$

$$P(\alpha + \beta) \equiv P\alpha \vee P\beta \quad (17b)$$

$$F(\alpha \oplus \beta) \equiv F\alpha \vee F\beta \quad (17c)$$

$$P(\alpha \oplus \beta) \equiv P\alpha \wedge P\beta \quad (17d)$$

which is precisely what we want to have! For example, (17c) is proved as follows: Let σ be some arbitrary state (world). Then:

$$\begin{aligned} \sigma \models F(\alpha \oplus \beta) &\text{ iff} \\ \sigma \models [\alpha \oplus \beta]V &\text{ iff} \\ \llbracket \alpha \oplus \beta \rrbracket(\sigma) \models V &\text{ iff} \\ \{\llbracket \alpha \rrbracket(\sigma), \llbracket \beta \rrbracket(\sigma)\}^V \models V &\text{ iff} \\ \llbracket \alpha \rrbracket(\sigma) \models V \text{ or } \llbracket \beta \rrbracket(\sigma) \models V &\text{ iff} \\ \sigma \models [\alpha]V \text{ or } \sigma \models [\beta]V &\text{ iff} \\ \sigma \models F\alpha \text{ or } \sigma \models F\beta &\text{ iff} \\ \sigma \models F\alpha \vee F\beta. \end{aligned}$$

Hence $\models F(\alpha \oplus \beta) \equiv F\alpha \vee F\beta$. Now (17d) follows immediately from (17c) and (9).

4. Ross's Paradox

In this section we shall examine what the consequences of our approach to free choice permissions are for Ross's Paradox:

$$O\alpha \supset O(\alpha \text{ OR } \beta), \quad (18)$$

where O stands for "it is obligated to do ..."

This paradox is well-known in the deontic literature. Although (18) is a theorem in nearly every logical deontic system, it is counterintuitive in examples such as

(19) "You are obliged to mail the letter" implies "You are obliged to mail the letter or burn it".

As the similarity of (19) and (2) shows already, Ross's Paradox is closely related to the problem with free choice permissions. In fact, what is counterintuitive in (19) is the apparently natural reading of the conclusion, in which the 'OR' seems to be a 'free' choice! The formula (18), however, is derived in a system where 'OR' has the interpretation of an 'imposed' choice. For instance in (19), it is certainly perfectly sound to state that if it is obliged to mail the letter, it is also obliged to mail the letter or burn it, where the latter choice is determined by the external environment of the actor. So, in the notation/terminology of the previous section (18) becomes:

$$O\alpha \supset O(\alpha + \beta), \quad (20)$$

which is a correct assertion. However, the natural reading of (19) corresponds to

$$O\alpha \supset O(\alpha \oplus \beta), \quad (21)$$

which must be held to be false!

How can we see this formally? What are the logical properties of $O(\alpha \oplus \beta)$?

At first sight we might try to follow the same way in which we have treated $O(\alpha + \beta)$ and $O\alpha$ in general in [Me]: some action is obligatory if it is forbidden not to do it: (We use the notation $\bar{\alpha}$ for not- α .†)

$$O\alpha \equiv F\bar{\alpha}. \quad (22)$$

In the system PD_eL in [Me] this amounts to

$$O\alpha \equiv [\bar{\alpha}]V, \quad (23)$$

the doing of not- α leads to liability to sanction. In the case of $\alpha = \alpha_1 + \alpha_2$

this amounts to

$$O(\alpha_1 + \alpha_2) \equiv F(\overline{\alpha_1 + \alpha_2}) \equiv [\overline{\alpha_1 + \alpha_2}]V. \quad (24)$$

If we would follow the same lines for $\alpha = \alpha_1 \oplus \alpha_2$, we would obtain

$$O(\alpha_1 \oplus \alpha_2) \equiv F(\overline{\alpha_1 \oplus \alpha_2}) \equiv [\overline{\alpha_1 \oplus \alpha_2}]V. \quad (25)$$

However, although it is conceptually easy to find an action expression that is equivalent to $\overline{\alpha_1 + \alpha_2}$ - viz. $\overline{\alpha_1} \& \overline{\alpha_2}$, the simultaneous performance of not- α_1 and not- α_2 , cf. [Me] - it is not clear at all what to make of $\overline{\alpha_1 \oplus \alpha_2}$. And this is crucial in order to give further properties of $O(\alpha_1 \oplus \alpha_2)$. For instance, in [Me] we could derive in such a way:

$$O\alpha_1 \equiv [\overline{\alpha_1}]V \supset [\overline{\alpha_1} \& \overline{\alpha_2}]V \equiv O(\alpha_1 + \alpha_2). \quad (26)$$

For $O(\alpha_1 \oplus \alpha_2)$ this route seems closed. But what we can do is transforming $O(\alpha_1 \oplus \alpha_2)$ into an expression that we *can* deal with. For this the following seems an obvious choice:

$$O(\alpha_1 \oplus \alpha_2) \equiv O(\alpha_1 + \alpha_2) \wedge P(\alpha_1 \oplus \alpha_2), \quad (27)$$

which expresses that the obligation to a free choice between α_1 and α_2 is the same as the obligation to the imposed choice between α_1 and α_2 together with the permission to choose freely between α_1 and α_2 .

This axiom amounts to the following property:

$$O(\alpha_1 \oplus \alpha_2) \equiv O(\alpha_1 + \alpha_2) \wedge P\alpha_1 \wedge P\alpha_2 \quad (28)$$

which has as a consequence:

$$O(\alpha_1 \oplus \alpha_2) \supset P\alpha_1 \wedge P\alpha_2 \quad (29)$$

i.e. the obligation to the free choice between α_1 and α_2 implies the permissibility of both α_1 and α_2 , which is intuitively sound and desirable.

By elaborating (28) some further we get the expression:

$$O(\alpha_1 \oplus \alpha_2) \equiv [\overline{\alpha_1} \& \overline{\alpha_2}]V \wedge \langle \alpha_1 \rangle \neg V \wedge \langle \alpha_2 \rangle \neg V \quad (30)$$

which states that if neither the action α_1 nor the action α_2 is performed, we get into states of liability to sanction; moreover, there is a way to perform α_1 and not to become liable to sanction, and the same holds for α_2 .

Furthermore, we can infer the following property:

$$O(\alpha_1 + \alpha_2) \wedge P(\alpha_1 \& \alpha_2) \supset O(\alpha_1 \oplus \alpha_2), \quad (31)$$

since $P(\alpha_1 \& \alpha_2) \supset (P\alpha_1 \wedge P\alpha_2) \equiv P(\alpha_1 \oplus \alpha_2)$.[‡] Concerning Ross's Paradox we note that (21) is *not* a theorem, since

$$\nVdash O\alpha \supset P(\alpha \oplus \beta). \quad (32)$$

Although this all seems very satisfactory, we are still faced with a modified version of Ross's Paradox:

$$O\alpha \wedge P\alpha \wedge P\beta \supset O(\alpha \oplus \beta). \quad (33)$$

This seems to be a paradox as strange and severe as the original paradox of Ross, witness the following example:

- (34) "You have to buy oranges",
 "It is permitted to buy oranges", and
 "It is permitted to buy bananas" imply
 "You have to buy oranges or bananas (and you have a free choice)".

This sounds very paradoxical indeed: it seems that you have a free choice of buying bananas instead of oranges, while you have the obligation to buy oranges!

On a closer examination of this paradox the reason why (33) is paradoxical appears to be that one would expect the antecedent $O\alpha \wedge P\alpha \wedge P\beta$ to be contradictory if $P\beta$ denotes a permission that would go against the obligation $O\alpha$ to do α , which is equivalent to the prohibition $F\bar{\alpha}$ or $\neg P\bar{\alpha}$, saying that it is not permitted to do not- α . If $\beta = \bar{\alpha}$, $O\alpha \wedge P\alpha \wedge P\beta (= \neg P\bar{\alpha} \wedge P\alpha \wedge P\bar{\alpha})$ is indeed contradictory. So in this case the paradox disappears, since now (33) is vacuously true because of a false antecedent. For example, consider the following situation which is a modified version of (34):

- (35) "You have to buy *only* oranges",

"It is permitted to buy *only* oranges", and

"It is permitted to buy bananas" imply

"You have to buy *only* oranges or bananas (and you have a free choice)".

However, in (34) $\beta \neq \bar{\alpha}$, since buying bananas is certainly not the same as not buying oranges: buying bananas does not exclude buying oranges. In fact, what is intuitively wrong regarding (33) in this case is that the consequent $O(\alpha \oplus \beta)$ is read as an obligation to do α or β , together with a *complete* freedom to choose between the two alternatives. This is simply not truly represented in (33), since from the premise $O\alpha \wedge P\alpha \wedge P\beta$ we may derive $\neg P\bar{\alpha} \wedge P\alpha \wedge P\beta$, so that in choosing β it must be incorporated that $\bar{\alpha}$ is not permitted. So in fact it is forbidden to do $\beta \& \bar{\alpha}$, since $\neg P\bar{\alpha}$, i.e. $F\bar{\alpha}$ implies $F(\beta \& \bar{\alpha})$. So there is *no* complete freedom of choosing α or β : if β is chosen it is imperative that $\beta \& \alpha$ is performed!

We can mend this by strengthening the axiom (27) for free choice obligation:

$$\begin{aligned} O(\alpha_1 \oplus \alpha_2) &\equiv O(\alpha_1 + \alpha_2) \wedge P((\alpha_1 \& \bar{\alpha}_2) \oplus (\bar{\alpha}_1 \& \alpha_2)) \\ &(\equiv O(\alpha_1 + \alpha_2) \wedge P(\alpha_1 \& \bar{\alpha}_2) \wedge P(\bar{\alpha}_1 \& \alpha_2)), \end{aligned} \quad (36)$$

which states that in order to have an obligation to do either α_1 or α_2 , where there is a completely free choice between the two, one must be permitted to do α_1 without α_2 , and vice versa.¶

Now indeed we have for arbitrary α and β that (33) does not hold any more, and, moreover, $O\alpha \wedge O(\alpha \oplus \beta) \supset \neg P\bar{\alpha} \wedge P(\alpha \& \bar{\beta}) \wedge P(\bar{\alpha} \& \beta) \supset \neg P\bar{\alpha} \wedge P(\bar{\alpha} \& \beta) \supset \neg P\bar{\alpha} \wedge P\bar{\alpha} \supset \text{false}$. So the obligation to do α and that of doing α or β having complete freedom to choose between these two are mutually inconsistent, as we would expect intuitively.

We *do* have again a modified version of (33), viz.

$$O\alpha \wedge P(\alpha \& \bar{\beta}) \wedge P(\bar{\alpha} \& \beta) \supset O(\alpha \oplus \beta), \quad (37)$$

that holds, but this is not a paradox any more, since the antecedent of (37) is contradictory:

$$O\alpha \wedge P(\alpha \& \bar{\beta}) \wedge P(\bar{\alpha} \& \beta) \supset \neg P\bar{\alpha} \wedge P(\bar{\alpha} \& \beta) \supset \neg P\bar{\alpha} \wedge P\bar{\alpha} \equiv \text{false}.$$

Finally in this section, we consider some consequences of axiom (36): first we look at the special case that $\alpha_2 = \bar{\alpha}_1$: $O(\alpha \oplus \bar{\alpha}) \equiv O(\alpha + \bar{\alpha}) \wedge P((\alpha \& \bar{\alpha}) \oplus (\bar{\alpha} \& \alpha)) \equiv O(\alpha + \bar{\alpha}) \wedge P(\alpha \& \alpha) \wedge P(\bar{\alpha} \& \bar{\alpha}) \equiv O(\alpha + \bar{\alpha}) \wedge P\alpha \wedge P\bar{\alpha}$, which corresponds exactly with the old axiom (27), as it should in this case! Next we look at another trivial case: $O(\alpha \oplus \alpha) \equiv O(\alpha + \alpha) \wedge P((\alpha \& \bar{\alpha}) \oplus (\bar{\alpha} \& \alpha)) \equiv O(\alpha + \alpha) \wedge P(\alpha \& \bar{\alpha}) \wedge P(\bar{\alpha} \& \alpha) \equiv \text{false}$, since $P(\alpha \& \bar{\alpha}) \equiv P\text{fail} \equiv \langle \text{fail} \rangle \neg V \equiv \text{false}$, since **fail** has no successor states (cf. [Me]). This can be defended as there is apparently no freedom in choosing between α and α . Likewise, $O(\alpha \oplus (\alpha \& \beta)) \equiv O(\alpha + (\alpha \& \beta)) \wedge P(\alpha \& (\bar{\alpha} \& \beta)) \wedge P(\bar{\alpha} \& (\alpha \& \beta)) \equiv \text{false}$, since $\bar{\alpha} \& (\alpha \& \beta) = (\bar{\alpha} \& \alpha) \& \beta = \text{fail}$. Again this can be defended as the right outcome, since the choice between doing α and α , together with something (β) else, is not really a free one. On the other hand, if one does not like this one may consider relaxing (36) to

$$O(\alpha_1 \oplus \alpha_2) \equiv O(\alpha_1 + \alpha_2) \wedge (P((\alpha_1 \& \bar{\alpha}_2) \oplus (\bar{\alpha}_1 \& \alpha_2)) \vee P(\alpha_1 \& \alpha_2)), \quad (38)$$

which allows for the possibility to do α_1 and α_2 simultaneously. Now we obtain $O(\alpha \oplus \alpha) \equiv O(\alpha + \alpha) \wedge (P\text{fail} \wedge P(\bar{\alpha})) \vee P(\alpha \& \alpha) \equiv O\alpha \wedge (\text{false} \vee P\alpha) \equiv O\alpha \wedge P\alpha$, and $O(\alpha \oplus (\alpha \& \beta)) \equiv O(\alpha + (\alpha \& \beta)) \wedge ((P(\alpha \& \bar{\alpha} \& \beta) \wedge P(\bar{\alpha} \& \alpha \& \beta)) \vee P(\alpha \& \alpha \& \beta)) \equiv O(\alpha + (\alpha \& \beta)) \wedge P(\alpha \& \alpha \& \beta) \equiv O(\alpha + (\alpha \& \beta)) \wedge P(\alpha \& \beta)$, which is not contradictory. However, taking this approach introduces again a Ross-like paradox, since now also $O\alpha \wedge P(\alpha \& \beta) \supset O(\alpha \oplus \beta)$ becomes valid. So instead of adopting (38) we prefer to adhere to (36), along with its (defensible) consequences.

In conclusion of this section, we claim that we have clarified Ross's Paradox some further, using our view on free choice permissions.

5. Conclusion

The main point we wanted to stress in this note is that we can resolve the problem regarding free choice permissions by distinguishing two kinds of nondeterministic actions: internal and external choices. Free choice permissions now become permissions to perform internal ('free') choices, whereas 'imposed' choice permissions are permissions to do externally determined ('imposed') choices. It is clear that both free and imposed permissions may now be put into one logical system, which is an extended version of PD_eL in [Me]. This is needed in case one wants to reason with both notions, e.g. in the case of specifying deontic integrity constraints for data and knowledge bases in a legal or normative domain (cf. [WM]). One has only to be careful when giving axioms for obligation. We need to separate the case $O(\alpha_1 \oplus \alpha_2)$ from the rest of the cases for $O\alpha$, thus:

$$O\alpha \equiv F\bar{\alpha}, \text{ if } \alpha \neq \alpha_1 \oplus \alpha_2 \quad (39a)$$

$$O(\alpha_1 \oplus \alpha_2) \equiv O(\alpha_1 + \alpha_2) \wedge P((\alpha_1 \& \bar{\alpha}_2) \oplus (\bar{\alpha}_1 \& \alpha_2)) \quad (39b)$$

We also gave an analysis of Ross's Paradox based upon our approach to free choice permissions which we believe sheds some new light upon this already much discussed subject.

Acknowledgements: I would like to thank Frans Birrer, Frans Voorbraak and Roel Wieringa for discussions we had on the topic of this paper, and which influenced the presentation of the material in it.

Notes

*The notions of internal and external nondeterminism are well-known in the theory of computer science (cf. [Ho]), but there the internal choice cannot be influenced by the actor (cq. programmer) whereas the external one can. This has to do with a different point of view: in [Ho] the actor cannot influence the inside of a machine while he himself is seen as the external environment; in our case we consider the actor himself as the internal nucleus of an external environment, which he cannot influence a priori.

†The semantics of negated actions is not entirely trivial. It is treated in [Me] and [DM]. Without plunging into technicalities, one may say informally that $\bar{\alpha}$ is the performance of any action not incorporating the performance of α .

‡In PD_eL the following are validities with respect to '&' (cf. [Me]): $P(\alpha \& \beta) \supset (P\alpha \wedge P\beta)$, $(F\alpha \vee F\beta) \supset F(\alpha \& \beta)$, and $O(\alpha \& \beta) \equiv (O\alpha \wedge O\beta)$

¶At first sight, $O(\alpha_1 \oplus \alpha_2) \equiv O(\alpha_1 + \alpha_2) \wedge (P((\alpha_1 \& \bar{\alpha}_2) \oplus (\bar{\alpha}_1 \& \alpha_2)) \wedge P(\alpha_1 \& \alpha_2))$ would perhaps seem to be the most free choice. This would, however, be far too restrictive a requirement, for it may well be that α_1 and α_2 exclude each other, and then the last conjunct on the right-hand side and so also the whole right-hand side would become **false**, which is not desirable, of course.

References

- [Å] Åqvist, L.: 1984, Deontic Logic, in: D.M. Gabbay & F. Guenther (eds.), *Handbook of Philosophical Logic, Vol. II*, Reidel, Dordrecht/Boston
- [Ca] Castañeda, H.-N.: 1981, The Paradoxes of Deontic Logic, in: R. Hilpinen (ed.), *New Studies in Deontic Logic*, Reidel, Dordrecht/Boston
- [DM] Dignum, F.P.M. and Meyer, J.-J.Ch.: 1990, Negations of Transactions and Their Use in the Specification of Dynamic and Deontic Integrity Constraints, in: *Semantics for Concurrency, Leicester 1990* (M.Z. Kwiatkowska, M.W. Shields & R.M. Thomas, eds.), Springer-Verlag, London/Berlin, 61-80
- [Har] Harel, D.: 1984, Dynamic Logic, in: D.M. Gabbay and F. Guenther (eds.), *Handbook of Philosophical Logic, Vol. II*, Reidel, Dordrecht/Boston
- [Hil] Hilpinen, R.: 1981, Conditionals and Possible Worlds, in: G. Fløistad (ed.), *Contemporary Philosophy, A New Survey, Vol. 1*, 99-335
- [Ho] Hoare, C.A.R.: 1985, *Communicating Sequential Processes*, Prentice-Hall International, Englewood Cliffs, New Jersey
- [Ka] Kamp, H.: 1973-74, Free Choice Permission, *Aristotelian Society Proceedings N.S.* 74, 57-74

- [Me] Meyer, J.-J.Ch.: 1988, A Different Approach to Deontic Logic: Deontic Logic Viewed as a Variant of Dynamic Logic, *Notre Dame Journal of Formal Logic* , 29(1) , 109-136
- [Ro] Ross, A.: 1941, Imperatives and Logic, *Theoria* 7 , 53-71
- [vW] von Wright, G.H.: 1951, *An Essay in Modal Logic* , North-Holland Publ. Co., Amsterdam
- [WM] Wieringa, R.J. and Meyer, J.-J.Ch.: 1991, Actor-Oriented Specification of Deontic Integrity Constraints, in Proceedings MFDBS '91 (B. Thalheim, J. Demetrovics and H.-D. Gerhardt, eds.), Rostock, Lect. Notes in Comp. Sc. 495, Springer, 89-103

Partial Objects and DRT

W.P.M. Meyer Viol
OTS University of Utrecht

Abstract

In this article the DRT rule for introduction of markers in a DRS is interpreted as the logical rule of *existential instantiation*. Consequently, markers are taken to be constants witnessing existential formulas, instead of existentially quantified variables. Models are introduced, where these witnesses denote *partial objects* and various proof systems are studied. Finally, a dynamic account of sequential sentence processing is proposed, where sentences are interpreted as *update* functions with sets of discourse markers as input and output.

Introduction

In traditional *Discourse Representation Theory* (DRT) the occurrence of the indefinite noun phrase 'a man' in the sentence "A man walks in" gives rise to a twofold action: (1) a *marker* ' x ' is introduced in the discourse representation structure (DRS) for future reference to this man, (2) a *condition* " $man(x)$ " is added to the DRS to the effect that this marker stands for a man. Semantically the marker is interpreted as an *existentially quantified variable*: the DRS is true with respect to a particular model if *there is* a variable assignment, mapping x to an element in the domain of the model, such that the predicate *man* holds for this element.

Now, a logician cannot fail to notice the strong resemblance between the DRT rule for the introduction of a marker for indefinite noun phrases and the logical rule of *existential instantiation*.

$$\frac{\exists x \text{ man}(x)}{\text{man}(a)}$$

and note that the introduction of the marker and the addition of a condition constitute a *single* logical action. Under this interpretation a marker is no longer seen as an existentially quantified variable, but as a *witness* for the corresponding existential sentence.

When we take markers to be variables, what holds for this marker is no more than that it stands for a man who walks in. It is true that any variable assignment will map the marker to an 'complete' man, with a lot more properties than just these two, but this surplus disappears when we take the properties of this marker to be the ones that are invariant under *all* possible variable assignments verifying the DRS. On the other hand, when we interpret markers as witnesses for existential sentences, we see them logically as *constants*, and, unlike variables, constants denote. What element of the domain can we take these constants to denote? Every individual we choose will have more properties than are accounted for by the text. In fact, the totality of his properties will constitute a *maximally consistent set*. Consequently, the interpretation of marker introduction as *existential instantiation*, forces us to

consider the denotation of markers as *partial(ly specified) objects*.

This is the connection between DRT and partial objects that will be explored on these pages. In the first section we will introduce the models containing a domain of partial objects. In these models, when we have a partial object a specified by the predicate 'man', then $\exists x \text{man}(x) \rightarrow \chi(a)$ will hold if and only if $\forall x(\text{man}(x) \rightarrow \chi(x))$ holds: the object a has all and only all properties that are logical consequences of it being a man. Notice that this condition is purely motivated by a natural interpretation of partial objects. However, what we have here is one of the 'donkey equivalences' [Kamp [10]]. The second section will deal with a classical proof system, adequate for these models and will consider some extensions adapted to exploit the richer models that have been developed. The third section will discuss DRT proper and will deal with a purely dynamic and partial interpretation in which DRS's are formalized as finite sets of partial objects, and sentences — natural language or first-order — as update functions, mapping these sets onto new sets.

1 Language, Models and Truth

Language

The language we will use to describe the structures of partial objects will be a first-order language $\mathcal{L} = \langle \mathcal{C}, \mathcal{A}, \mathcal{P} \rangle$ where $\mathcal{C}$ and $\mathcal{A}$ are denumerable sets of constants, denoting standard individuals and partial objects respectively and $\mathcal{P}$ is a set of relation symbols.

We will let $\phi, \chi, \xi \dots$ range over $\mathcal{L}$ -formulas, and $\Delta, \Gamma \dots$ range over subsets of $\mathcal{L}$ -formulas. With $\mathcal{L}(x)$ we will denote the set of all $\mathcal{L}$ -formulas with x as the only free variable. For any formula ϕ , $\mathcal{A}(\phi)$ will be the set of constants from $\mathcal{A}$ occurring in ϕ .

Models

Our models for partial objects are slight variations of K.Fine's *Arbitrary Object Models*. Arbitrary Object Models contain two disjoint domains: a domain M of standard individuals and a domain A of arbitrary (partial) objects. Each arbitrary object in A is associated with a set of elements from M , the set of its *instances* or *completions*. This association is formalized by a set V of *finite functions* from A into M : for $v \in V$, the element $v(a)$ of M is taken to be an *instance* or *completion* of a .

Definition 1 $\mathcal{M}^* = \langle M, A, V, <, I \rangle$ is an *Arbitrary Object Model*, (AO-Model) if

- M and A are disjoint sets. M the *standard* domain consists of standard individuals, A , the *generic* domain, consist of arbitrary objects.
- $V \subseteq M^A$ is a set of finite functions from A in M .

- $< \subseteq A \times A$, the *dependency relation*, is a strict partial order that is conversely well-founded
- I is an interpretation function, satisfying: for $\bar{c} \in \mathcal{C}$: $I(\bar{c}) \in M$ and for $\bar{a} \in \mathcal{A}$: $I(\bar{a}) \in A$.

So AO-models essentially consist of two components

- a standard first-order $\mathcal{L}$ -model $\mathcal{M} = \langle M, I_0 \rangle$, where $I_0 = I \upharpoonright \mathcal{C}$
- a *generic extension* $\langle A, <, V \rangle$ over $\mathcal{M}$. The *generic domain* A consists of arbitrary objects, connected to M by the assignments in V . The next section will discuss the dependency relation.

The set V of assignments of instances to arbitrary objects has to satisfy the following closure properties:

Partial Extendibility: If $v \in V$, then there is a $v' \in V$ such that $v \subseteq v'$ and $\text{dom}(v)$ is closed.

Restriction: if $B \subseteq \text{dom}(v)$, then $v \upharpoonright B \in V$

Piecing: If v, v' are assignments with closed domains and $v \cup v'$ is a function, then $v \cup v' \in V$.

Where $B \subseteq A$ is closed if, whenever $a \in B$ and $a < b$, then $b \in B$

Truth

The idea behind the notion of truth regarding sentence containing constants denoting arbitrary objects is that these objects have a certain property if all their completions have that property. This is the principle of *generic attribution*.

Given $\mathcal{M}^* = \langle M, A, I, <, V \rangle$, we will assume to have constants $\bar{m}$ for all elements of M . The variable assignments g will always be mappings from VAR , the set of individual variables of $\mathcal{L}$, to M ; so formulas without constants from $\mathcal{A}$ are interpreted standardly in $\mathcal{M} = \langle M, I \upharpoonright \mathcal{C} \rangle$, the first-order structure underlying $\mathcal{M}^*$. Finally, $v(\phi(\bar{a}_1, \dots, \bar{a}_n))$ will denote the result of the simultaneous substitution in ϕ of $\bar{m}_i$ for $\bar{a}_i$, where, for $v \in V$, $1 \leq i \leq n$, $v(I(\bar{a}_i)) = m_i$.

Definition 2 Truth Definition.

Let ϕ be an $\mathcal{L}$ -formula such that $\mathcal{A}(\phi) = \{\bar{a}_1, \dots, \bar{a}_n\}$ and let $V_{\mathcal{A}(\phi)} = \{v \in V \mid \mathcal{A}(\phi) \subseteq \text{dom}(v)\}$, then

$$\mathcal{M}^* \models \phi [g] \text{ iff } \forall v \in V_{\mathcal{A}(\phi)}: \mathcal{M} \models v(\phi) [g]$$

Syntactically, the $\mathcal{A}$ -terms are constants of the language, but semantically they resemble variables. Suppose we define $\mathcal{M} \models \forall x \phi(x)$ to mean: for all variable assignments $g \in G$ $\mathcal{M} \models \phi(x) [g]$, where G is the set of all variable assignments. Then

we have the classical definition for universal quantification. Arbitrary object semantics arises when, for specific variables, we use the above definition with respect to a proper subset of G . In this way, a constant $a \in A$ can be interpreted as a variable in the domain of only a subset of variable assignments. And, as we will see, in standard AO-models this restriction will take the form of *restricted universal quantification*. According to the truth definition we have

$$\mathcal{M}^* \models \phi \wedge \psi \iff \mathcal{M}^* \models \phi \ \& \ \mathcal{M}^* \models \psi$$

$$\mathcal{M}^* \models \phi \ \& \ \mathcal{M}^* \models \phi \rightarrow \psi \Rightarrow \mathcal{M}^* \models \psi$$

So the set of properties of an $a \in A$ is a consistent, deductively closed, set. But we do not have

$$\mathcal{M}^* \models \neg\phi \iff \mathcal{M}^* \not\models \phi$$

$$\mathcal{M}^* \models \phi \vee \psi \iff \mathcal{M}^* \models \phi \text{ or } \mathcal{M}^* \models \psi$$

So the set of properties of an object is not maximally consistent.

When we turn to the notion of a *valid inference* the variable like nature of arbitrary objects becomes even more prominent.

Definition 3 Validity

For T a set of $\mathcal{L}$ -sentences and ϕ an $\mathcal{L}$ -sentence,

- the inference $T \models \phi$ is *truth-to-truth valid* in model $\mathcal{M}^*$ if, whenever $\mathcal{M}^* \models T$ then $\mathcal{M}^* \models \phi$
- the inference $T \models \phi$ is *case-to-case valid* in model $\mathcal{M}^*$ if, for all $v \in V$ defined over the constants in T and ϕ , whenever $\mathcal{M} \models v(T)$ then $\mathcal{M} \models v(\phi)$.

Truth-to-truth validity deals with inferences leading from properties of arbitrary objects to other properties. Here the A -constants behave purely as constants. This can be compared to an inference of the form ' $\forall x T(x)/\forall x \phi(x)$ '. This interpretation does not allow us to conclude the universal validity of $\phi \rightarrow \psi$ from the validity of the inference ϕ/ψ (we can not infer $\forall x(T(x) \rightarrow \phi(x))$). Case-to-case validity of inferences, on the other hand, can be compared to 'for all $g \in G : T(x) [g]/\phi(x) [g]$ '. This allows us to *conditionalize*; for finite T : if the inference T/ϕ is case-to case valid, then the sentence $T \rightarrow \phi$ is universally valid. In case-to-case validity, the *scope* of the (hidden) quantifier in the premiss includes the conclusion. The case-to-case interpretation makes the constants act like *free variables*, where premiss and conclusion are evaluated with respect to the same variable assignment.

1.1 The Dependency Relation

Markers are introduced at certain points in a discourse, and a new marker may *depend* on previously introduced ones. A representation of the text "A man walks in. His mother follows him as usual" must incorporate the fact that the marker

for "His mother" depends on the marker for "A man": If an individual m is an instance of the marker for the mother, then m must be the mother of the instance chosen for the marker for "A man". Similarly, the sentence "A farmer who owns a donkey, never beats it" would lead to the introduction of an 'independent' marker for "A farmer" and a marker for "a donkey" that *depends* on the first marker. These dependencies between the objects are dynamically created in the course of a text and are sensitive to the *order* in which the sentences are presented. For instance, the text "A woman walks in. As usual she is preceded by her son" will give rise to other dependencies than the ones generated by our first example.

In AO-models, the *dependency relation* formalizes the 'static projection' of these dependencies. This relation is a strict partial order which is conversely well-founded: dependencies are irreflexive and transitive, and there are no infinite dependency chains.

Formally, we will interpret the introduction of a discourse marker as the application of the rule of *existential instantiation*. Under this interpretation, markers that have previously been introduced may occur as *parameters* in the existential formula.

$$\frac{\exists x \phi(x, \bar{a}_1 \dots \bar{a}_n)}{\phi(\bar{a}, \bar{a}_1 \dots \bar{a}_n)}$$

If $\mathcal{M}^* \models \exists x \phi(x, \bar{a}_1 \dots \bar{a}_n)$ then the values the *witness* $\bar{a}$ can take depend on the values already assigned to the parameters $a_1, \dots, a_n$, so a depends on $\bar{a}_1 \dots \bar{a}_n$, in other words, $a < a_1, \dots, a < a_n$.

In order to get a general perspective on this interpretation of the dependency relation we will define set $\mathcal{A}_L$ of *indexed constants* the elements of which will exhibit dependencies *explicitly*: whenever a witness is introduced for an existential formula $\exists x \phi(x)$, it will be indexed by the formula that led to its introduction. So the occurrence of the indefinite noun phrase "a man" leads to the introduction of a marker $a_{man(x)}$

$$\frac{\exists x \text{ man}(x)}{\text{man}(\bar{a}_{man(x)})}$$

Or, in general,

$$\frac{\exists x \phi(x, \bar{a}_1 \dots \bar{a}_n)}{\phi(\bar{a}_{\phi(x, \bar{a}_1 \dots \bar{a}_n)}, \bar{a}_1 \dots \bar{a}_n)}$$

Note that the constants from $\exists x \phi(x)$, its parameters, will occur in the index $\phi(x)$ of $\bar{a}_{\phi(x)}$. Consequently the constant $\bar{a}_{\phi(x)}$ has an index in which other indexed constants may occur, the indices of which may again contain indexed constants. This motivates our recursive definition of a set of indexed constants.

Definition 4 Indexed Constants

The set $\mathcal{A}_L$ of *indexed constants* is given by the following recursive definition

- $\mathcal{A}_L^0 = \emptyset$, $\mathcal{L}^0 = \langle \mathcal{C}, \mathcal{A}_L^0, \mathcal{P} \rangle$
- $\mathcal{A}_L^{n+1} = \{ \langle \bar{a}, \phi(x) \rangle \mid \bar{a} \in \mathcal{A}, \phi(x) \in \mathcal{L}^n(x) \}$
where $\mathcal{L}^n(x)$ is the set of all $\mathcal{L}^n$ -formulas in which only variable x is free

- $\mathcal{L}^{n+1} = \mathcal{L}^n \cup \mathcal{A}_{\mathcal{L}}^n$
- $\mathcal{A}_{\mathcal{L}} = \bigcup_{n < \omega} \mathcal{A}_{\mathcal{L}}^n$ and $\mathcal{L} = \bigcup_{n < \omega} \mathcal{L}^n$

Note that $\mathcal{L} = \mathcal{L}^0 \cup \mathcal{A}_{\mathcal{L}}$. The index $\phi(x)$ of a constant $\bar{a}_\phi = \langle \bar{a}, \phi(x) \rangle \in \mathcal{A}_{\mathcal{L}}$ will be called its *identifier*. An identifier may itself contain constants from $\mathcal{A}_{\mathcal{L}}$; this induces a natural ordering on the constants.

Definition 5 Ordering of Constants

For $\bar{a}_\phi, \bar{a}_\psi \in \mathcal{A}_{\mathcal{L}}$

- $\bar{a}_\phi \ll \bar{a}_\psi \iff \bar{a}_\psi \in \mathcal{A}(\phi)$
- $\bar{a}_\phi \prec \bar{a}_\psi \iff \exists a_1, \dots, a_n$ such that $\bar{a}_\phi = a_1$, $\bar{a}_\psi = a_n$ and for $0 \leq i < n$: $a_i \ll a_{i+1}$
- $|\bar{a}_\psi| = \{\bar{a}_\phi \in \mathcal{A}_{\mathcal{L}} \mid \bar{a}_\phi \prec \bar{a}_\psi\}$. For ϕ an $\mathcal{L}$ -formula, $|\phi| = \bigcup_{\bar{a}_\psi \in \mathcal{A}(\phi)} |\bar{a}_\psi|$.

$\langle \mathcal{A}_{\mathcal{L}}, \prec \rangle$ is a strict partial order that is finitely branching and conversely well-founded. Consequently, $|\bar{a}_\phi|$ is a finite set for every $\bar{a}_\phi$.

We will now take these indexed constants to refer to *partial objects*. Partial objects will be arbitrary objects, indexed by formulas from $\mathcal{L}(x)$

Definition 6 Partial Object Models

A *Partial Object Model* (PO-Model) is an AO-model $\mathcal{M}^* = \langle M, \mathcal{A}_{\mathcal{L}}, V, <, I \rangle$ where

- $\mathcal{A}_{\mathcal{L}} = A \times \mathcal{L}(x)$ with A a non-empty set disjoint from M
- $I(\bar{a}_\phi) = a_\phi$ for some $a \in A$
- $\mathcal{M}^* \models \exists x \phi(x) \rightarrow \phi(\bar{a}_\phi)$
- $a_\phi < a_\psi \iff \bar{a}_\phi \prec \bar{a}_\psi$

Standard Models

We will call an PO-model $\mathcal{M}^* = \langle M, A, V, <, I \rangle$ *standard* if, $\mathcal{M}^* \models \psi(\bar{a}_\phi)$ implies $\mathcal{M}^* \models \forall x(\phi(x) \rightarrow \psi[x/\bar{a}_\phi])$ for all ψ and $\bar{a}_\phi$. In a standard model,

- if $\mathcal{M} \models v(\phi(\bar{m}))$, then $\exists v' : v \subseteq v'$ and $v'(a_\phi) = m$
- if $\mathcal{M} \not\models v(\exists \phi(x))$, then $\forall m \in M \exists v' \in V : v \subseteq v' \ \& \ v'(a_\phi) = m$.

Standard models are special in the following way. Let $V(a_\phi) = \{v(a_\phi) \mid \mathcal{M} \models v(\phi(\bar{a}_\phi))\}$ then for standard model $\mathcal{M}^*$ we have

- $V(a_{\phi \wedge \psi}) = V(a_\phi) \cap V(a_\psi)$
- $V(a_{\phi \vee \psi}) = V(a_\phi) \cup V(a_\psi)$
- $\overline{V(a_\phi)} = V(a_{\neg \phi})$

For PO-models in general this need not hold. For instance, there is nothing against a PO-Model where $V(a_\phi) \cap V(a_{\phi \wedge \phi}) = \emptyset$.

Marker Models

Now this purely syntactic relation on the set of partial objects is not completely satisfactory for the dynamics of marker introduction. For instance, the sentence $\exists x \phi(x, \bar{a}_\psi) \wedge \neg \exists x \psi(x)$ has PO-models. In such a model $\phi(\bar{a}_\phi)$ will hold and $\bar{a}_\phi \ll \bar{a}_\psi$, but $\psi(\bar{a}_\psi)$ will be falsified: the marker a_ϕ depends on a marker *that cannot have been introduced*.

In order to render sentences like the above inconsistent, we will introduce so-called *foundation formulas*. To understand these formulas, note that each constant $\bar{a}_\phi$ determines a finite tree structure in $\mathcal{A}_\mathcal{L}$ with nodes $|\bar{a}_\phi| \cup \{\bar{a}_\phi\}$. With each node $\bar{a}_\psi$ in this structure we can associate the existential formula $\exists x \psi(x)$ it witnesses. Consequently, with a_ϕ we can associate a finite tree of existential formulas. We will call an object a_ϕ *representing* in $\mathcal{M}^*$ with respect to an assignment v if and only if v verifies all the existential formulas in this tree.

Definition 7 Foundation Formulas

The *foundation formula* $F(a_\phi)$ for object a_ϕ is defined by

- For $\lambda(x)$ a literal in $\mathcal{L}(x)$, $F(a_\lambda) = \begin{cases} \top & \text{if } |a_\lambda| = \emptyset \\ \bigwedge \{\psi(\bar{a}_\psi) \mid \bar{a}_\lambda \ll \bar{a}_\psi\} & \text{otherwise} \end{cases}$
- $F(a_{\phi \wedge \psi}) = F(a_{\phi \vee \psi}) = F(a_\phi) \wedge F(a_\psi)$, $F(a_{\exists x \phi(x)}) = F(a_{\forall x \phi(x)}) = F(a_{\phi(x)})$.

Definition 8 Representational Mappings

Let $\mathcal{M}^*$ be an PO-model. An assignment $v \in V$ is a *representational map* if

$$a_\phi \in \text{dom}(v) \text{ only if } \mathcal{M} \models v(F(a_\phi))$$

This definition entails that v is defined over all objects denoted in $F(a_\phi)$ if a_ϕ is in the domain of v ; consequently the foundation formulas for all these constants also hold under v . So a representational map has a closed domain.

Let V^+ be the set of all representational mappings of $\mathcal{M}^*$.

Definition 9 A Marker Model (M-Model) $\langle \mathcal{M}^*, V^+ \rangle$ based on PO-model $\mathcal{M}^* = \langle M, \mathcal{A}_\mathcal{L}, V, <, I \rangle$ is $\mathcal{M}^*$ with V replaced by the set V^+ of all representational mappings in V , and $\mathcal{A}_\mathcal{L}$ replaced by $A^+ = \{a \in \mathcal{A}_\mathcal{L} \mid \exists v \in V^+ : a \in \text{dom}(v)\}$.

On both $\mathcal{M}^*$ and $\langle \mathcal{M}^*, V^+ \rangle$ the rule $\psi(\bar{a}_\phi) / \exists x \psi(x)$ is valid, but the stronger rule $\psi(\bar{a}_\phi) / \exists x (\psi(x) \wedge \phi(x))$ holds only on the latter.

2 Proof Systems

2.1 Proof System for PO-Models

The proof system belonging to partial object models consists of a standard system of rules of natural deduction for the boolean connectives extended by the following *instantial rules*

Instantial Rules

$$\begin{array}{ll}
 (UI) \quad \frac{\forall x\phi(x)}{\phi(\bar{t})} & (EI) \quad \frac{\exists x\phi(x)}{\phi(\bar{a}_\phi)} \\
 (UG) \quad \frac{\phi(\bar{t})}{\forall x\phi(x)} & (EG) \quad \frac{\phi(\bar{t})}{\exists x\phi(x)}
 \end{array}$$

Here, $\bar{a}_\phi \in \mathcal{A}$ and $\bar{t}$ is any $\mathcal{L}$ -term free for x in $\phi(x)$. Notice that only in EI the instancial term is indexed by ϕ ; in the constant $\bar{a}_\phi$ introduced by EI, a is uniquely associated with $\phi(x)$ (modulo logical equivalence). So if we have $\bar{a}_\phi$ and $\bar{b}_\phi$, where $\bar{a} \neq \bar{b}$, at most one can be the result of an application of EI.

The absence of an index in the terms of rules EG, UG, and UI reflects the fact that any term can occur in a valid application of these rules.

To be *sound*, a deduction D in the system has to satisfy the following restrictions (adapted from Copi [2])

Definition 10 Derivations

A derivation D is correct if and only if it satisfies:

WF Weak Flagging: no term a_ϕ that results from an application of EI in D, is also an UG instancial term.

IND Independence: in any application of UG, no A-letter occurring in either the conclusion or the suppositions to the inference can be identical to or depend upon the instancial term.

Weak Flagging speaks for itself, it prevents the drawing of universal conclusions from existential premisses. *Independence* extends the usual proviso for UG — that $\bar{t}$ does not occur in any of the undischarged suppositions on which $\phi(\bar{t})$ is based — to cover all constants on which $\bar{t}$ depends: it may be the case that $\bar{t}$ does not occur in the suppositions, but that we have a supposition $\psi(\bar{a}_\phi)$ where $\bar{a}_\phi < \bar{t}$. Independence rules out this case as a foundation for the application of UG. Given these restrictions, ϕ is derivable from premiss set T in our proof system if and only if ϕ is derivable from T in any standard axiomatization of first-order logic. Notice that only the *identity* of the constant enters in the formulation of the restrictions on deductions; consequently, as long as we respect these restrictions, it does not matter whether the constants involved are elements of $\mathcal{A}$ or $\mathcal{C}$. The difference between these constants only shows up in the semantics.

Definition 11 For T a set of L -sentences and ϕ a L -sentence, $T \vdash \phi$ holds in case there is a derivation ϕ with premiss set T , satisfying the Weak Flagging and Independence.

Without proof we state the following Lemma's

Lemma 1 Soundness

For T and ϕ free from $\mathcal{A}$ -constants, if $T \vdash \phi$ then T/ϕ is case-to case valid.

Lemma 2 Completeness

If T/ϕ is case-to-case valid, then $T \vdash \phi$.

Notice that EI only forces the *value range* of $\bar{a}_\phi$ to be a subset of $\{m \in M \mid \mathcal{M}^* \models \phi(\bar{m})\}$. It does not require the range to be this whole set. Consequently, beyond the fact that the ranges of $\bar{a}_\phi$ and $\bar{a}_{\phi \wedge \psi}$ are both subsets of $\{m \in M \mid \mathcal{M}^* \models \phi(\bar{m})\}$, there need be no more restrictions.

Standard Models

Not all PO-models will be standard in the sense that every m satisfying $\phi(x)$ in $\mathcal{M}$ will be the value of some v on a_ϕ . Addition of the following rule to the basic proof system will result in a system sound and complete for all standard PO-models:

$$\frac{\phi(\bar{a}_{\psi(x)})}{\forall x \exists y_1 \dots \exists y_n (\psi(x) \rightarrow \phi[x/\bar{a}_\psi, y_1/\bar{b}_1 \dots y_n/\bar{b}_n])}$$

Where $\mathcal{A}(\phi) = \{\bar{b}_1, \dots, \bar{b}_n\} \cup \{\bar{a}_\psi\}$. Derivability with this additional rule will be denoted by $\vdash_S$.

2.2 Proof System for M-Models

An obvious extension to the classical first-order system of natural deduction gives a sound and complete set of rules for the class of PO-models with only *representational* mappings. First we replace EI by *Founded Existential Instantiation* (FEI)

$$\frac{\exists x \phi(x) \quad F(a_\phi)}{\phi(\bar{a}_\phi)}$$

Secondly we add the *foundation rule* (FR)

$$\frac{\phi}{\phi \wedge F(\phi)}$$

where $F(\phi)$ is defined by

$$F(\phi) = \begin{cases} \top & \text{if } \mathcal{A}(\phi) = \emptyset \\ \bigwedge \{F(a_\psi) \mid \bar{a}_\psi \in \mathcal{A}(\phi)\} & \text{otherwise} \end{cases}$$

Notice that for atomic ϕ , $F(\phi)$ is a conjunction of $\psi(\bar{a}_\psi)$'s where $\bar{a}_\xi \ll \bar{a}_\psi$ for some $\bar{a}_\xi \in \mathcal{A}(\phi)$; $\xi(\bar{a}_\xi)$ itself is not a conjunct in $F(\phi)$. Notice furthermore that the foundation formula of a disjunction reduces to the *conjunction* of the foundation formulas of the disjuncts (we do not want $\phi(\text{he}) \vee \psi(\text{she})$ to mean that only one of 'he' and 'she' need be founded).

The foundation rule expresses the fact that all constants in ϕ are founded, i.e.

depend on markers that have actually been introduced. Iteration of the foundation rule takes care of the fact that $\psi(\bar{a}_\psi)$ is derivable, for every $\bar{a}_\psi \in |\bar{a}_\xi|$ for any derivable ϕ and $\bar{a}_\xi \in \mathcal{A}(\phi)$. The remainder of the deduction system is standard, except that we have to take care that in rules where we introduce (sub)formulas “out of the blue”, *extendibility* of representational mappings over these formulas is guaranteed. This merely involves modifying the rule for disjunction introduction to (FVI)

$$\frac{\phi \quad F(\psi)}{\phi \vee \psi}$$

Derivability according to the system FEI, FR, and FVI will be denoted by $\vdash_M$. Completeness of this set of rules with respect to the class of M-models follows easily when we interpret the $\mathcal{A}_L$ -constants as individual constants. Because of the foundation rule any set of sentences classically implying $\neg\psi(\bar{a}_\psi)$ for one of the constants in its foundation, is classically inconsistent, and has no first-order model. So any $\vdash_M$ -consistent set of $\mathcal{L}$ -sentences has a first order model. Note that for $T \cup \{\psi\}$ a set of $\mathcal{L}^0$ -sentences (no $\mathcal{A}_L$ -constants) The rule of founded existential instantiation reduces to existential instantiation simpliciter. It can be shown that, in that case, $T \vdash_M \psi$ if and only if ψ is derivable from T in some complete axiomatization of first-order logic. Notice that in the course of a derivation, starting from the empty set, the ‘domain of constants’ is extended by every application of (F)EI, and diminished by an application of EG. Simultaneously with an increasing domain, we get an increasing dependency relation.

2.3 Extended Proof Systems

In the basic proof system the identifiers of the objects are structurally meaningless. The proof of soundness of the deductive system only mentions the *identity* of the constants. As yet, we have no *relations* between partial objects and even, for instance, the objects a_ϕ and $a_{\phi \wedge \psi}$, or $a_{\phi \wedge \psi}$ and $a_{\psi \wedge \phi}$, may lack common instances. This implies that the general system is sensitive to all structural differences between the identifiers, and we can create a *representational system* by formulating rules that relate objects in terms of the identifiers. To bring this into familiar territory we will introduce a relation of ‘local consequence’ $\models \subseteq \mathcal{L} \times \mathcal{L}$ between sentences in the *same* constants, where will say that $\phi(a) \models \psi(a)$ holds in model $\mathcal{M}^*$ if $\mathcal{M}^* \models \psi(\bar{a}_\phi)$. To take a classic example from non-monotonic logic. Let $P(x)$ stand for the property of being a penguin, $B(x)$ the property of being a bird, and $F(x)$ the property of flying, then we can have $\mathcal{M}^* \models B(a_P) \wedge F(a_B) \wedge \neg F(a_P)$. This does not constitute an inconsistency, it only means that the value ranges of a_P and a_B will be disjoint. So in this set-up inconsistency is not the problem — we do not have to get rid of certain logical principles — the object, on the contrary, is to add principles in order to *relate* objects. So

$$\frac{\psi(a_\phi) \wedge \xi(a_\psi)}{\xi(a_\phi)}$$

would give us transitivity of the conditional $\vdash$ and inconsistency of the sentence above. And

$$\frac{\xi(a_{\phi \wedge \psi})}{\xi(a_{\psi \wedge \phi})}$$

would give us permutation of premisses for $\vdash$, etcetera. We are free to fix this inferential relation in any way we want, by formulating rules relating the properties of objects with different identifiers.

Reflexivity

An important difference between the conditional as defined here, and its standard formalization, comes out when we look at the principle of *reflexivity*, common to most sub-structural logics. i.e. for all $\phi(\bar{a})$

$$\phi(\bar{a}) \vdash \phi(\bar{a})$$

This is not a universally valid inference in our system, because $\mathcal{M}^* \models \phi(\bar{a}_{\phi})$ only holds in models where $\exists x \phi(x)$ holds. In other words $\phi(\bar{a}_{\phi})$ is not an axiom in our system, but the conclusion of a rule.

$$\frac{\exists x \phi(x)}{\phi(\bar{a}) \vdash \phi(\bar{a})}$$

Here $\exists x \phi(x)$ gives rise to the introduction of a *reflexivity statement*. What this means is that our proof system will only give us sound inferences for objects that have been *introduced* (markers).

Here are two valid rules: *Right Weakening* and *Conjunction*

$$\text{R.W.} \quad \frac{\psi(\bar{a}_{\phi}) \vdash \psi(x) \rightarrow \chi(x)}{\chi(\bar{a}_{\phi})} \quad \text{Conj.} \quad \frac{\psi(\bar{a}_{\phi}) \quad \chi(\bar{a}_{\phi})}{\psi(\bar{a}_{\phi}) \wedge \chi(\bar{a}_{\phi})}$$

That Reflexivity does not hold in general has the important consequence that $\phi(\bar{a}) \vdash \neg \phi(\bar{a})$ is generally not an inconsistent statement. This is as it should be, for in PO-models: $\neg \phi(\bar{a}_{\phi}) \leftrightarrow \forall x \neg \phi(x)$. To derive an inconsistency – by Right Weakening – we need $\phi(\bar{a}) \vdash \phi(\bar{a})$. Without reflexivity for $\phi(\bar{a})$ there is no guarantee that $\mathcal{M}^* \models \psi(\bar{a}_{\phi})$ for *classical* consequences ψ of ϕ .

In the next paragraph the rules *Cautious Monotony* and *Cautious Cut* will be used to get models for Discourse Representation Structures.

$$\text{C.M.} \quad \frac{\psi(\bar{a}_{\phi}) \quad \xi(\bar{a}_{\phi})}{\xi(\bar{a}_{\phi \wedge \psi})} \quad \text{C.C.} \quad \frac{\psi(\bar{a}_{\phi}) \quad \xi(\bar{a}_{\phi \wedge \psi})}{\xi(\bar{a}_{\phi})}$$

Together with Reflexivity, this gives the so-called Cumulative Logic (Gabbay [6], Kraus et al.[13]). Addition of the *Disjunction Rule* we get the so-called *Preference Logic*

$$\text{DISJ} \quad \frac{\phi(\bar{a}_\psi) \quad \phi(\bar{a}_\xi)}{\phi(\bar{a}_\psi \vee \bar{a}_\xi)}$$

Finally, adding the following *Monotonicity Rule*, we get Monotonic Logic.

$$\text{MON} \quad \frac{\phi(\bar{a}_\psi)}{\phi(\bar{a}_\psi \wedge \bar{a}_\xi)}$$

We can add conjuncts to the identifier without losing any properties.

3 Discourse Representation Theory

The framework of partial objects, seems to me a fruitful one for DRT for more than one reason. The main reason is that a marker never occurs in a DRS without an *inaugural description* — this happy phrase is by Marcus Kracht (Kracht[11]), i.e., the description it has been introduced with. So if we consider a DRS to be merely a set containing two kind of elements (markers and conditions) we overlook the fact that for every marker there invariably is a ‘special’ condition to which it is intimately related. Now, if we take a DRS to consist of a set of *indexed* objects and a set of conditions, this ever-present connection between a marker and its inaugural description is accounted for.

In a DRS, a marker $\bar{a}_\psi$ is *introduced* in the sentence $\phi(\bar{a}_\psi)$, its ‘first occurrence’ in the representation. When $\bar{a}_\psi$ occurs in any other sentence, the constant is used *anaphorically*. We will say that an occurrence is *bound* in a sentence ψ if $\psi \vdash \phi(\bar{a}_\psi)$. Dynamically, the introduction of a marker constitutes an application of EI. But here we must be flexible and allow for introduction of different kinds of markers, dependent on the phrase it is supposed to translate. The standard interpretation seems to suffice for indefinite noun phrases like “a man”. On the other hand quantified phrases like “most men” or bare plurals like “men” require introduction of markers that behave differently. Here is where the extended proof system comes to the fore. Depending on the kind of noun phrase we can introduce objects into different *logical domains*, i.e., domains of objects governed by a different logic on the indices (Cumulative Logic, Preferential Logic, Monotonic Logic, etcetera).

In the translation of sentences of natural language to logical formulas, we will have to choose whether to render phrases as *identifiers* of markers, or as *conditions* on them (as properties of objects). For instance, the sentence “A man walks” may give rise to $walk(a_{man})$, to $man(a_{walk})$ or to $man(a_{man \wedge walk}) \wedge walk(a_{man \wedge walk})$. The differences between these translations amount to *quantificational* differences: In the case of $\mathcal{M}^* \models walk(a_{man})$ we quantify over all men ($\forall v \in V : \mathcal{M} \models walk(v(a_{man}))$), but in $\mathcal{M}^* \models man(a_{walk})$ the domain of quantification consists of walkers. In the third case quantification is over the domain of all walking men. In the first two cases we can say that the predicate supplies *information* about the identified object: it *restricts the range of the objects*. Now I want to take this as the *basic interpretation* of formulas with identified objects: the predicates supply information about

the objects. In sentences of natural language we can invariably make a distinction between 'given' and 'new', 'topic' and 'comment', 'theme' and 'rheme': one part of a sentence supplies information about the other part. The sentence "a man *walks*" — where intonational stress is represented by italic — would then be translated into $walk(a_{man})$ and "a *man* walks" by $man(a_{walk})$.

As a general strategy for sentences with unmarked intonation contour, we will translate *noun phrases* into identifiers of markers, and *intransitive verbs* and the like, into conditions on the markers. An immediate consequence of the choice of moving only noun-phrases to the identifier is that *pronouns* will correspond to objects without identifiers.

In standard DRT and its dynamic variant PDL (Groenendijk & Stokhof [8]) only the most elementary use is made of the syntactic information "floating around" in a DRS. For instance, even a distinction as fundamental as that between noun phrase and verb phrase is lost in the translation: both translate to mere conditions on markers. It seems to me that in this way we are forced to construct representations with one hand tied behind our back. Also in *linguistically* motivated dynamic logic, we should allow ourselves a wide range of basic actions, under the control of all available linguistic information, instead of taking basic *programming* constructions as 'value assignment', 'sequential composition', etcetera as fundamental. Intimately connected with this point is the matter of the *signature* of our first-order representation language. Do we allow ourselves only predicate symbols relevant for the determination of *truth* of a DRS, or do we take a wider view and include predicates representing, for instance, linguistic features? The second position seems to me to be the correct one when we consider that truth-functionally equivalent DRS's can still determine different *correct continuations*. On the most general level, we can say that the first sentence of a text introduces a particular discourse to which the subsequent sentences refer anaphorically.

Within this wider perspective formulas $\psi(\bar{a}_\phi)$ can be interpreted as *attribute-value*, or feature-*dimension* feature-*value* expressions, where $\bar{a}_\phi$ determines an attribute (feature dimension) which is given a value by $\psi(x)$.

3.1 Basic Discourse Representation Structures

To start our formal discussion of DRT within the framework of partial object models, we restrict ourselves to *basic* DRS's, i.e., formulas $\psi(\bar{a}_\phi)$ where ϕ as well as ψ are *conjunctions of literals*. So, in this paragraph, whenever we refer to $\mathcal{L}$ -sentences, $\mathcal{L}(x)$ -formulas or $\mathcal{A}_{\mathcal{L}}$ constants, we mean only elements from this restricted language. We can think of such sentences and formulas as *derived* from a purely first-order theory through the basic proof system.

Definition 12 Let S be a finite set of $\mathcal{L}$ -sentences and $\mathcal{A}(S)$ the set of $\mathcal{A}_{\mathcal{L}}$ -constants occurring in S , then

- the set $M(S)$ of *markers* of S is given by $M(S) = \{\bar{a}_\phi \in \mathcal{A}(S) \mid S \vdash \phi(\bar{a}_\phi)\}$

- S is *closed* if $\mathcal{A}(S) \subseteq M(S)$
- S is *founded* if $|\mathcal{A}(S)| \subseteq M(S)$, i.e., for all $\bar{a}_\phi \in \mathcal{A}(S)$, if $\bar{a}_\psi \prec \bar{a}_\phi$ then $\bar{a}_\psi \in M(S)$

So, the markers of a set of sentences S , are the constants occurring in it that have been *introduced* ($\exists x \phi(x) \leftrightarrow \phi(\bar{a}_\phi)$), S is closed if all constants in S are markers and S is founded if all constants an element of $\mathcal{A}(S)$ depends on are markers.

Definition 13 Discourse Representation Structures

A *Discourse representation Structure* (DRS) is a finite, founded set of $\mathcal{L}$ -sentences. A DRS is *open* if it is not closed. A set S of $\mathcal{L}$ -sentences is a *possible continuation* of a closed DRS D if $D \cup S$ is closed.

A continuation S of structure D need not itself be closed. The set $\Delta(S) = (\mathcal{A}(S) \cup |\mathcal{A}(S)|) - M(S)$ will be called the set of *anaphors* of S . Anaphors can occur up to any finite depth along the dependency relation: a pronoun like 'he' is a top-level anaphor ($\in \mathcal{A}(S) - M(S)$), while a phrase like 'his brother' will translate to $\bar{a}_{\text{Brother}(x, \bar{b})} \in M(S)$ where $b \notin M(S)$. A finite set S of $\mathcal{L}$ -sentences is a possible continuation of some D if and only if S is $\vdash_M$ consistent.

We will represent discourses $\phi(\bar{a}_\phi) \wedge \xi(\bar{a}_\chi)$ by $[(\bar{a}_\phi) (\xi(\bar{a}_\chi))]$ where the left element contains the *markers* and the right one the *conditions*.

Definition 14 Truth Definition for DRS's

$\mathcal{M}^* \models [(\bar{a}_{\phi_1}^1 \dots \bar{a}_{\phi_n}^n) (\psi_1, \dots, \psi_k)]$ iff

- for all markers $\bar{a}_{\phi_i}^i : \mathcal{M}^* \models \phi_i(\bar{a}_{\phi_i}^i)$
- for all conditions $\psi_i : \mathcal{M}^* \models \psi_i$

Fact The following three equivalent

- $\mathcal{M}^* \models [(\bar{a}_\phi) (\emptyset)]$
- $\mathcal{M}^* \models [(\bar{a}_\phi) (\phi(\bar{a}_\phi))]$
- $\mathcal{M}^* \models [(\emptyset) (\phi(\bar{a}_\phi))]$

Consequently, we can turn the markers, as well as whole DRS's into sets of conditions. In this way, a DRS may occur as a condition in another DRS. In the next paragraph we will consider the reverse direction: turning DRS's into sets of markers. Remember that we are only considering structures built from literals; when we extend this to include *complex* DRS's, a structure may contain a second structure as a complex condition — of the form $\phi(\bar{a}_\phi) \wedge \psi(\bar{a}_\psi)$ — and such a condition will be seen to correspond to a DRS containing markers with *complex* indices.

DRS Logic

A logic for DRS's consists of a set of rules of the form $DRS_1 \dots DRS_n / DRS_{n+1}$ where $DRS_1, \dots, DRS_n$ are *premiss structures* and DRS_{n+1} is the *conclusion structure*.

Definition 15 Validity of DRS-Inferences

DRS_{n+1} is a consequence of $DRS_1 \dots DRS_n$ if the rule

$$\frac{DRS_1 \dots DRS_n}{DRS_{n+1}}$$

is case-to-case valid.

We can consider inferences within the framework of our *basic proof system*. However, I want to concentrate here on formulating DRS-rules within an *extended proof system*, because, for a 'dynamic' interpretation of DRT-semantics we need to relate markers via their identifiers, i.e., we need a logic on the *indices*. In the basic system only the *identity* of the index is relevant and even objects with logically equivalent indices may be mapped to disjoint subsets of M .

3.2 DRT and Extended Proofsystems

A DRS may contain two kinds of constants. (1) Constants that have been introduced by an application of EI: for any two such constants $\langle \bar{a}, \phi(x) \rangle, \langle \bar{b}, \psi(x) \rangle$ we have $\bar{a} \neq \bar{b}$. (2) Constants $\langle \bar{a}, \phi(x) \rangle, \langle \bar{b}, \psi(x) \rangle$ where $\bar{a} = \bar{b}$. Here, at least one of them is not the product of an application of EI. Such a pair, we will interpret as referring to the same object: we will consider one of them as an *update* of the other. Given $\phi(\bar{a}_\phi)$, the condition $\psi(\bar{a}_\phi)$ gives *information* about $\bar{a}_\phi$ and updating the marker with this information results in the new marker $\bar{a}_{\phi \wedge \psi}$ where the *same constant* from $\mathcal{A}$ is indexed by the extended formula. We will introduce some proof rules to achieve this.

3.2.1 Reduction to Objects

We will consider models for DRS's to be PO-models satisfying two additional principles, formulated as proof rules.

Definition 16 DRS-Models

Any model for a DRS will satisfy the rules of *Cautious Monotony* and *Cautious Cut*.

Corollary The following rule is valid on all DRS models

E-rule

$$\frac{\psi(\bar{a}_\phi)}{\xi(\bar{a}_\phi) \leftrightarrow \xi(\bar{a}_{\phi \wedge \psi})}$$

This follows, by conditionalization, from C.M. and C.C. So if ψ holds for $\bar{a}_\phi$, then $\bar{a}_{\phi \wedge \psi}$ and $\bar{a}_\phi$ satisfy the same formulas, they denote the same partial object.

Definition 17 Markers $\bar{a}_\phi, \bar{a}_\psi$ are equivalent, $\bar{a}_\phi \sim \bar{a}_\psi$ if $\mathcal{M}^* \models \chi(\bar{a}_\phi) \iff \mathcal{M}^* \models \chi(\bar{a}_\psi)$ for all $\chi \in \mathcal{L}(x)$. Equivalence classes are defined accordingly, i.e., $[\bar{a}_\phi] = \{\bar{a}_\psi \mid \bar{a}_\phi \sim \bar{a}_\psi\}$

As a consequence of the E-rule we have

$$\frac{\phi(\bar{a}_\phi) \quad \psi(\bar{a}_\psi)}{\phi(\bar{a}_{\phi \wedge \psi}) \wedge \psi(\bar{a}_{\phi \wedge \psi})}$$

Notice that we have a *reflexivity statement* $(\phi(\bar{a}_{\phi \wedge \psi}) \wedge \psi(\bar{a}_{\phi \wedge \psi}))$ that does not result from an application of EI. Now we have the following reduction lemma:

Lemma 3 Reduction to Objects

For every DRS, $[(\bar{a}_\phi) (\psi(\bar{a}_\phi))]$ and model $\mathcal{M}^*$ satisfying the E-Rule: if $\mathcal{M}^* \models [(\bar{a}_\phi) (\psi(\bar{a}_\phi))]$, then

- $\mathcal{M}^* \models [(\bar{a}_{\phi \wedge \psi}) (\emptyset)]$
- $a_\phi \sim a_{\phi \wedge \psi}$

Corollary Every closed discourse is reducible to a dependency-closed set of markers.

Note the key assumption that the DRS's are *closed*: an open DRS may be partially reducible, but it will always contain irreducible conditions.

This lemma will form the basis for our interpretation of Discourse Representation Structures. It entails that we can consider the *condition* $\psi(\bar{a}_\phi)$ of DRS $[(\bar{a}_\phi) (\psi(\bar{a}_\phi))]$ to be a map from $[(\bar{a}_\phi) (\emptyset)]$ to $[(\bar{a}_{\phi \wedge \psi}) (\emptyset)]$ preserving all conditions that hold for $\bar{a}_\phi$. So from a general perspective, DRS conditions can be viewed as *update functions* of the sets of markers involved.

$$[\psi(\bar{a})] : [(\bar{a}_\phi) (\emptyset)] \mapsto [(\bar{a}_{\phi \wedge \psi}) (\emptyset)]$$

and the relation between the traditional and the update perspective can be stated in a simple way:

Definition 18 Fixed Points

An equivalence class $[\bar{a}_\phi]$ of marker $\bar{a}_\phi$ is a *fixed point* for $\psi(\bar{a}_\phi)$ if $[\psi(\bar{a}_\phi)]([\bar{a}_\phi]) = [[\psi(\bar{a}_\phi)](\bar{a}_\phi)] = [\bar{a}_\phi]$.

Lemma 4 Update Lemma

$$\mathcal{M}^* \models [(\bar{a}_\phi) (\psi(\bar{a}_\phi))] \text{ iff } [\bar{a}_\phi] \text{ is a fixed point of } [\psi(\bar{a}_\phi)]$$

So, if $\psi(\bar{a}_\phi)$ already holds for marker $\bar{a}_\phi$ in $\mathcal{M}^*$, then updating only results in change of the *name* for the object a_ϕ .

To formulate the update perspective more formally. In closed DRS's, we may drop the indices of the constants occurring in the conditions. The *basic update functions*

will be literals in a language with *unindexed constants*; a constant $\bar{a}$ will be taken to refer to any object $a_\psi = \langle a, \psi(x) \rangle$ where $I(\bar{a}) = a$. These constants will play the role of *pronouns* and they will pick out from any state the current version of the a object. (Notice that the constants occurring in a condition of a DRS are thus taken to be *pronouns*).

Definition 19 Let ST be the set of all finite, founded, sets of markers. For $P(\bar{a}_1, \dots, \bar{a}_n)$ a $\mathcal{L}$ -literal with exactly the *unindexed constants* $\bar{a}_1, \dots, \bar{a}_n$, $[P(\bar{a}_1, \dots, \bar{a}_n)]$ is a partial function from ST into ST with domain $\{s \in ST \mid \forall i \leq n \exists \psi_i \langle a_i, \psi_i \rangle \in s\}$ and defined by

$$[P(\bar{a}_1, \dots, \bar{a}_n)](s) = \bigcup \{ [P(\bar{a}_1, \dots, \bar{a}_n)](\{a\}) \mid a \in s \}$$

where

$$[P(\bar{a}_1, \dots, \bar{a}_n)](\{ \langle b, \psi \rangle \}) = \begin{cases} \{ \langle b, \psi \wedge P[x/\bar{a}_i] \rangle \} & \text{if } a_i = b \\ \{ \langle b, \psi \rangle \} & \text{otherwise} \end{cases}$$

The dynamic strategy is now as follows:

- starting from a DRS D such that $M(D)$ consists of markers in $\mathcal{M}^*$, we update $M(D)$ with the conditions $C(D)$ in D . This gives a set $M^*(D)$ of markers in $\mathcal{M}^*$ for which the conditions in $C(D)$ hold. Hence $\mathcal{M}^*$ is a *standard model* for DRS $[M^*(D), C(D)]$
- a *basic continuation* of D is now interpreted as an extension $C^*(D)$ of $C(D)$, for which we repeat the first step.

Note that we only consider *basic* updates (which do not introduce new markers). Now, given a DRS D true in $\mathcal{M}^*$, the markers in $M(D)$ may satisfy more than just the elements of $C(D)$. These are the formulas *accidentally* satisfied by the markers in $\mathcal{M}^*$ (and we may consider them as *default* conditions). Updating with one of these formulas leaves the DRS unchanged (according to the update Lemma). On the other hand, they may be lost when we update by a formula not already true in $\mathcal{M}^*$. However, we never lose the conditions that have *explicitly* occurred in a DRS, as these have moved to the index.

Remark 1 The use of Cautious Monotony and Cautious Cut suggests that there will be an interpretation of DRS-consequence in terms of a *preferential relation*. In fact, this relation can be distilled from the update structure itself.

Definition 20 Refinement Relation

For markers $\bar{a}_\phi, \bar{b}_\psi$: $\bar{a}_\phi \triangleright \bar{b}_\psi \iff \bar{a} = \bar{b} \ \& \ \exists \xi \in \mathcal{L}(x) : \vdash \psi \leftrightarrow \phi \wedge \xi$

The relation $\bar{a}_\phi \triangleright \bar{a}_\psi$ is defined with respect to the classical notion of consequence; its intuitive interpretation is that $\bar{a}_\psi$ is a *further specification* or *refinement* of $\bar{a}_\phi$, i.e. ψ is ϕ extended by a conjunct. Note that $\bar{a}_\phi \triangleright \bar{a}_\psi$ if and only if there is an update mapping $\bar{a}_\phi$ to $\bar{a}_\psi$.

Proposition 1 Characteristic properties of the basic updates of objects

- $\bar{a} \triangleright [P](\bar{a})$
- $[P]([P](\bar{a})) \triangleright [P](\bar{a})$
- $\bar{a} \triangleright \bar{a}' \Rightarrow [P](\bar{a}) \triangleright [P](\bar{a}')$

Consequently, the basic updates are *fixpoint* operations on $\mathcal{A}$. These characteristic properties are a consequence of the fact that we have defined the $\triangleright$ -relation relative to the classical notion of derivability.

For equivalence classes of objects we set

- $[\bar{a}_\phi] \triangleright [\bar{a}_\psi]$ if there is a $\bar{a}_\xi \in [\bar{a}_\psi] : \bar{a}_\phi \triangleright \bar{a}_\xi$

So $[\bar{a}_\psi]$ is a refinement of $[\bar{a}_\phi]$ if $\bar{a}_\psi$ is equivalent to an update of $\bar{a}_\phi$. This is a reflexive and antisymmetric relation on equivalence classes (the latter is a consequence of the C.M. and C.C. principles).

Proposition 2 Characteristic properties of the basic updates of equivalence classes

- $[\bar{a}] \triangleright [P]([\bar{a}])$
- $[\bar{a}] \triangleright [\bar{a}'] \triangleright [P]([\bar{a}]) \Rightarrow [P]([\bar{a}]) \triangleright [P]([\bar{a}'])$
- $[\bar{a}] \triangleright [\bar{a}'] \triangleright [P]([\bar{a}]) \Rightarrow [P]([\bar{a}']) \triangleright [P]([\bar{a}])$

Again the basic updates are *fixpoint* operations on equivalence classes on $\mathcal{A}$.

This relation of refinement can be seen as a simple *preferential relation* on equivalence classes of markers, witness the following lemma:

Lemma 5

$\mathcal{M}^* \models \psi(\bar{a}_\phi)$ iff for all $\triangleright$ -minimal $[\bar{a}_\xi]$ such that $\mathcal{M}^* \models \phi(\bar{a}_\xi)$ we have $\mathcal{M}^* \models \psi(\bar{a}_\xi)$

Proof Whenever we have $\mathcal{M}^* \models \phi(\bar{a}_\xi)$, then, by the E-rule, $\bar{a}_\phi \wedge \xi \sim \bar{a}_\xi$ and consequently $[\bar{a}_\phi] \triangleright [\bar{a}_\xi]$. This shows both that $[\bar{a}_\phi]$ lies below all $[\bar{a}_\xi]$ satisfying $\phi(x)$ and, because of the antisymmetry of the $\triangleright$ relation, that $[\bar{a}_\phi]$ itself is the unique minimum of all these equivalence classes. $\square$

As we have mentioned, a formula $\psi(\bar{a}_\phi)$ can be interpreted as a conditional of the form $\phi(\bar{a}) \vdash \psi(\bar{a})$. In any DRS-model this conditional will satisfy Cautious Monotony and Cautious Cut, but Reflexivity does not hold in general: the set of markers of a DRS gives the formulas for which reflexivity holds, i.e., only if $\bar{a}_\phi \in M(D)$ then $\phi(\bar{a}) \vdash \phi(\bar{a})$.

So, as a general principle, whenever we need reflexivity $\phi(a) \vdash \phi(a)$ with respect to a formula $\phi(a)$ in a derivation from the conditions in a DRS, we have to look at the markers: if a_ϕ belongs to this set, then we can use reflexivity. From a constructive point of view, we can turn derivations into DRS's by starting with a set of conditions

and an empty set of markers; whenever we need reflexivity in the derivation, we add the appropriate marker to the discourse.

Consider the following DRS:

$$[(\bar{a}_\phi, \bar{a}_\psi) (\psi(\bar{a}_\phi), \phi(\bar{a}_\psi))]$$

Now, by Lemma 3, this DRS is reducible to

$$[(\bar{a}_{\phi \wedge \psi}) (\emptyset)]$$

where $\bar{a}_\phi \sim \bar{a}_\psi \sim \bar{a}_{\phi \wedge \psi}$. So a_ϕ and a_ψ satisfy the same formulas from $\mathcal{L}(x)$. But from the DRS $[(\bar{a}_\phi) (\phi(\bar{a}_\psi), \psi(\bar{a}_\phi))]$ we cannot conclude to the equivalence of a_ϕ and a_ψ as we could in the example above. Analyzing this situation within a derivational framework, the above reduction amounts to the rule

$$\frac{\phi(\bar{a}_\psi) \quad \psi(\bar{a}_\phi) \quad \xi(\bar{a}_\psi)}{\xi(\bar{a}_\phi)}$$

And, in the Cumulative Logic — axiomatized by C.M., C.C. and *Reflexivity* — this rule is derivable. (As an important corollary we have that for logically equivalent $\psi(x)$ and $\phi(x)$ the markers $\bar{a}_\phi$ and $\bar{a}_\psi$ are equivalent.) Within the present context, we notice that in this derivation we need reflexivity twice, in the form of $\phi(\bar{a}_\phi)$ and $\psi(a_\psi)$. This is why we need the markers $\bar{a}_\phi$ and $\bar{a}_\psi$ to prove the equivalence of the appropriate discourses.

In the present formalization of DRS's there are two kinds of statements. (1) Reflexivity statements, claiming the *existence* of a marker (the existence of an *attribute*), (2) conditions, supplying information about markers (giving *values* to attributes). This is a flexible framework when we take the DRS *signature* to include linguistic features beyond the predicates that function in the determination of truth values. The set-up of markers and conditions can then be seen as one of *feature-dimensions* and *feature-values* and anaphoric relations can occur on any level in the representational structure.

The update perspective is more general than the standard one in several ways. First of all, it makes sense for conditions that do not hold in a model; suppose $\mathcal{M}^* \models [(\bar{a}_\phi) (\psi(\bar{a}_\phi))]$ and we update $\bar{a}_\phi$ with $\chi(\bar{a}_\phi)$. This will result in $\{\bar{a}_{\phi \wedge \chi}\}$ where $\bar{a}_{\phi \wedge \chi}$ is a marker if $\mathcal{M}^* \models \phi(\bar{a}_{\phi \wedge \chi}) \wedge \chi(\bar{a}_{\phi \wedge \chi})$. Now the old and the new markers are different constants, so they may satisfy different formulas. In particular, we may have $\mathcal{M}^* \not\models [(\bar{a}_{\phi \wedge \chi}) (\psi(\bar{a}_{\phi \wedge \chi}))]$.

This implies that it we may have $\bar{a}_\phi \sim \bar{a}_{\phi \wedge \psi}$, but fail to have $\bar{a}_{\phi \wedge \chi} \sim \bar{a}_{\phi \wedge \psi \wedge \chi}$: updating equivalent markers with the same formula does not necessarily result in equivalent markers. Preservation of equivalence can be gained by addition of the *monotonicity* rule

$$\frac{\psi(\bar{a}_\phi)}{\psi(\bar{a}_{\phi \wedge \xi})}$$

Secondly, the update perspective can interpret conditions that fall outside of classical DRT altogether. Consider the structure $[(\bar{a}_\phi) (\exists x \psi(x, \bar{a}_\phi))]$. In standard DRT, this would not be a DRS because there no quantified conditions are allowed. But considered as an update $[\exists x \phi(x \bar{a}_\phi)](\bar{a}_\phi)$, it makes perfect sense: amongst other things, a new object will be introduced.

Definition 21 Existential Updates

For $\mathcal{L}$ -sentence $\exists x \psi$, the partial function $[\exists x \psi]$ from ST to ST with $dom([\exists x \phi(x)]) = dom([\phi(x)])$ is defined by $[\exists x \phi(x)](s) = [\phi(\bar{b}/x)](s \cup \{b_\top\})$ where $\bar{b}$ is a constant uniquely associated with $\phi(x)$.

3.3 Subordinate Structures

Finally, we will briefly discuss the extension of this framework to include *complex* DRS's.

First of all we will introduce a one-place predicate $D(y)$ where $D(\bar{d})$ is to be interpreted as " d is a discourse". Secondly, we will change the language: every $\mathcal{L}$ -predicate will get an extra position reserved for a *discourse identifier*. So predicates $P(x)$ become $P(x, y)$ where the y position has a special status in the sense that $\forall y (\phi(y) \rightarrow D(y))$ will always hold for this position. The markers in a DRS will now be dependent on a discourse identifier, present as a marker in the DRS. Whereas before the sentence $\exists x P(x)$ led to the introduction of $P(\bar{a}_{P(x)})$, we now have $\exists y \exists x P(x, y)$, which leads first to $\exists x P(x, \bar{d}_{\exists x P(x, y)})$, the introduction of the discourse identifier. Subsequent applications of EI introduce the markers of the discourse, now dependent on the discourse identifier (their 'visa').

The *first* sentence of a text introduces this identifier; it existentially quantifies the discourse position in the predicates. Subsequent sentences refer *anaphorically* to the identifier, i.e., they are of the form $\exists x \phi(x, \bar{d})$

It should be noted that, by introducing discourse identifiers, we can define the notion of *linked* discourse structures, where structure D is linked to D' if the identifier of D' is *dependent* on the identifier of D . By a linking mechanism we can treat, for instance, relative clauses.

The point of introducing discourse identifiers becomes clear when we consider Lemma 3 again. For now this Lemma allows us to move entire discourse structures to the index of the discourse identifier. Consider the following discourse structure with complex condition:

$$[(\bar{d}_\chi) ((\phi(\bar{a}_\phi, \bar{d}) \wedge \psi(\bar{a}_\phi, \bar{d})) \vee (\psi(\bar{a}_\phi, \bar{d}) \wedge \xi(\bar{a}_\phi, \bar{d})))]$$

By Lemma 3, this reduces to

$$[(\bar{d}_{\chi \wedge \gamma}) (\emptyset)]$$

where $\gamma(x) = (\phi(\bar{a}_\phi, x) \wedge \psi(\bar{a}_\phi, x)) \vee (\psi(\bar{a}_\phi, x) \wedge \xi(\bar{a}_\phi, x))$. In this way the semantics of complex discourse will be a simple extension of that for basic ones. However, it

is questionable whether we should proceed in such a direct way. For instance, implicational DRS's will be reduced to discourse identifiers with implicational indices. But there is much to say for a viewpoint where the implicational DRS's are treated as *inferences*, instead of implications. According to this perspective the *conclusion* of an inference is taken to supply *information* about the *premiss* (cf. Belnap[1], Kracht[12]). In this case we translate the DRS

$$[(\bar{d}_x) ((\phi(\bar{a}_\phi, \bar{d}) \wedge \psi(\bar{a}_\phi, \bar{d})) \rightarrow (\psi(\bar{a}_\phi, \bar{d}) \wedge \xi(\bar{a}_\phi, \bar{d})))]$$

into

$$[(\bar{d}_x) (\psi(\bar{a}_\phi, \bar{d}_{\phi(\bar{a}_\phi, x) \wedge \psi(\bar{a}_\phi, x)}) \wedge \xi(\bar{a}_\phi, \bar{d}_{\phi(\bar{a}_\phi, x) \wedge \psi(\bar{a}_\phi, x)}))]$$

The result is an *open* discourse as $\bar{d}_{\phi(\bar{a}_\phi, x) \wedge \psi(\bar{a}_\phi, x)}$ need not be a marker.

References

- [1] Belnap Nuel D., Jr.: 1970, *restricted Quantification and Conditional Assertion*. Truth, Syntax and Modality, Proc. Conf. Alternative Semantics, Temple Univ., Philadelphia Pa.
- [2] Copi I.: 1954, *Symbolic Logic*, Macmillan, New York, First edition.
- [3] Fine K.: 1985, Natural Deduction and Arbitrary Objects, *Journal of Philosophical Logic*, 14, pp 57–107.
- [4] Fine K.: 1985, *Reasoning with Arbitrary Objects*, Blackwell, Oxford.
- [5] Fine K.: 1984, A Defence of Arbitrary Objects, in E. Landman, F. Veltman (eds.) *Varieties of Formal Semantics*, Foris Publications, Dordrecht–Holland, pp 123–42.
- [6] Gabbay D.: 1985, Theoretical Foundations for non monotonic reasoning, in K. Apt (ed.) *Expert Systems, Logics and Models of Concurrent Systems*, Springer Verlag, pp 439–459.
- [7] Gabbay D.: 1991, *Labelled Deduction Systems*, Lecture Notes Third European Summer School in Language, Logic and Information, Saarbrücken.
- [8] J. Groenendijk & M. Stokhof: 1991, Dynamic Predicate Logic. *Linguistics and Philosophy*, 14.
- [9] Heim I.: 1982, *The Semantics of definite and Indefinite Noun Phrases*, Ph.D. thesis, University of Massachusetts.
- [10] Kamp H.: 1981, A Theory of Truth and Semantic Representation, in J.A.G. Groenendijk, T.M.V. janssen, M.B.J. Stokhof (eds.), *Formal Methods in the Study of Language*, Mathematical Centre Tracts, Amsterdam, Volume 136, pp 277–322.

- [11] Kracht M.: 1989, When can you say 'It', Gruppe für Logik, Wissenstheorie und Information, Freien Universität Berlin, Bericht Nr. 3/89.
- [12] Kracht M.: 1990, Traditional Linguistics can solve Logical Puzzles, Gruppe für Logik, Wissenstheorie Und Information, Freien Universität Berlin, Bericht Nr. 12/90
- [13] S. Kraus, D. Lehman, and M. Magidor, Preferential models and cumulative logics, to appear in *Artificial Intelligence*.
- [14] Zeevat H.: 1989, A Compositional Approach to Discourse Representation Theory. *Linguistics and Philosophy*, 12, pp 95-131.

Labelled Deductive Systems for categorial theorem proving

Michael Moortgat, OTS & CWI*

Abstract

Gabbay ([14], [15]) introduces a general theory of *structured* consequence relations and presents the discipline of Labelled Deductive Systems (LDS) as a unifying framework to study algorithmic proof theory (theorem proving) for structured deduction. In the present paper we present an LDS perspective on a range of categorial type systems. From the logical point of view, these systems are inhabitants of the landscape of *substructural logics*, cf. [5]. For a number of familiar type calculi the LDS discipline offers attractive solutions to problems of proof search when combined with the *proof net* approach of [32]. In the present paper a uniform LDS-based proof-net formulation for the occurrence logics L, LP, NL, NLP is developed and for the communication between these logics in terms of structural modalities, using an algebra of structural labelling to capture the distinct assumptions about the organization of the database (strings, multisets, trees, mobiles). Secondly, the paper shows that the LDS approach naturally accomodates type systems that extend the range of substructural logics in dimensions not covered by the standard structural rules, such as the dependency dimension of [25] which together with the dimensions of domain and order sensitivity generates a cube $\mathcal{L}$ of count-invariant logics.

1 Multi-dimensional linguistic inference

The point of departure of the categorial approach to natural language analysis is the idea that the properties of linguistic objects in various dimensions (syntax and semantics, on a very general level) can be captured uniformly in the form of type-declarations. The dynamic composition of *meaning* and the composition of linguistic *form* thus become integrated aspects of a general type-inference discipline. The

*Earlier versions of this paper were presented under the title 'Logics of accessible resources' at the ASL/LSA Conference on Logic and Linguistics (UCSC, July 1991) and at the VII Congrés de Llenguatges Naturals i Llenguatges Formals (Vic/Barcelona, September 1991). I thank these audiences, and that of the Amsterdam Colloquium, for comments. All errors are my own. Due to space limitations this paper has the character of an extended abstract. An expanded version will appear in due time. For linguistic background the reader should consult Morrill (1992) or Moortgat & Morrill (1991).

central objective in this area of research then is to study the properties of linguistic type-inference within an appropriate logical and computational setting.

The architectural components of categorial grammar are (i) a type theory, and (ii) a type logic. The role of the type theory is to provide an integrated specification of the properties of linguistic objects in the various dimensions which taken together provide the basis for a parametric characterization of natural language systems. The ingredients of the type theory are: a family of *basic types*, and a family of *type constructors*. The basic type inventory can be taken as a substantive theory of possible linguistic categories. The type-constructors provide the means of recursively constructing complex category objects out of more elementary ones. The theory of type-constructors sets the margins of available operations (syntactically, semantically) for this recursive build-up. The logic of the type-constructors then characterizes well-formed *inhabitants* of types in terms of logical derivability. Since we are dealing with constructive logics, the grammatical notion of well-formedness becomes a *constructive* notion — a theme which is fully exploited in the computational study of categorial type systems.

What exactly do we mean when we call linguistic inference multi-dimensional? On the most general level, as remarked above, linguistic material can be seen as a two-dimensional composite:

- *form* — syntax, in the widest sense, including the aspects of a highly organized prosodic form, versus
- *meaning* — semantics, likewise in the most general sense, including information structuring in addition to truth-conditional content.

Syntax and semantics in this liberal interpretation are themselves multifaceted wholes. The present paper studies in some detail the *structural* dimension of linguistic organization. More specifically, we study the effect of the fine-structure of linguistic *form* on derivable meanings. For the general methodological background of multi-dimensional composition, the reader should consult [29].

From a logical perspective, categorial type systems can be situated within a landscape of substructural logics — logics with a structure-sensitive consequence relation. The categorial research programme thus fits within a broader field of logical investigation aimed at establishing the connections between distinct ‘styles of inference’ (for dynamic, non-monotonic, resource conscious, ... reasoning) and their characteristic packages of structural rules, cf. [5, 6]. Research on substructural logics (cf. [9, 10] for a systematic study) has shown that the inhabitants of the substructural hierarchy can be systematically related by embedding translations on the basis of so-called *structural modalities*. The modal operators offer controlled access to stronger logics from within weaker ones by licensing of structural operations. From a linguistic point of view, the modalities provide the means to characterize phenomena of structural relaxation ultimately in the form of lexical type-declarations — the central point of the categorial agenda. Linguistic material exhibits structure in dimensions not covered by the standard structural rules. But current research shows that it is possible to generalize the modalisation and licensing strategy to

such dimensions: next to function-argument structure, we can address phrasal constituency structure, dependency structure, the organisation of linguistic material in terms of metrical prominence, pragmatic perspective, etc.

Summarizingly, we address inference problems of the general form

$$f(A_1, \dots, A_n) \Rightarrow B$$

(Can B be deduced from $A_1, \dots, A_n$?), to be interpreted as follows:

- $f(A_1, \dots, A_n)$: a *structured* configuration of assumptions $A_1, \dots, A_n$
- Assumptions: typed linguistic objects
- Structure: spatio-temporal organisation of linguistic form
- Dimensions: linear order, constituency, prosodic prominence, ...
- ' $\Rightarrow$ ': structure-sensitive consequence relation

In what follows, this general schema is worked out in the form of a family of categorial type systems. Sections 2–4 provide the logical background of the linguistic framework, bringing together lines of work from [16, 26, 28, 21, 23, 25], among others. In Sections 5–6 we study the logics from a computational point of view, providing them with a uniform algorithmic proof theory.

2 Logics of structured resources

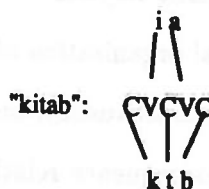
The common characteristic of the type systems that will be presented here — a characteristic which they share with Linear Logic — is resource-sensitivity: in the process of inference we actually use up the assumptions, to be considered as pieces of linguistic form. The logics differ in the distinct degrees of sensitivity for the structure of the resources. Each of the logics is essentially determined by the properties of a *structure-building* type-constructor which generically will be written here as ' $\circ$ '. Compare the tensor product of Linear Logic, which here stands for matter-like "addition" of structured configurations. The properties of the functor type-constructors (' $\leftarrow$ ' and ' $\rightarrow$ ', again in their generic form) are obtained from those of ' $\circ$ ' by *residuation*, cf. the defining inequalities below.

Residuation

$$A \leq C \leftarrow B \quad \Leftrightarrow \quad A \circ B \leq C \quad \Leftrightarrow \quad B \leq A \rightarrow C \quad (1)$$

Here is the intuition behind (1). To deal with linguistic inference, we need connectives for putting configurations together and for taking them apart. We are dealing with a logic, hence the pleasant duality between the two: what is put together can be taken apart; what is taken apart can be put together again. A compound $A \circ B$ is in principle asymmetric in its A and B component; therefore, right and left residuals of ' $\circ$ ' should in principle be distinguished. This asymmetry linguistic

composition shares with the sequential compositor ‘;’ of dynamic logic. But one should be careful not to restrict the generic ‘o’ to concatenation where the asymmetry is simple left-right order. Think, for example, of Semitic morphology, where the composition of linguistic form consists in the association of a vocalic melody and a consonantal root with a *CV* skeleton, as in (2). The work of Bird [7] shows that the notion of concatenation (or, as we would prefer: structural composition) can be generalized so as to cover these cases. Likewise, the concept of residuation would apply here, and provide ‘o’ in this autosegmental association mode with a residuation dual, just as it does in the ‘less exotic’ concatenation case.



(2)

Count invariance (Van Benthem)

Count-invariance is another way of stating the resource sensitivity and the positive/negative duality of the type systems under discussion. The type constructors exhibit the following monotonicity properties.

$$+ \circ + \quad - \rightarrow + \quad + \leftarrow -$$

Valid inference in our resource-conscious type logics requires matching counts for occurrences of atomic subtypes P , i.e. for derivable inferences $f(A_1, \dots, A_n) \Rightarrow B$ we have $c_P(A_1) + \dots + c_P(A_n) = c_P(B)$, where

$$c_P(A \circ B) = c_P(A) + c_P(B), \quad c_P(A \rightarrow B) = c_P(B \leftarrow A) = c_P(B) - c_P(A)$$

$$c_P(P') = 1 \text{ if } P = P', 0 \text{ otherwise } (P, P' \text{ atomic})$$

Semantics

When we talk about semantics in these sections, we refer to the interpretation of types in the structural dimension of linguistic form. The type logics will be interpreted in terms of structural algebras $\langle S, + \rangle$ with the ‘+’ operation acting as the counterpart of the generic addition type constructor ‘o’. Distinct settings of the properties of ‘+’ (in terms of parameters such as associativity, commutativity, ...) yield models for the distinct logics of $\mathcal{L}$. Basic types are interpreted as arbitrary subsets of S . The interpretation of compound types is then given by the clauses below. See e.g. [8] for completeness results.

$$\begin{aligned}
 [A \circ B] &= \{z \mid \exists x y [x \in [A] \ \& \ y \in [B] \ \& \ z = x + y]\} \\
 [C \leftarrow B] &= \{y \mid \forall x [x \in [B] \rightarrow y + x \in [C]]\} \\
 [A \rightarrow C] &= \{y \mid \forall x [x \in [A] \rightarrow x + y \in [C]]\}
 \end{aligned}
 \tag{3}$$

2.1 Rudimentary Lambek calculus (Lambek 1961)

The most rudimentary type system that qualifies as a logic in the sense of the above attributes to the structure building operator 'o' no properties at all beyond what is required for residuation. We obtain the non-associative type calculus of Lambek (1961). Types are interpreted in a structural algebra $\langle S, + \rangle$ of bracketed strings (trees). In the presentation below, we write $f : A \rightarrow B$ for a statement of a proof of the inclusion $[A] \subseteq [B]$. The arrows (4) thus capture the reflexivity and transitivity of the derivability relation. The inferences of (5) rephrase the residuation inequalities of (1).

$$\text{id} : A \rightarrow A \quad \frac{f : A \rightarrow B \quad g : B \rightarrow C}{g \circ f : A \rightarrow C} \quad (4)$$

$$\begin{array}{cc} \frac{f : A \bullet B \rightarrow C}{\beta(f) : A \rightarrow C/B} & \frac{f : A \bullet B \rightarrow C}{\gamma(f) : B \rightarrow A \backslash C} \\ \frac{g : A \rightarrow C/B}{\beta^{-1}(g) : A \bullet B \rightarrow C} & \frac{g : B \rightarrow A \backslash C}{\gamma^{-1}(g) : A \bullet B \rightarrow C} \end{array} \quad (5)$$

2.2 Structural combinators (cf. Došen)

From the rudimentary type system above more liberal type logics can be obtained by adding axiom schemata that attribute extra properties to '•'. One can make a distinction here between forms of structural relaxation that keep the type system within the family of resource-sensitive occurrence logics, and structural relaxation that leads beyond count invariance. The focus of the present paper is on the former. Below one finds examples.

Structural relaxation *within* the family of occurrence logics:

$$\begin{array}{ll} \text{Associativity} & \alpha : A \bullet (B \bullet C) \longleftrightarrow (A \bullet B) \bullet C : \alpha^{-1} \\ \text{Permutation} & \pi : A \bullet B \longleftrightarrow B \bullet A \end{array} \quad (6)$$

Structural options not preserving count invariance:

$$\begin{array}{ll} \text{Contraction} & \kappa : A \rightarrow A \bullet A \\ \text{Expansion} & \kappa^{-1} : A \bullet A \rightarrow A \\ \text{Monotonicity} & \mu : A \bullet B \rightarrow A \end{array} \quad (7)$$

Addition of the axiom schemata of (6) and (7) has a *global* effect: a system takes a quantum leap from absolute structure sensitivity in the relevant dimension to an absolute lack of it. For linguistic purposes, such abrupt modulation is much too crude. If language systems were 'pure', in the sense of exhibiting, say, total order sensitivity, or order freedom, parametric characterization would come down to fixing a suitable package of structural combinators. Linguistic reality is different: natural

languages in general will qualify as *mixed* systems with respect to the substructural hierarchy of type logics.

To deal with the mixed situation, we enrich the type language with *modal* operators, and replace the structural combinators by their modalized versions. Structural relaxation has to be explicitly *licensed* by modal types. As before, we'll use ' $\Box$ ' as the generic modality, and use subscripting for the different structural instantiations. See the modalized versions of the axiom schemata (6) and (7) below. For $\Box_\alpha, \Box_{\alpha^{-1}}, \Box_\pi$, we make the proviso that one of the factors A, B, C is modalized, i.e. of the form $\Box D$.

$$\begin{aligned} \Box_\alpha : A \bullet (B \bullet C) &\longleftrightarrow (A \bullet B) \bullet C & : \Box_{\alpha^{-1}} \\ \Box_\pi : A \bullet B &\longrightarrow B \bullet A \\ \Box_\kappa : \Box A &\longleftrightarrow \Box A \bullet \Box A & : \Box_{\kappa^{-1}} \\ \Box_\mu : A \bullet \Box B &\longrightarrow A \end{aligned} \quad (8)$$

How can we extend the structural interpretation of types to the modal type constructors? As will be shown below (in Section 4) the linguistic modalities exhibit S4 behavior. Their interpretation can be based then on the semantics for S4 logics, as has been shown for the categorial case in [16] and [28]: we assume a structural algebra $\langle S, + \rangle$ with subalgebra $\langle S', + \rangle$ ($S' \subseteq S$) and extend (3) with (9).

$$[\Box A] = \{x \mid x \in [A] \ \& \ x \in S'\} \quad (9)$$

Hepple [16] introduces this interpretation for a boundary modality which does not involve restructuring. Morrill [28] shows that in the case of structural modalities $\Box$; the operation '+' is governed by an appropriate set of structural equations explicitly linked to the elements s' of the modal subalgebra S' . For example, in the case of $\Box_\pi$ (the permutation modality), the relevant equation will be (10).

$$E_\pi : \forall s \in S, \forall s' \in S', \quad s + s' = s' + s \quad (10)$$

2.3 Dependency structure, Moortgat & Morrill (1991)

For the feasibility of the categorial programme outlined at the beginning of this paper, it is important to notice that parameters such as associativity and commutativity do not exhaust the possibilities for structural variation. Moortgat & Morrill [25] illustrate with another variation on the residuation theme, introducing *headed* variants of the structures considered so far (trees, strings, ...). The basic headed type system is obtained from its parent (the non-associative calculus) in terms of residuation on the basis of *two* product operators: lefthanded ' $\circ$ ' versus righthanded ' $\odot$ ' — lateralized versions of the ambidextrous ' $\bullet$ '. For linguistic motivation (in terms of semantic dependency, or prosodic prominence) the reader is urged to consult [25]. The decomposition of ' $\bullet$ ' introduces a dimension of *dependency structure* orthogonal with respect to function/argument structure. Compare: dependency grammar. Contrast: [2] where dependency structure is enforced by global derivational constraints.

In (11) and (12), we present the righthanded and lefthanded versions of the residuation inferences, leading to a distinction between *dominant* and *dependent* functors, $\{\bullet-, \rightarrow\}$ versus $\{\circ-, \rightarrow\}$ respectively.

$$\frac{\frac{f : A \circ B \rightarrow C}{\beta_{ws}(f) : A \rightarrow C \circ B} \quad g : A \rightarrow C \circ B}{\beta_{ws}^{-1}(g) : A \circ B \rightarrow C} \quad \frac{\frac{f : A \circ B \rightarrow C}{\gamma_{ws}(f) : B \rightarrow A \bullet C} \quad g : B \rightarrow A \bullet C}{\gamma_{ws}^{-1}(g) : A \circ B \rightarrow C} \quad (11)$$

$$\frac{\frac{f : A \bullet B \rightarrow C}{\beta_{sw}(f) : A \rightarrow C \bullet B} \quad g : A \rightarrow C \bullet B}{\beta_{sw}^{-1}(g) : A \bullet B \rightarrow C} \quad \frac{\frac{f : A \bullet B \rightarrow C}{\gamma_{sw}(f) : B \rightarrow A \circ C} \quad g : B \rightarrow A \circ C}{\gamma_{sw}^{-1}(g) : A \bullet B \rightarrow C} \quad (12)$$

In the algebraic semantics, the '+' operation is split up into right-dominant '+_{ws}' versus left-dominant '+_{sw}', yielding a structural algebra $\langle S, +_{ws}, +_{sw} \rangle$, where the interpretation function is realised as (13), for the family $\{\rightarrow, \bullet, \circ\}$, similarly for $\{\rightarrow, \bullet, \circ\}$. (The *ws*, *sw* subscripting refers to the strong (dominant, head) versus weak (dependent, non-head) daughter in the tree models.)

$$\begin{aligned} [A \circ B] &= \{z \mid \exists xy[x \in [A] \ \& \ y \in [B] \ \& \ z = x +_{ws} y]\} \\ [C \circ B] &= \{y \mid \forall x[x \in [B] \rightarrow y +_{ws} x \in [C]]\} \\ [A \bullet C] &= \{y \mid \forall x[x \in [A] \rightarrow x +_{ws} y \in [C]]\} \end{aligned} \quad (13)$$

Dependency: structural combinators

The introduction of an autonomous dependency dimension requires readjustment in our earlier structural combinators (and their modalized versions) if we wish to keep all the logical possibilities for interplay. The combinator $\delta_{ws/sw}$ destroys the contrast between left- and right-handed structures. In its presence, the type constructors ' $\circ$ ' and ' $\bullet$ ' collapse into the ambidextrous ' $\circ$ '. The combinators $\alpha_{ws/sw}$ and $\pi_{ws/sw}$ are the dependency variants of the earlier associativity and commutativity combinators — refinements which preserve dependency structure.

$$\begin{aligned} \alpha_{ws} : A \circ (B \circ C) &\longleftrightarrow (A \circ B) \circ C & : \alpha_{ws}^{-1} \\ \alpha_{sw} : A \bullet (B \bullet C) &\longleftrightarrow (A \bullet B) \bullet C & : \alpha_{sw}^{-1} \\ \pi_{sw} : A \bullet B &\longleftrightarrow B \bullet A & : \pi_{ws} \\ \delta_{ws} : A \circ B &\longleftrightarrow A \bullet B & : \delta_{sw} \end{aligned} \quad (14)$$

3 The $\{\alpha, \delta, \pi\}$ Cube

In Figure 1 (next page), we bring together the systems one obtains by considering the different combinations of the following three structural parameters:

- dependency: sensitivity for the distinction head/non-head
- commutativity: order sensitivity

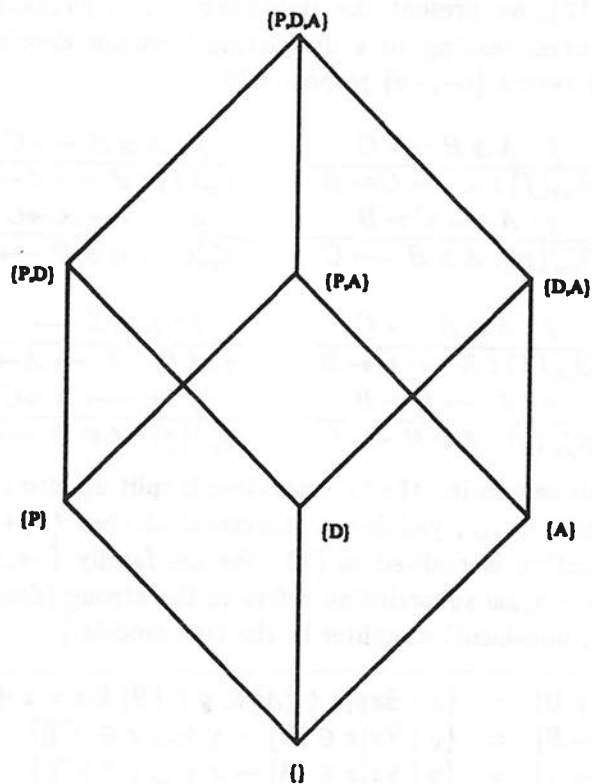


Figure 1: The family $\mathcal{L}$

- associativity: domain (constituent) sensitivity

As the reader will notice, the nomenclature is pretty chaotic (and by no means standard), mainly because Lambek's (1958) system L does not occupy a natural position (top or bottom) in the hierarchy. For the bottom plane of the cube, one obtains descriptive characterizations by taking 'headed' variants of the northern neighbors. Similarly, for the models, these systems are interpreted in terms of headed structures (trees, strings, mobiles or multisets).

RULES	NAME	DESCRIPTION	MODELS
$\{\}$	HNL		
$\{\delta\}$	NL	non-associative Lambek calculus	trees
$\{\alpha\}$	HL		
$\{\pi\}$	HNLP		
$\{\delta, \alpha\}$	L	associative Lambek calculus	strings
$\{\delta, \pi\}$	NLP	non-associative Lambek-Van Benthem	mobiles
$\{\alpha, \pi\}$	HLP		
$\{\delta, \alpha, \pi\}$	LP	Lambek-Van Benthem calculus	multisets

4 Gentzen calculus

The previous section discussed categorial systems in terms of type-constructors and a relation of derivability holding between types. From a computational point of view it is attractive to move to an equivalent Gentzen style presentation. For the logics under consideration, the Cut rule (the Gentzen variant of the transitivity of the derivability relation) can be eliminated: proof search thus gets an immediate algorithmic interpretation. The reader should notice that Gentzen's original notion of a calculus of *sequents* ('Sequenzenkalkül') is a bit misleading in our present setting, since we do not want to restrict ourselves to the logic of sequences, but attempt to characterize the properties of deduction on the basis of structured configurations of assumptions in general. Following Belnap [3, 4], we can make a distinction here between *logical* connectives (i.e. our type constructors, taking types into types) and *structural* connectives (taking structures into structures). One could say that Gentzen calculus characterizes the properties of its logical connectives in terms of a structural connective: the comma of sequent calculus. In the generalized setting, the properties of families of logical connectives are characterized in terms of families of structural connectives. See Dunn's work on relevant implication (e.g. [12]) for such an approach.

Just as we had the generic family of logical connectives $\{\rightarrow, \circ, \leftarrow\}$, the comma punctuation in our Gentzen calculi becomes the generic structural connective, with distinct instantiations for the various logics in $\mathcal{L}$. As an illustration, Figure 2 (next page) gives the Gentzen presentation of the dependency logic HNL. The generic punctuation mark is instantiated here in the form of the structural connectives ' $(_, _)^+$ ' and ' $(_, _)^-$ ', structural counterparts of the logical connectives ' $\odot$ ' and ' $\oslash$ ' respectively.

4.1 Substructural travel: modalities

Simple modalities

Let us turn now to the Gentzen presentation of the modal operators. Just like any logical connective/type constructor, ' $\Box$ ' needs a rule of use $[\Box L]$ and a rule of proof $[\Box R]$. The rule of use states that a modal assumption $\Box A$ will ultimately be used as an ordinary datum of type A . The rule of proof states that a modal conclusion can only be obtained from a database of assumptions which is fully modalized. In $[\Box L]$ and $[\Box R]$ we recognize the sequent rules for modal S4 systems.

$$[\Box L] \frac{\Gamma[A] \Rightarrow B}{\Gamma[\Box A] \Rightarrow B} \quad \frac{\Box \Gamma \Rightarrow A}{\Box \Gamma \Rightarrow \Box A} [\Box R] \quad (15)$$

Next to the generic S4 base, structural modalities have an operational rule for controlled structural manipulation. The control part states that a subconfiguration Δ in Γ can be replaced by a structurally equivalent Δ' provided Δ contains a modal component *licensing* structural relaxation. The general pattern thus becomes:

$$\begin{array}{c}
[Ax] \frac{}{A \Rightarrow A} \quad \frac{\Delta \Rightarrow A \quad \Gamma[A] \Rightarrow B}{\Gamma[\Delta] \Rightarrow B} [Cut] \\
[R\bullet-] \frac{(\Delta, B)^- \Rightarrow A}{\Delta \Rightarrow A \bullet B} \quad \frac{\Delta \Rightarrow B \quad \Gamma[A] \Rightarrow C}{\Gamma[(A \bullet B, \Delta)^-] \Rightarrow C} [L\bullet-] \\
[L\bullet] \frac{\Gamma[(A, B)^-] \Rightarrow C}{\Gamma[A \bullet B] \Rightarrow C} \quad \frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{(\Gamma, \Delta)^- \Rightarrow A \bullet B} [R\bullet] \\
[R\multimap] \frac{(B, \Delta)^- \Rightarrow A}{\Delta \Rightarrow B \multimap A} \quad \frac{\Delta \Rightarrow B \quad \Gamma[A] \Rightarrow C}{\Gamma[(\Delta, B \multimap A)^-] \Rightarrow C} [L\multimap] \\
[R\multimap] \frac{(B, \Delta)^+ \Rightarrow A}{\Delta \Rightarrow B \multimap A} \quad \frac{\Delta \Rightarrow B \quad \Gamma[A] \Rightarrow C}{\Gamma[(\Delta, B \multimap A)^+] \Rightarrow C} [L\multimap] \\
[L\bullet] \frac{\Gamma[(A, B)^+] \Rightarrow C}{\Gamma[A \bullet B] \Rightarrow C} \quad \frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{(\Gamma, \Delta)^+ \Rightarrow A \bullet B} [R\bullet] \\
[R\multimap] \frac{(\Delta, B)^+ \Rightarrow A}{\Delta \Rightarrow A \multimap B} \quad \frac{\Delta \Rightarrow B \quad \Gamma[A] \Rightarrow C}{\Gamma[(A \multimap B, \Delta)^+] \Rightarrow C} [L\multimap]
\end{array}$$

Figure 2: Gentzen calculus HNL

$$[\Box_i] \frac{\Gamma[\Delta'] \Rightarrow B}{\Gamma[\Delta] \Rightarrow B} \text{ provided } \Delta \text{ contains a component carrying the } \Box_i \text{ modality (16)}$$

with the following instantiations for $\Box_i$:

$$\begin{array}{ll}
\alpha_{ws} & \Delta = (\Delta_1, (\Delta_2, \Delta_3)^+)^+ \quad \Delta' = ((\Delta_1, \Delta_2)^+, \Delta_3)^+ \\
\alpha_{sw} & \Delta = (\Delta_1, (\Delta_2, \Delta_3)^-)^- \quad \Delta' = ((\Delta_1, \Delta_2)^-, \Delta_3)^- \\
\pi_{sw} & \Delta = (\Delta_1, \Delta_2)^- \quad \Delta' = (\Delta_2, \Delta_1)^+ \\
\pi_{ws} & \Delta = (\Delta_1, \Delta_2)^+ \quad \Delta' = (\Delta_2, \Delta_1)^- \\
\delta_{sw} & \Delta = (\Delta_1, \Delta_2)^- \quad \Delta' = (\Delta_1, \Delta_2)^+ \\
\delta_{ws} & \Delta = (\Delta_1, \Delta_2)^+ \quad \Delta' = (\Delta_1, \Delta_2)^-
\end{array} \tag{17}$$

Composite modalities

Simple modalities regulate the communication between immediate neighbors in $\mathcal{L}$. We can combine different options of (17) into composite modalities allowing traffic between more distant relatives. For an example outside $\mathcal{L}$, consider the $\Box!$ exponential from Linear Logic which to the S4 basis adds the combination of Contraction and Weakening.

- Reusability of resources (Contraction)

$$[\Box; C] \frac{\Gamma[\Box; A, \Box; A] \Rightarrow B}{\Gamma[\Box; A] \Rightarrow B}$$

- Waste of resources (Weakening)

$$\frac{\Gamma \Rightarrow B}{\Gamma[\Box; A] \Rightarrow B} [\Box; W]$$

For such composite modalities we'll use sets of structural indices taken from $\mathcal{P}(\{\alpha, \delta, \pi\})$, using S, M as metavariables ranging over such sets.

4.2 Embedding translations

The essential point about the modalities is that they provide the capacity to recover the stronger logics from within the weaker (i.e. more structure sensitive) ones by means of embedding translations. See [34] for Girard's translation between Intuitionistic and Linear Logic, and [11] for the general pattern. In (18) the operators $\circ$ and $\circ'$ are relatives with respect to some set of structural rules/modalities M . For example, let $\circ$ be the structural addition operator of NLP and let M be $\{\pi\}$, then $\circ'$ is the addition operator of NL. The permutation modality $\Box_\pi$ mediates between these two relatives.

$$\begin{aligned} |A| &:= A \text{ for } A \text{ atomic} \\ |(A \circ B)| &:= \Box_M |A| \circ' \Box_M |B| \\ |(A \rightarrow B)| &:= \Box_M |A| \rightarrow' |B| \\ |(B \leftarrow A)| &:= |B| \leftarrow' \Box_M |A| \end{aligned} \tag{18}$$

Embedding: licensing modalities

In the standard interpretation of structural modalities, the purpose of modalization is to *license* a structural operation which is lacking from the propositional part of the logic in question. In the landscape $\mathcal{L}$, licensing modalities represent travel 'upwards', from a more highly structure-sensitive logic to a logic with coarser structure-sensitivity. Let $\mathcal{L}_S$ be a system in the hierarchy characterized by a set of structural modalities S , and $\mathcal{L}_{S \setminus M}$ a neighboring system lacking the structural modalities M . To obtain an embedding, i.e. recover $\mathcal{L}_S$ from within $\mathcal{L}_{S \setminus M}$, one modalizes all critical subtypes in $\mathcal{L}_{S \setminus M}$ with the licensing modality $\Box_M$.

$$\mathcal{L}_S \vdash \Gamma \Rightarrow A \text{ iff } \mathcal{L}_{S \setminus M} \vdash \Box_M |\Gamma| \Rightarrow |A| \tag{19}$$

Illustration

We illustrate the modalization strategy with the type transition

$$A/B \Rightarrow (A/C)/(B/C) \tag{20}$$

which is the characteristic theorem discriminating between systems without and systems with constituency sensitivity — in the former (20) is valid, in the latter it is not.

Consider the situation where the point of departure is NL, the non-associative Lambek calculus. The derivation of the sequent (20) blocks where we try to consume the resource C in using the functor B/C . Although as far as the dimension of linear order is concerned, the resource C is appropriately situated with respect to B/C , it is inaccessible because of the intervening bracket which locks B/C together with A/B within a constituency domain. (We use ceilings for structural bracketing here to avoid confusion with the bracketing in types. Outer structural brackets omitted.)

$$\frac{\text{FAIL}}{\frac{[A/B, B/C], C \Rightarrow A}{A/B, B/C \Rightarrow A/C}} \quad (21)$$

$$\frac{}{A/B \Rightarrow (A/C)/(B/C)}$$

However, the modal translation of (20)

$$\Box_\alpha(A/\Box_\alpha B) \Rightarrow (A/\Box_\alpha C)/\Box_\alpha(B/\Box_\alpha C) \quad (22)$$

is derivable within NL. Where the non-modal derivation halted, the licensing modality $\Box_\alpha$ allows $\Box_\alpha C$ to find its proper position within the dominance structure — a position where it forms a constituent together with $\Box_\alpha(B/\Box_\alpha C)$.

$$\frac{\Box_\alpha(A/\Box_\alpha B), [\Box_\alpha(B/\Box_\alpha C), \Box_\alpha C] \Rightarrow A}{[\Box_\alpha(A/\Box_\alpha B), \Box_\alpha(B/\Box_\alpha C)], \Box_\alpha C \Rightarrow A} [\Box_\alpha] \quad (23)$$

Excursion: constraining modalities

From a linguistic point of view, it would be attractive to develop an alternative, complementary view, where modalities are interpreted as *constraints*, i.e. where modalization blocks a structural operation which is by default available in the propositional part of the logic in question. In the hierarchy $\mathcal{L}$, constraining modalities would then represent travel 'downwards', from a structurally permissive logic to a system with a higher degree of structure-sensitivity. To obtain $\mathcal{L}_{S \setminus M}$ from within $\mathcal{L}_S$, one would again modalize all critical subtypes in $\mathcal{L}_S$ with the constraining modality $\Box'_M$.

$$\mathcal{L}_{S \setminus M} \vdash \Gamma \Rightarrow A \quad \text{iff} \quad \mathcal{L}_S \vdash \Box'_M |\Gamma| \Rightarrow |A| \quad (24)$$

Linguistic motivation for such a complementary view on modalities comes from the study of parametric variation. Consider the problem of situating a language system within the hierarchy $\mathcal{L}$. Suppose one only has structural relaxation (i.e. licensing modalities) for the characterization of mixed language systems. Then one has no alternative but to localize a language system as low as possible within the hierarchy $\mathcal{L}$, with modal type declarations for all lexical elements whose behaviour is governed by a more permissive type logic. This will result in a distorted picture

of, say, non-configurational language systems with some tiny subsystem characterized by rigid order sensitivity. With the complementary perspective of constraining modalities, one could anchor mixed language systems within $\mathcal{L}$ where they fit in most congenially, and have modal type declarations both for travel to more permissive and to more constrained areas. This would introduce the notion of *default* parameter settings in the categorial discussion of parametric variation.

Limitations of time and space make it impossible to pursue here the proof rules and algebraic interpretation of the constraining variant of the modalities. We leave the subject as a desideratum for further research.

5 Labelled Deductive Systems

In order to realize the categorial parsing-as-deduction programme one would like to have an efficient and uniform algorithmic proof theory for the distinct logics in $\mathcal{L}$ and, in particular, for the communication between these logics in terms of structural modalities. It turns out that Gabbay's general theory of structured consequence relations and Labelled Deductive Systems (cf. [14], [15]) provide a particularly natural framework for the computational study of categorial type systems.

LDS considers logics where the basic declarative unit is not the formula, but the *labelled* formula $t : A$. The traditional consequence relation

$$A_1, \dots, A_n \Rightarrow B \quad (25)$$

(deduction of B from a database of assumptions $A_1, \dots, A_n$) is replaced by

$$t_1 : A_1, \dots, t_n : A_n \Rightarrow s : B \quad (26)$$

The labels are taken from a labelling algebra exhibiting the structural properties appropriate for the application one has in mind. Labels enter into configurations: the concept of a *set* of assumptions is replaced by that of a structured database. Logical inference then depends on the structure of the assumptions. Structured consequence is defined not just in terms of proof rules on the formulas, but in terms of rules that operate on both the formulas and their labels. From a sequent perspective, for every logical rule introducing a connective in the antecedent (database formulae) or the succedent (goal formula), we have to indicate how in the labelling algebra the labels *propagate* between conclusions and premises.

The present section introduces the concept of labelled deduction in a categorial setting, with reference to the *sign-based* categorial systems of [22] and [23]. The purpose of this section is twofold: to systematize our earlier work in the light of the more articulate substructural landscape $\mathcal{L}$, and to extend it so as to obtain an LDS account of the structural modalities. We proceed in two stages. First, we present a uniform Gentzen characterization of the family of occurrence logics in terms of *structural* labelling. The Gentzen presentation is chosen so as to make a smooth transition between this section and the discussion in the previous sections. Computationally, the Gentzen presentation is not particularly efficient because of

the spurious non-determinism inherent in sequent proof search. In the second stage, then, we address the efficiency issue, and move to an LDS proof net presentation.

5.1 Sign-based labeling

In the figure below, we give the Gentzen presentation of the family $\mathcal{L}$ in LDS terms. Because we want to represent linguistic structure in a number of relevant dimensions, it will be handy to represent the declarative unit $t : A$ as an *array* $\langle A, L_1, \dots, L_n \rangle$, representing a type A labelled in dimensions $L_1, \dots, L_n$. The units below are *signs* in the sense of UCG or HPSG, i.e. structures of the form

$$\langle \text{Type}, \text{Semantics}, \text{Structure} \rangle$$

The semantic labelling implements the Curry-Howard correspondence between type deductions and λ terms, cf. [5]. The correspondence is constant for the logics under consideration. The set of derivable terms will depend on the ‘filtering’ effect of structure-sensitivity in the dimension of linguistic form, of course. The structural labelling algebra is equipped with a ‘+’ operation, with distinct properties for the various structural options. Depending on the associativity, commutativity and dependency properties attributed to ‘+’, we obtain the distinct occurrence logics in $\mathcal{L}$. Labels are built up from parameters/constants and variables (the latter indicated by overlining in (32) below). The LDS variant of a goal sequent $A_1, \dots, A_n \Rightarrow B$ is obtained by assigning fresh parameter labels to the antecedent types $A_1, \dots, A_n$, and computing the unknown labels for the succedent type B by means of the proof rules (27)–(31), with label propagation as in (32).

$$[\text{Ax}] \frac{}{\langle A, t, \sigma \rangle \Rightarrow \langle A, u, \tau \rangle} \quad (27)$$

$$[\text{R} \leftarrow_s] \frac{\Delta, \langle B, v, \tau \rangle \Rightarrow \langle A, u, \phi \rangle}{\Delta \Rightarrow \langle A \leftarrow_s B, w, \sigma \rangle} \quad \frac{\Delta, \langle B, v, \tau \rangle \Rightarrow \langle A, u, \psi \rangle}{\Delta \Rightarrow \langle B \rightarrow_s A, w, \sigma \rangle} [\text{R} \rightarrow_s] \quad (28)$$

$$[\text{L} \leftarrow_s] \frac{\Gamma \Rightarrow \langle B, x, \tau \rangle \quad \Delta[\langle A, s, \phi \rangle] \Rightarrow C}{\Delta[\langle A \leftarrow_s B, t, \sigma \rangle, \Gamma] \Rightarrow C} \quad \frac{\Gamma \Rightarrow \langle B, x, \tau \rangle \quad \Delta[\langle A, s, \psi \rangle] \Rightarrow C}{\Delta[\langle B \rightarrow_s A, t, \sigma \rangle, \Gamma] \Rightarrow C} [\text{L} \rightarrow_s] \quad (29)$$

$$[\text{Lo}_s] \frac{\Delta[\langle A, r, \sigma \rangle, \langle B, s, \tau \rangle] \Rightarrow C}{\Delta[\langle A \circ_s B, t, \phi \rangle] \Rightarrow C} \quad (30)$$

$$[\text{Ro}_s] \frac{\Delta \Rightarrow \langle A, u, \sigma \rangle \quad \Gamma \Rightarrow \langle B, v, \tau \rangle}{\Delta, \Gamma \Rightarrow \langle A \circ_s B, w, \phi \rangle} \quad (31)$$

Let us comment on the proof rules. Notice first that the interpretation of the comma punctuation is maximally non-committed now (i.e. multiset union, for our family of occurrence logics): resource management is taken care of by the structural labelling. As before, we generalize over dimensions for structural discrimination (linear precedence, dominance, prominence) by subscripting the type-constructors

with the structural packages that regulate their behavior, cf. the indexing of connectives in [3]. For the structural labelling algebras we use the algebraic semantics for type systems discussed above. This highlights a characteristic feature of the LDS approach: the mixture of syntax (here: Gentzen proof rules) and semantics (here: the structural algebras where types find their interpretation). The propagation of labels has to satisfy the equations of (32).

SEMANTICS	SYNTAX	
$\bar{u} = t$	$\langle \bar{\tau} = \sigma \rangle_S$	in (27)
$w = \lambda v. \bar{u}$	$\phi = \bar{\sigma} +_S \tau, \psi = \tau +_S \bar{\sigma}$	in (28)
$s = t\bar{x}$	$\phi = \sigma +_S \bar{\tau}, \psi = \bar{\tau} +_S \sigma$	in (29)
$r = \pi_0(t), s = \pi_1(t)$	$\phi = \sigma +_S \tau$	in (30)
$w = \langle \bar{u}, \bar{v} \rangle$	$\phi = \bar{\sigma} +_S \bar{\tau}$	in (31)

(32)

For the Axiom case, we require antecedent and succedent labels to unify. In the case of structural labels this means unification under the appropriate equational theory, which with a slight abuse of notation we refer to by the corresponding structural index S . For example: unification under Associativity for L, under Associativity+Commutativity for LP, etc. Cf. Siekmann [33] for a general overview of the computational properties of such unification problems.

5.2 Labelled proof nets

Proof nets provide an optimal data structure to represent type deductions. See [32] for discussion. The proof net approach is based on formula (type) decomposition and resolution matching of signed literals. To check well-formedness of candidate proof-net graphs, different algorithms have been proposed, notably the graph coloring algorithm of [32] (for L, LP) and the string-unification algorithm of [22] (for L). The purpose of this section is to show that the generalization of string-unification, viz. structure dependent unification on syntactic labelling in the LDS approach defended here, encompasses the complete spectrum of occurrence logics.

The translation from the labelled sequent inferences above to proof net formula decomposition is straightforward. A sequent $\Gamma \Rightarrow B$ is represented as a multiset of labelled types, i.e. arrays of the form

$$\langle \text{Type}, \text{Polarity}, \text{Semantics}, \text{Syntax} \rangle$$

The polarity label keeps track of the positive/negative nature of types, which in the sequent format is encoded in their occurrence to the left or to the right of the $\Rightarrow$ sign. Antecedent (database) types $A \in \Gamma$ are assigned a label 1, the succedent (goal) type B is labelled 0. The association of types with a lambda term label and a structural label is exactly as before. Proof net unfolding of types then assumes the following form.

$$[Ax] \frac{}{\langle A, 1, t, \sigma \rangle \quad \langle A, 0, \bar{u}, \bar{\tau} \rangle} \quad \bar{u} = t, \langle \bar{\tau} = \sigma \rangle_S \quad (33)$$

Axiom links cancel an antecedent atom against a matching succedent atom under the unifying substitution for the lambda term label and the structural label (representing strings, trees, ... depending on the logic one deals with). The logical links recursively decompose types of arbitrary complexity into their atomic subtypes, and propagate labels according to the relevant structure dependent notion of addition '+s'.

$$\begin{array}{c} [L \leftarrow_s] \frac{\langle A, 1, t\bar{u}, \sigma +_s \bar{\tau} \rangle \quad \langle B, 0, \bar{u}, \bar{\tau} \rangle}{\langle A \leftarrow_s B, 1, t, \sigma \rangle} \\ [R \leftarrow_s] \frac{\langle B, 1, v, \tau \rangle \quad \langle A, 0, \bar{u}, \bar{\sigma} +_s \tau \rangle}{\langle A \leftarrow_s B, 0, \lambda v.\bar{u}, \bar{\sigma} \rangle} \end{array} \quad (34)$$

$$\begin{array}{c} [L \rightarrow_s] \frac{\langle A, 1, t\bar{u}, \tau +_s \sigma \rangle \quad \langle B, 0, \bar{u}, \bar{\tau} \rangle}{\langle B \rightarrow_s A, 1, t, \sigma \rangle} \\ [R \rightarrow_s] \frac{\langle B, 1, v, \tau \rangle \quad \langle A, 0, \bar{u}, \tau +_s \bar{\sigma} \rangle}{\langle B \rightarrow_s A, 0, \lambda v.\bar{u}, \bar{\sigma} \rangle} \end{array} \quad (35)$$

$$\begin{array}{c} [Lo_s] \frac{\langle A, 1, \pi_0(t), \sigma \rangle \quad \langle B, 1, \pi_1(t), \tau \rangle}{\langle A \circ_s B, 1, t, \phi \rangle} \quad \phi = \sigma +_s \tau \\ [Ro_s] \frac{\langle B, 0, \bar{u}, \bar{\sigma} \rangle \quad \langle A, 0, \bar{v}, \bar{\tau} \rangle}{\langle A \circ_s B, 0, \langle \bar{u}, \bar{v} \rangle, \bar{\sigma} +_s \bar{\tau} \rangle} \end{array} \quad (36)$$

5.3 World labels for modalities

Let us turn now to the computational treatment of the modalities. In the field of Automated Deduction for modal logics a common strategy consists in annotating formulas with terms, and supplying specialized unification algorithms capturing the properties of the accessibility relation for the logic in question for these terms. Wallen [35] thus proposes a *string* annotation, and distinct string unification algorithms for modal logics K, S4 and S5. The string annotation approach can be recast in LDS terms. Cf. also Chapter 6 in [14]. Hollenberg [17] shows how Wallen's string unification method can be incorporated into the labelled proof net approach. This takes care of the S4 basis for the categorial modalities. We present Hollenberg's approach below, and must refer here to [17] for explanation and illustrations. Our objective in this section is to show how the approach can be extended to deal with the *structural* component of the modalities.

Labelled formulas are extended to quintuples of the form

$$\langle \text{Type}, \text{Polarity}, \text{Semantics}, \text{Syntax}, \text{World} \rangle$$

where the extra World label is built up from world constants and world variables with the string concatenation operator '.' (associative, non-commutative). For the axiom links, labels have to unify. String unification on the World labels, in the case of S4, is such that variables unify with any string, including the empty string.

$$[Ax] \frac{}{\langle A, 1, t, \sigma, q \rangle \quad \langle A, 0, u, \tau, p \rangle} \quad u = t, \langle \tau = \sigma \rangle_s, p = q \quad (37)$$

The crucial new component is the propagation of World labels in the $[L\Box]$ and $[R\Box]$ case. To prove an assumption $\Box A$ at label p , we create a new (accessible) world, represented by the label $p \cdot c$ (c a fresh world constant) and prove A at $p \cdot c$. To use an assumption $\Box A$ associated with world label p , use an assumption A associated with label $p \cdot v$ (v a fresh variable) — where the label expresses the condition that it is a point reachable from p in the present configuration.

$$v \in \text{VAR}_W \quad [L\Box] \frac{\langle A, 1, t, \sigma, p \cdot v \rangle}{\langle \Box A, 1, t, \sigma, p \rangle} \quad [R\Box] \frac{\langle A, 0, u, \sigma, p \cdot c \rangle}{\langle \Box A, 0, u, \sigma, p \rangle} \quad c \in \text{CON}_W \quad (38)$$

The rules for non-modal types propagate the World labels unchanged from conclusion to premises. Below the ' $\leftarrow$ ' case; similarly for ' $\rightarrow$ '.

$$\begin{aligned} [L\leftarrow] & \frac{\langle A, 1, tu, \sigma + \tau, p \rangle \quad \langle B, 0, u, \tau, p \rangle}{\langle A \leftarrow B, 1, t, \sigma, p \rangle} \\ [R\leftarrow] & \frac{\langle B, 1, v, \tau, p \rangle \quad \langle A, 0, u, \sigma + \tau, p \rangle}{\langle A \leftarrow B, 0, \lambda v.u, \sigma, p \rangle} \end{aligned} \quad (39)$$

$$\begin{aligned} [L\circ] & \frac{\langle A, 1, \pi_0(t), \sigma, p \rangle \quad \langle B, 1, \pi_1(t), \tau, p \rangle}{\langle A \circ B, 1, t, \sigma + \tau, p \rangle} \\ [R\circ] & \frac{\langle B, 0, t_0, \sigma, p \rangle \quad \langle A, 0, t_1, \tau, p \rangle}{\langle A \circ B, 0, \langle t_0, t_1 \rangle, \sigma + \tau, p \rangle} \end{aligned} \quad (40)$$

Structural modalities

We now extend this treatment of the 'bare' modality to the case of structural modalities, making use of the extra algebraic structure in our structural labelling algebras. Recall that the structural modalities $\Box_i$ are interpreted in an algebra $\langle L, + \rangle$ with subalgebra $\langle L_i, + \rangle$ where the equational theory E_i for the '+' operation explicitly refers to elements of L_i . The proof net rules for a structural modality $\Box_i$ copy the structural label σ unchanged from conclusion to premise, but they install the constraint that the structural label σ is an element of L_i , $L_i \subseteq L$, the subset of L where modal types find their denotation. This information will then determine the solutions to the unification problem $\langle \sigma = \tau \rangle_{E_i}$ for the Axiom links.

$$[L\Box_i] \frac{\langle A, 1, t, \sigma, p \cdot v \rangle}{\langle \Box_i A, 1, t, \sigma, p \rangle} \quad \sigma \in L_i \quad v \in \text{VAR}_W \quad (41)$$

Example

As an illustration we give the LDS derivation of the modalized version of functional composition in the non-associative calculus NL.

$$A/B, B/C \Rightarrow_{\text{NL}} A/\Box_\alpha C \quad (42)$$

The $\Box_\alpha$ modal types are interpreted in a subalgebra $\langle L_\alpha, + \rangle$ of the structural algebra $\langle L, + \rangle$ which in the case at hand is governed by the parametrized form of associativity in (43), cf. [28].

$$E_\alpha: \forall xyz \in L, \forall s \in L_\alpha, x + (y + z) = (x + y) + z \text{ if } s = x \vee s = y \vee s = z \quad (43)$$

The goal sequent assumes the form of the multiset of labelled formulas of (44) for which we have to build a proof net (λ semantics omitted.)

$$\{\langle A/B, 1, s, p \rangle, \langle B/C, 1, s', p \rangle, \langle A/\Box_\alpha C, 0, S, p \rangle\} \quad (44)$$

Formula decomposition yield the following proof frame.

$$\frac{\langle A, 1 \rangle \quad \langle B, 0 \rangle}{\langle A/B, 1 \rangle} \quad \frac{\langle B, 1 \rangle \quad \langle C, 0 \rangle}{\langle B/C, 1 \rangle} \quad \frac{\langle A, 0 \rangle \quad \frac{\langle C, 1 \rangle}{\langle \Box_\alpha C, 1 \rangle}}{\langle A/\Box_\alpha C, 0 \rangle} \quad (45)$$

Below one finds the leaves of this proof frame annotated with structural labels and world labels. The step from $\langle \Box_\alpha C, 1 \rangle$ to $\langle C, 1 \rangle$ installs the constraint that s'' , the structural label associated with $\Box_\alpha C$, belongs to L_α .

	LITERALS	CONSTRAINTS
1.	$\langle A, 1, s + X, p \rangle$	$s'' \in L_\alpha$
2.	$\langle B, 0, X, p \rangle$	
3.	$\langle B, 1, s' + Y, p \rangle$	
4.	$\langle C, 0, Y, p \rangle$	
5.	$\langle A, 0, Z + s'', p \rangle$	
6.	$\langle C, 1, s'', p \cdot P \rangle$	

(46)

The proof frame is turned into a proof net by linking the literals with opposite polarity. To establish the axiom linkings one has to solve the unification problems on the labels — unification under the equational theory E_α , i.e. the parametrized form of associativity of (43).

RESOLVENTS	PROBLEM	SUBSTITUTION
4, 6	$\langle Y = s'' \rangle_{E_\alpha}$	$\{Y \leftarrow s''\}$
2, 3	$\langle X = s' + Y \rangle_{E_\alpha}$	$\{Y \leftarrow s'', X \leftarrow s' + s''\}$
1, 5	$\langle Z + s'' = s + X \rangle_{E_\alpha}$	$\{Y \leftarrow s'', X \leftarrow s' + s'', Z \leftarrow s + s'\}$

(47)

The matching of literals 1 and 5 requires a solution to the problem

$$\langle Z + s'' = s + (s' + s'') \rangle_{E_\alpha}.$$

Notice that with the empty theory E , i.e. without associativity assumptions about the operation '+', there would be no solution.

6 Conclusion

In the LDS proof net approach presented here the computational burden is shifted to efficient unification algorithms for the $\langle s = t \rangle_E$ problems. In the case of 'pure' systems, one can go for optimal data structures that build in the equivalences of E , e.g. lists plus list unification in the case of L (Associativity), bags plus bag unification in the case of LP (Associativity+Commutativity), etc. In the 'mixed' modal systems, the situation is more complex — it is only when we know to what subalgebra a structural label belongs that we can start computing solutions to the $\langle s = t \rangle_E$ problems. This is a first area which invites further research. A second follow-up project for the work outlined in this paper would be to extend the LDS categorical theorem prover of [24], which supports labelled deduction for the distinct logics of $\mathcal{L}$, with an implementation of the structural modalities along the lines of this paper.

References

- [1] Barry, G. & G. Morrill (eds.) (1990) *Studies in Categorical Grammar*. Edinburgh Working Papers in Cognitive Science, Volume 5. CCS, Edinburgh.
- [2] Barry, G. & M. Pickering (1990) 'Dependency and constituency in categorical grammar', in [1].
- [3] Belnap, N.D. (1982) 'Display Logic'. *Journal of Philosophical Logic*, 11, 375–417.
- [4] Belnap, N.D. (1990) 'Linear Logic Displayed'. *Notre Dame Journal of Formal Logic*, 31, 1, 14–25.
- [5] Benthem, J. van (1991a) *Language in Action. Categories, Lambdas, and Dynamic Logic*. Studies in Logic, North-Holland, Amsterdam.
- [6] Benthem, J. van (1991b) 'Logic and the flow of information'. ITLI Prepublication, Amsterdam.
- [7] Bird, S. (1990) *Constraint-based Phonology*. Ph.D. Dissertation, CCS, Edinburgh.
- [8] Buszkowski, W. (1986) 'Completeness results for Lambek systactic calculus'. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 32, 13–28.
- [9] Došen, K. (1988) 'Sequent systems and groupoid models'. *Studia Logica* 47, 353–385 (Part I), and 48, 41–65 (Part II).
- [10] Došen, K. (1990a) 'Modal logic as metalogic'. SNS Report, Tübingen.
- [11] Došen, K. (1990b) 'Modal translations in substructural logics'. Report 10-90, Universität Konstanz.

- [12] Dunn, J.M. (1973) 'A "Gentzen system" for positive relevant implication'. *Journal of Symbolic Logic*, **38**, 356–357.
- [13] Dunn, J.M. (1991) 'Gaggle theory: an abstraction of Galois connections and residuation, with applications to negation, implication, and various logical operators'. In Van Eijck (ed.) *Logics in AI. JELIA Proceedings*. Springer, Berlin.
- [14] Gabbay, D. (1991a) *Labelled Deductive Systems*. Draft. Oxford University Press (to appear).
- [15] Gabbay, D. (1991b) 'A general theory of structured consequence relations'. Draft. To appear in *Theoretical Foundations for Non-Monotonic Reasoning*—Part 3.
- [16] Hepple, M. (1990) *The Grammar and Processing of Order and Dependency: A Categorical Approach*. Ph.D. Dissertation, Edinburgh.
- [17] Hollenberg, M. (1991) 'Proof nets and modalities'. Proceedings 2nd CLIN Conference, Amsterdam (to appear).
- [18] Jakobson, R. (ed.) (1961) *Structure of Language and Its Mathematical Aspects*. Proceedings of the Twelfth Symposium in Applied Mathematics. Providence, Rhode Island.
- [19] Lambek, J. (1958) 'The Mathematics of Sentence Structure', *American Mathematical Monthly* **65**, 154–170.
- [20] Lambek, J. (1961) 'On the Calculus of Syntactic Types', in [18].
- [21] Moortgat, M. (1988) *Categorical Investigations. Logical and Linguistic Aspects of the Lambek Calculus*. GRASS 9, Mouton-De Gruyter, Berlin.
- [22] Moortgat, M. (1990) 'Categorical logics: a computational perspective'. In van de Goor (ed.) *Proceedings Computing Science in the Netherlands 1990*. Mathematisch Centrum, Amsterdam, pp. 329–347.
- [23] Moortgat, M. (1991a) 'Generalized Quantification and Discontinuous type constructors'. In Sijtsma & Van Horck (eds.) *Proceedings Tilburg Symposium on Discontinuous Constituency*. De Gruyter, Berlin.
- [24] Moortgat, M. (1991b) 'Automated deduction in categorical logics'. Lecture notes 3rd European Summer School in Logic, Language and Information. Saarbrücken.
- [25] Moortgat, M. & G. Morrill (1991) 'Heads and phrases. Type calculus for dependency and constituent structure'. ms. OTS, Utrecht. (To appear in *Journal of Logic, Language and Information*.)
- [26] Morrill, G. (1990a) 'Grammar and Logical Types'. In Stokhof & Torenvliet (eds.) *Proceedings 7th Amsterdam Colloquium*. ILLI, Amsterdam.

- [27] Morrill, G. (1990b) 'Intensionality and boundedness'. *Linguistics & Philosophy*, 13, 699–726.
- [28] Morrill, G. (1992) *Type-Logical Grammar*. OTS Working Papers, CL-92-02, Utrecht.
- [29] Oehrle, R.T. (1988) 'Multidimensional compositional functions as a basis for grammatical analysis', in [30], 349–390.
- [30] Oehrle, R.T., E. Bach, D. Wheeler (eds.) (1988), *Categorial Grammars and Natural Language Structures*. Reidel, Dordrecht.
- [31] Pratt, V. (1991) 'Action logic and pure induction', in Van Eijck (ed.) *Logics in AI*. JELIA Proceedings. Springer, Berlin.
- [32] Roorda, D. (1991) *Resource Logics. Proof-theoretical Investigations*. Ph.D. Dissertation, Amsterdam.
- [33] Siekmann, J.H. (1989) 'Unification Theory'. *Journal of Symbolic Computation*, 7, 207–274.
- [34] Troelstra, A. (1990) *Lectures on Linear Logic*. ILLI Prepublications, Amsterdam.
- [35] Wallen, L.A. (1990) *Automated Deduction in Nonclassical Logics*. MIT Press, Cambridge, London.
- [36] Wansing, H. (1990) 'Formulas-as-types for a hierarchy of sublogics of Intuitionistic Propositional Logic', ms. Institut für Philosophie, Freie Universität Berlin.

Constraint-Based Semantics

John Nerbonne*

Deutsches Forschungszentrum für Künstliche Intelligenz

Abstract

Montague's famous characterization of the homomorphic relation between syntax and semantics naturally gives way in computational applications to CONSTRAINT-BASED formulations. This was originally motivated by the close harmony it provides with syntax, which is universally processed in a constraint-based fashion. Employing the same processing discipline in syntax and semantics allows that their processing (and indeed other processing) can be as tightly coupled as one wishes—indeed, there needn't be any fundamental distinction between them at all. In this paper, we point out several advantages of the constraint-based view of semantics processing over standard views. These include (i) the opportunity to incorporate nonsyntactic constraints on semantics, such as those arising from phonology and context; (ii) the opportunity to formulate principles which generalize over syntax and semantics, such as those found in HEAD-DRIVEN PHRASE STRUCTURE GRAMMAR; (iii) a characterization of semantic ambiguity, which in turn provides a framework in which to describe disambiguation, and (iv) the opportunity to underspecify meanings in a way difficult to reconcile with other views. The last point is illustrated with an application to scope ambiguity in which a scheme is developed which underspecifies scope but eschews auxiliary levels of logical form.

1 Introduction

"Standard" natural language algorithms and processing techniques were ill-equipped to deal with feature-based grammars; this difficulty gave rise to "unification-based" formalisms and processing models, which systematize the processing and even the theory of feature-based grammars. The application and development of unification-based approaches quickly demonstrated additional virtues, e.g., the ability to cater to the information shared in linguistic signs, and the ability to exploit partial information. We assume throughout this paper a familiarity with feature-based linguistic description; Shieber 1986 is the fundamental introduction.

Syntax/Semantics interfaces using unification-based or feature-based formalisms may be found in Shieber 1986, Pollard and Sag 1987, Fenstad et al. 1987, and Moore 1989. The primary reason for specifying a syntax/semantics interface in feature structures is that it harmonizes so well with the way in which syntax is now normally described; this close harmony means that syntactic and semantic processing (and indeed other processing, see below) can be as tightly coupled as one wishes—indeed, there needn't be any fundamental distinction between them at all. In feature-based formalisms, the structure shared among syntactic and semantic values constitutes the interface in the only sense in which this exists.

1.1 A Simple Illustration

The fundamental idea is simply to use feature structures to represent semantics. If one wishes to compute the semantics of a sentence such as *Sam walks*, one first defines a feature SEMANTICS which must be lexically provided for in the case of *Sam* and *walks*, and which

*Thanks to Bob Carpenter, Kris Halvorsen, Bob Kasper, Uli Krieger, Joachim Laubsch and Carl Pollard for discussion of the ideas presented here. This work was partially supported by a research grant, ITW 9002 0, from the German Bundesministerium für Forschung und Technologie to the DFKI DISCO project.

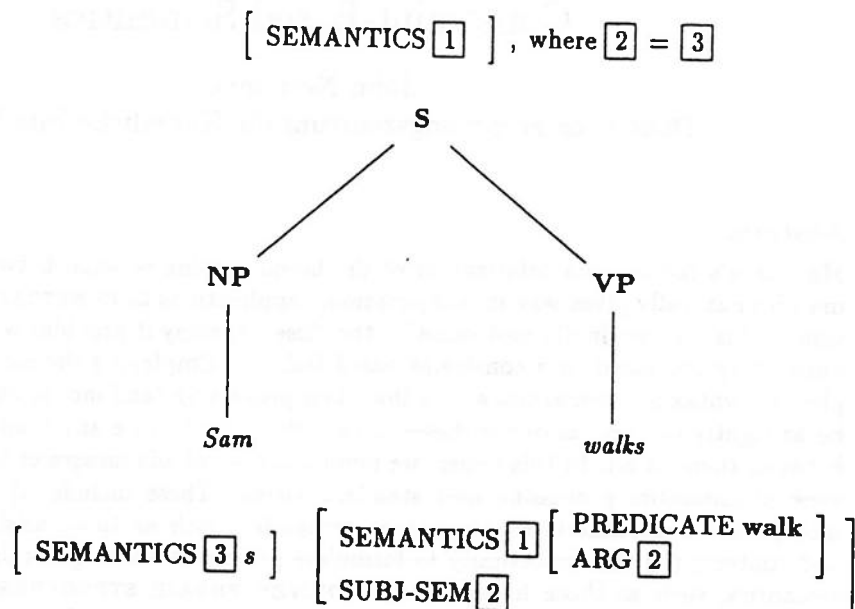


Figure 1: A sketch of the semantic derivation of *Sam walks*, *walk(s)* as this would proceed using unification. Unification applies to syntactic and semantic representations alike, eliminating the need to compute these in distinct processes, and unification is employed to bind variables to argument positions, eliminating the need for (a great deal of) β -reduction as used in schemes derived from Montague and the lambda calculus.

can be computed from these (together with other information) in the case of the sentence. Figure 1 provides an illustration of how this works.

This paper adduces several advantages of a constraint-based view of the syntax/semantics relation over standard views, including (i) the opportunity to incorporate nonsyntactic constraints on semantics, such as those arising from phonology or context; (ii) the opportunity to formulate principles which generalize over syntax and semantics, such as those found in HEAD-DRIVEN PHRASE STRUCTURE GRAMMAR (Pollard and Sag 1987). We then argue that feature-based semantics ought to be understood as specifying constraints on logical forms, not model structures. The virtues of viewing semantics processing as manipulation of logical form descriptions includes not only a characterization of semantic ambiguity, which in turn provides a framework in which to describe disambiguation, but also the opportunity to underspecify meanings in a way difficult to reconcile with other views. The latter point is illustrated with an application to the notorious scope ambiguity problem.

2 Constraint-Based Semantics

There are several advantages of the unification-based view of the syntax/semantics interface over the more familiar (Montogovian) view of this interface, which is characterized by a homomorphism from syntax into semantics. The constraint-based view sees the interface as characterized by a set of constraints to which nonsyntactic information may contribute, including phonological and pragmatic information. We consider these in Sections 2.2 and 2.3 below, and Fenstad et al. 1987, pp.12-17 discusses the distinction between the constraint-based and the homomorphic views of the relation between syntax and semantics from processing and other more general points of view.

We first examine the opportunity for expressing syntax/semantics generalizations which

the constraint-based view provides, and then turn to the semantics of sentence accent and that of deixis as examples which indicate that there should be no homomorphism from syntax (narrowly understood) to semantics. In its place we argue that a constraint-based view is preferable.

2.1 Syntax/Semantics Generalizations

The constraint-based of the syntax/semantics relationship allows syntactic and semantic information to be bundled in complex, but useful ways. This contrasts with the Montgagovian view which sees them as domain and range of a homomorphism—objects of potentially very different kinds. The opportunity to bundle information in complex ways allows the very simple statements of syntax/semantics relationships that make HPSG attractive (e.g. the SUBCATEGORIZATION PRINCIPLE (Pollard and Sag 1987, p.71) ultimately responsible for the semantics computation shown in Figure 1).

To see this, let us provide a more general set of specifications for the example in Figure 1. We may begin by noting that verb phrases not only allow reference to their subject semantics, they also may impose restrictions on their subjects (e.g., that they agree with the head of the VP). This leads us to a specification in which not only SUBJ-SEM is specified on the VP, but more generally SUBJECT, which in turn specifies potentially both syntax and semantics. But we note that SUBJECT is only one of several potential complements, each of which could be specified in this fashion, leading us to a view of the VP as follows:

$$\left[\begin{array}{l} \text{SEMANTICS } \boxed{1} \left[\begin{array}{l} \text{PREDICATE walk} \\ \text{ARG } \boxed{2} \end{array} \right] \\ \text{COMPLEMENTS } \langle \left[\begin{array}{l} \text{SEMANTICS } \boxed{2} \\ \text{SYN|AGR } 3s \end{array} \right] \rangle \end{array} \right]$$

But the sense of this sort of specification is that it contributes to the definition of well-formed phrases, in this case by specifying the complements with which a head may combine (as in categorial grammar). This is provided by a restriction on the complement-head phrasal type, the SUBCATEGORIZATION PRINCIPLE:

$$\left[\begin{array}{l} \text{SEMANTICS } \boxed{1} \\ \text{DAUGHTERS } \left[\begin{array}{l} \text{HEAD-DTR } \left[\begin{array}{l} \text{SEMANTICS } \boxed{1} \\ \text{COMPLEMENTS } \langle \boxed{2}, \dots \rangle \end{array} \right] \\ \text{COMPLEMENT-DTR } \boxed{2} \end{array} \right] \end{array} \right]$$

Taken together (and assuming an NP specification of the sort provided in Figure 1), these specifications guarantee that the correct semantics (for a large subset of examples) are provided. What is striking here is that no special semantics principle of combination is required—the subcategorization principle effects a unification of a complement daughter with a head's complement specifications. But since semantics are among these, the principle has as a consequence that the argument position corresponding to the complement semantics is bound.

Before concluding this section, we should note that my presentation of the subcategorization principle abstracts away from an essential part, in that the actual subcategorization principle also accounts for the COMPLEMENTS specifications of phrases as these are derived from the phrase's components—in much the same fashion as categorial grammar. But this is merely additional, not conflicting information. The interested reader may consult Pollard and Sag 1987, p.87. Pollard 1989 investigates a more speculative interweaving of syntax and semantics.

2.2 Semantics/Phonology

Focus-sensitive adverbs such as *only* depend for their interpretation on phonological sentence accent, and not only on syntactic structure. (I should like to emphasize that the point is NOT that syntax plays no role in the interpretation of focus, since it surely does; e.g., when it is noted that phonological sentence accent must be realized within a focused constituent—i.e., within a syntactically defined unit, the definitions depend on syntax. The point is merely that the phonology, too, requires semantic interpretation.)

A treatment of the semantics of focus which provides correct semantics for sentences such those in (1) is found in Rooth 1985:

- (1) a. Tom only introduced BILL to Sue.
b. Tom only introduced Bill to SUE.

(where uppercase is used to designate accented words.) Rooth's treatment assigns a single semantics—independently of accent—to the VP *introduce Bill to Sue* in both (1a) and (1b), i.e.

$$\lambda x.\text{introduce}(x, b, s)$$

The semantics of the particle *only*, used here adverbially, applies to this property to derive another whose intuitive content is just the property of individuals who have no "contextually" relevant property other than that denoted by the VP:

$$\lambda y(\forall P[P(y) \wedge C(P) \rightarrow P = \lambda x.\text{introduce}(x, b, s)])$$

Thus Rooth's treatment relies on a specification of the ALTERNATIVES to a given assertion to provide a semantics for focus.

The restriction to contextually relevant properties solves an important technical difficulty which arises when one tries to specify that an individual has only a single property: we cannot simply ignore properties of self-identity, etc. $\lambda x.x = x$. Failing this restriction, the specification in question *only introduce BILL to Sue* could only be satisfied in counterintuitive models—those with a single element, denoted by each of *Tom*, *Bill* and *Sue*.

The specification of 'C', the "contextually relevant" properties, is delimited by focus (sentence accent). For example, the properties required in the case of (1a) are defined as follows:

$$C = \{P | \exists x P = \lambda y.\text{introduce}(y, x, s)\}$$

This stands today as one of relatively few successful treatments of focus phenomena.¹ But one can ask whether it would not be intuitively preferable to derive the meanings required not via functions whose sole input is the meanings of phrases, but rather via constraints based on both phrase meanings and phonological properties. In that case, we should postulate that the VP *introduce BILL to Sue* has following semantic and phonological properties:

$$\left[\begin{array}{l} VP \\ \text{SEMANTICS } \text{introduce}(x, y, \text{sue}) \\ \text{PHONOLOGY|FOCUS|SEMANTICS} \left[\begin{array}{l} \text{PLACEHOLDER } y \\ \text{CONTENT } \text{bill} \end{array} \right] \end{array} \right]$$

In this we ascribe semantic properties to phonological entities, but this is plausible given their ultimate source in sentence accent. In addition, we require a lexical specification of *only*:

¹ An interesting recent alternative is provided by Krifka 1991, building on von Stechow 1989. As will become apparent, my point here is not to argue for Rooth's analysis, but rather to show how a similar analysis fits rather nicely into a constraint-based view of semantics. So I will not try to justify using Rooth's work rather than Krifka's except to say that Rooth's is earlier and better known.

$$\left[\begin{array}{l} Adv \\ SEMANTICS \vee \boxed{1}(\boxed{3} \leftrightarrow \boxed{1} = \boxed{3}) \\ MOD|PHONOLOGY|FOCUS|SEMANTICS| \left[\begin{array}{l} PLACEHOLDER \boxed{1} \\ CONTENT \boxed{2} \end{array} \right] \end{array} \right]$$

Given two general principles about the interpretation of adjuncts, viz. (i) that the semantics of adjunct head combinations is derived from the adjunct, and (ii) that the adjunct's MOD specifications must unify with the head, these specifications yield the result that the VP *only introduce BILL to Sue* means:

$$\forall y(\text{introduce}(x, y, \text{sue}) \leftrightarrow y = \text{bill})$$

(We do not specify the VP meaning as a λ -abstract since we do not rely on function application to derive meanings.) This is equivalent in all relevant respects to Rooth's analysis except in that it identifies the source of focus as phonological rather than syntactic.

2.3 Semantics/Context

In this section we would like to show that a constraint-based approach is preferable to a homomorphic one because it provides a natural modus for the expression of constraints arising from context.

A number of contemporary theories of meaning are radically noncompositional with respect to the role played by discourse context. In these theories semantics is NOT effected via a homomorphism from syntax alone, but rather via a two-place mapping from syntax and context (or pragmatics). Situation semantics (Barwise and Perry 1983, Gawron and Peters 1990) is merely a stricter codification of prevalent views in this respect. Noncompositionality is most evident in the analysis of nonquantified noun phrases in Gawron and Peters 1990, which has its roots in the analysis of "singular noun phrases" in Barwise and Perry 1983. The semantic contribution a nonquantified NP makes is simply an individual to which the NP refers—but the determination of WHICH individual depends on the context in which the NP is uttered, and not only on the meanings of the NP's syntactic constituents.

This may be illustrated using deictic expressions. Gawron and Peters might analyze an utterance of *That dog walks* as having the logical form $\text{walk}(x)$, and as importing the contextual restrictions that x denotes only if it holds both that $\text{dog}(x)$ and $\text{REFREL}(\text{that}, x)$. The latter condition obtains when an utterance of *that* is used to refer to the object assigned to x . Of course, if either of these conditions is not met, then the utterance of the expression *that dog* fails to denote, and no assertion is made.

For our programmatic argument the important point is not merely that the context of utterance plays a role in determining the semantic contribution of a phrase (surely a trivial property of the semantics of deictics), but also that this contextual dependence is not a part of the semantic contribution made by deictics, i.e., that there are apparently no semantic contexts which use or modify the contextual dependency in anyway. The deictics are "rigid designators", whose denotation exhausts their semantic contribution.

Gawron and Peters introduce RESTRICTED PARAMETERS as a technical aid in explicating the context dependence of nonquantificational noun phrases. Restricted parameters are logical terms of the form ' $(x|\phi)$ ' for x an individual variable, ϕ a condition (state-of-affairs), which denote the individual assigned to x in case ϕ holds, and which otherwise fail to denote. This looks like a perfectly compositional (partial) definition, except that the conditions involved may involve conditions on context, e.g., 'REFREL'. But we may highlight the importance of constraint-based view more effectively by providing a formulation in the

feature specification language employed here throughout. A candidate for the right sort of representation (for our example *That dog walks*) is provided:

$$\left[\begin{array}{c} \text{sign} \\ \text{SEMANTICS} \left[\begin{array}{c} \text{CONTENT walk}(x) \\ \text{CONTEXT REFREL}(\text{that}, x) \wedge \text{dog}(x) \end{array} \right] \end{array} \right]$$

According to the Gawron and Peters view, the usual recursive definition of SEMANTICS (based on DAUGHTER|SEMANTICS and SYNTAX) would give way to simultaneous recursive definitions of SEMANTICS|CONTENT and SEMANTICS|CONTEXT (based on DAUGHTER|SEMANTICS|CONTENT, DAUGHTER|SEMANTICS|CONTEXT and SYNTAX).

There are of course several responses available to adherents of compositionality. The most frequent response is to reinterpret meanings so that contextual dependency is inherent in them. (We have in mind the work on context in Kaplan 1979.) This saves compositionality at the expense of weakening it, perhaps completely, since context is a very broad concept.

Important for our present purposes is the fact that there are theories of semantic representation according to which semantics is noncompositional, and that the constraint-based view accommodates these comfortably.

3 The Nature of Constraint-Based Semantics

The usefulness of the feature-based view of semantics temporarily postpones the foundational question of the nature of the semantic representations being employed. The postponement is only temporary, however, since an answer to the question is crucial to several issues in the practical use of these representations:

- How are quantifiers, and more generally, higher-order operators, to be represented?
- How can lambda expressions be represented? If they are inexpressible, how can one treat those areas where lambda expressions seem to be employed crucially—type-raising (*The CEO and every division manager left*), VP and common noun phrase ellipsis (*Tom read every book before Sam did* and *A good big man will beat a small one*) and predicative traces (*How tall is Sam?*, *How tall did Al say that Sam is?*)?
- There exist standard model theories for feature formalisms (of somewhat different sorts—cf. Kasper and Rounds 1986; Smolka 1988 and Johnson 1988; Reape 1991 and Blackburn 1991 and Carpenter et al. 1991) Are these sufficient for the use of feature structures to represent natural language semantics?

There is a simplest view of the nature of feature-based semantic representations, which is that they may be interpreted as directly denoting elements within a semantic model. I shall contrast this with the view that feature structures are better used to describe, not model structures directly, but rather logical forms (which in turn denote elements within the model structure). Figure 2 provides a graphic rendition of the second, more complicated view, which shall be defended here.

There are several problems with the simplest view. First, feature structure formalisms normally make no provision for quantifiers or other higher-order operators. Second, there seems to be no way of defining a lambda operator in these formalisms. See Moore 1989 for a discussion of the difficulties this entails in the case of type-raising. Third (and generalizing over the last two points), a semantic representation language must be extremely powerful if it is to represent the range of the possible meanings in natural language. The needs of a feature formalism which excludes semantics are much more modest, which allows the formalism to retain better computational properties. If we tried to represent semantics directly in

feature structures, we would be left with a representation scheme so rich that its computational properties could not be attractive. It is worth noting that several research strategies deal with this difficulty either by avoiding it—concentrating on applications which do not exercise the richer expressive capabilities of natural language, or by ignoring it—employing nonlogical semantic representation schemes whose expressive and computational properties are not understood exactly. The latter tack suffers in many obvious ways, but particularly when it comes to attempting inference on any but an *ad hoc* basis. The former tack—seeking applications with reduced semantics demands—is certainly defensible, but very difficult to maintain because semantic complexity can arise in unexpected places. A particular problem with this strategy arises in interface applications, where the “flexibility” of natural language interfaces is adduced as a justification for their development. Flexibility invariably requires rich expressive means. The present proposal does NOT claim to banish computational complexity from linguistic formalisms—but it does shield the feature formalism from the complexity (incompleteness) inherent in semantic representation schemes.

Finally, I would like to demonstrate a conceptual problem which the metalinguistic view solves neatly, but which is quite puzzling on the direct denotation view. This concerns the distinction between vagueness and ambiguity, and will be introduced by an example. The noun *bank* is ambiguous between the ‘\$-home’ and ‘riverside’ meanings in a way that *glove* is not, even though *glove* can refer only to left-hand gloves or right-hand gloves—quite distinct categories of objects. This distinction is crucial in quantification and anaphora. Thus *three gloves* ignores the distinction between left- and right-hand gloves in a way that *three banks* may not (this cannot refer to a pair of financial institutions and a riverside); similarly *Sam bought a bank* and *Bob sold one* cannot describe a situation involving a financial institution on the one hand and a riverside on the other; i.e., it allows only a pair of the four possible

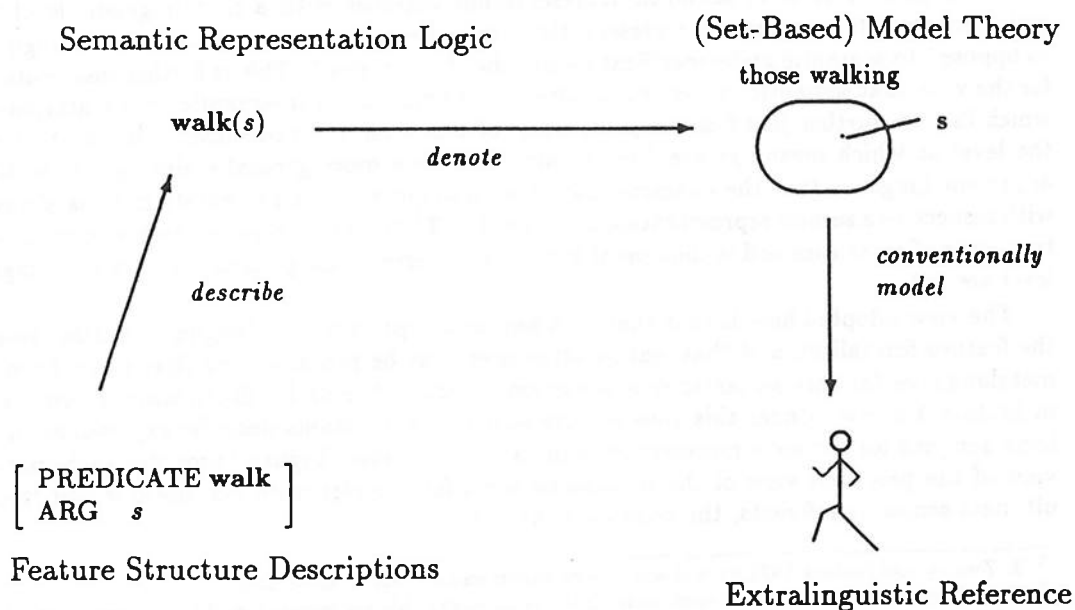


Figure 2: The relation of feature-based semantics to semantic representation logic, model theory, and the extralinguistic objects of discourse. A simpler, but ultimately unsatisfactory view, would eliminate the semantic representation logic.

combinations of the two meanings. The vague (or underspecified *glove*) is less restrictive; any of the four possible combinations of meanings can be at play in the following example:²

Sam lost a glove and Bob found one

If we allow that semantics makes metalinguistic specifications, then it specifies that 'bank' has the meaning [PRED {\$-HOME,RIVERSIDE}], using metalinguistic disjunction. Under the (quite reasonable) assumption that there is no single object-language predicate denoted by this disjunction, we can explain the quantificational and anaphoric facts. *Glove*, on the other hand, is unambiguous, even if its semantics may be equivalent to an object-language disjunction: [PRED *glove*], where the object-language disjunction holds:

$$\text{glove}(x) \leftrightarrow \text{left-glove}(x) \vee \text{right-glove}(x)$$

The standard solution here is to postulate distinct lexical items for ambiguity, but this solution essentially denies purely semantic ambiguity, reducing it to lexical ambiguity, and it fails to generalize to the view of the lexicon as a disjunction of words, common in the feature-based theories (cf. HPSG, Pollard and Sag 1987 p.147). Under the view of the lexicon as a disjunction of words (or word descriptions), the postulation of lexical items distinct only in semantics reduces to the postulation of a single item with disjunctive semantics, since:

$$(p_1 \wedge \dots \wedge p_n \wedge q) \vee (p_1 \wedge \dots \wedge p_n \wedge r) \leftrightarrow (p_1 \wedge \dots \wedge p_n \wedge (q \vee r))$$

The solution offered here is consistent with the view that multiple lexical entries are involved, but it immediately suggests a more perspicuous representation (in analogy to the right-hand side of the biconditional); it differs only in insisting that there be two levels of semantics: a level which constrains semantic representations, and the level of semantic representations themselves. Lexical ambiguity involves underspecification at the first level; lexical vagueness is underspecification at the second.³

Thus, given a level of semantic representation together with a metalinguistic level at which constraints on semantic representation are expressed, a notion of semantic ambiguity as opposed to semantic underspecification may be characterized. This is further justification for the view that semantic processing involves the manipulation of semantic representation—which in turn further justifies the postulation of this level of representation in addition to the level at which meanings are directly modeled. The more general claim on which this argument hinges is that the characterization of ambiguity in a representation L_1 is always with respect to a second representational system L_2 . The scheme proposed here distinguishes two levels of semantics and is thus capable of characterizing ambiguity; systems with a single level are not.

The view adopted here is that there is a semantic representation language, distinct from the feature formalism, and that feature structures may be profitably employed as a formal metalanguage for this semantic representation scheme. A lengthy illustration is provided in Section 4 below. Under this view feature structure expressions describe expressions in a logic designed for semantic representation in natural language. Figure 2 provides a schematic view of the proposed view of the relation between feature structure descriptions and their ultimate semantic referents, the objects spoken of.

²Cf. Zwicky and Sadock 1975 for a discussion of anaphora as a test of ambiguity.

³While there is not space here to anticipate all the reactions to this argument, I would like to note that the quantification facts show that this is not an issue of semantic grain, in the sense in which this is debated, e.g., in situation semantics and possible worlds semantics. Cf. Barwise and Perry 1983, Barwise 1989. Thus it will not do simply to point to logics in which there may be a relation of material equivalence, but no relation of logical equivalence between the left and right sides above. This is so because natural language quantification is insensitive to distinctions finer than material equivalence.

The Section 4 illustrates in more detail how a semantics conceived along these lines functions.

4 An Illustration

We illustrate the view that semantics involves the accumulation of constraints expressed about a semantic representation language, by applying the view to a well-known semantic representation language, the language of generalized quantifiers; second, by developing its metalanguage within the feature description language used in HPSG; and third by demonstrating its application to problems of scope.

4.1 Logic—A Generalized Quantifier Language

To illustrate the approach we shall first need a target semantic representation language. Here we deliberately use a popular semantic representation scheme—a version of the language of generalized quantifiers; this is a kind of *lingua franca* among theoretically oriented computational linguists. We emphasize that the overall scheme—that of employing feature structure descriptions as a formalized metalanguage for a semantic representation logic—is general, and could easily be applied to other logics, e.g., first order logic, higher-order or intensional logics, discourse representation structures, or a language of situation theory. It could even be applied to nonlogical representations, but that would be harder to motivate.

A BNF for a Language of Generalized Quantifiers—LGQ

$\langle \text{var} \rangle ::= x_0 \mid x_1 \mid \dots$	
$\langle \text{const} \rangle ::= c_0 \mid c_1 \mid \dots$	$\langle \text{det} \rangle ::= \forall \mid \exists \mid \text{MOST} \mid \dots$
$\langle \text{pred} \rangle ::= P_0 \mid P_1 \mid \dots$	$\langle \text{g-quant} \rangle ::= \langle \text{det} \rangle \langle \text{var} \rangle \langle \text{wff} \rangle$
$\langle \text{term} \rangle ::= \langle \text{var} \rangle \mid \langle \text{const} \rangle$	$\langle \text{wff} \rangle ::=$
$\langle \text{atomic-wff} \rangle ::= (\langle \text{pred} \rangle \langle \text{term} \rangle^*)$	$\langle \text{atomic-wff} \rangle$
	$\mid (\langle \text{wff} \rangle \langle \text{conn} \rangle \langle \text{wff} \rangle)$
	$\mid (\neg \langle \text{wff} \rangle)$
	$\mid (\langle \text{wff} \rangle \rightarrow \langle \text{wff} \rangle)$
	$\mid (\langle \text{g-quant} \rangle \langle \text{wff} \rangle)$
$\langle \text{conn} \rangle ::= \wedge \mid \vee \mid \leftrightarrow$	

There is one detail about this syntax definition which may seem peculiar to non-computational semanticists. The definition foresees quantified formulas not in the Barwise-Cooper notation (cf. Barwise and Cooper 1981) where the quantified formula normally had λ -terms in the restrictor and scope, but rather in the notation more frequently used in computational linguistics, in which these positions are filled by open formulas. The formula below highlights the difference:

Barwise and Cooper Notation	Present Notation
$\forall x (\lambda y. \text{man}(y), \lambda z. \text{mortal}(z))$	$(\forall x \text{ man}(x) \text{ mortal}(x))$

This is generally preferred in order to keep the visual complexity of formulas (the number of λ 's) at a minimum. Cf. Moore 1981 for an early use; and Dalrymple et al. 1991, pp.414-17 for model-theoretic definitions of the alternative forms.

We turn now to the description of expressions in this language using a typed feature description language of the sort common in grammar processing.

4.2 Metalogic—AVM Specifications for LGQ

In this section we employ typed feature logic (often referred to as ATTRIBUTE-VALUE MATRICES or AVMs) to provide a set of type definitions for expressions in the logical language just presented. We shall not present the typed feature logic formally, relying on

(Carpenter to appear 1992) for definitions. The initial specifications are limited to very vanilla-flavored uses of the feature description scheme, and we shall warn when more particular assumptions are made (Section 5).

$\langle \text{var} \rangle ::= x_0 \mid x_1 \mid \dots$	$\left[\begin{array}{l} \text{var} \\ \text{INDEX} \end{array} \right]$
$\langle \text{atomic-wff} \rangle ::= (\langle \text{pred} \rangle \langle \text{term} \rangle^*)$	$\left[\begin{array}{l} \text{atomic-wff} \\ \text{PRED } [\text{pred}] \\ \text{FIRST } [\text{term}] \\ \vdots \\ \text{N-TH } [\text{term}] \end{array} \right]$
$\langle \text{g-quant} \rangle ::= \langle \text{det} \rangle \langle \text{var} \rangle \langle \text{wff} \rangle$	$\left[\begin{array}{l} \text{gen-quant} \\ \text{DET } [\text{det}] \\ \text{VAR } [\text{var}] \\ \text{REST } [\text{wff}] \end{array} \right]$
$\langle \text{wff} \rangle ::= \dots$ $\quad \mid (\langle \text{wff} \rangle \langle \text{conn} \rangle \langle \text{wff} \rangle)$	$\left[\begin{array}{l} \text{connective-wff} \\ \text{CONN } [\text{conn}] \\ \text{WFF1 } [\text{wff}] \\ \text{WFF2 } [\text{wff}] \end{array} \right]$
$\dots \mid (\neg \langle \text{wff} \rangle)$	$\left[\begin{array}{l} \text{negation} \\ \text{SCOPE-WFF } [\text{wff}] \end{array} \right]$
$\dots \mid (\langle \text{wff} \rangle \rightarrow \langle \text{wff} \rangle)$	$\left[\begin{array}{l} \text{implication} \\ \text{ANTE } [\text{wff}] \\ \text{CONSEQ } [\text{wff}] \end{array} \right]$
$\dots \mid (\langle \text{g-quant} \rangle \langle \text{wff} \rangle)$	$\left[\begin{array}{l} \text{q-wff} \\ \text{QUANT } [\text{gen-quant}] \\ \text{SCOPE } [\text{wff}] \end{array} \right]$

We shall make use of TYPE predicates, e.g., $\text{var}(x)$, which holds iff x is of type var . It is easy to note the absence of several expression types from the set of feature-structure types defined; for example, terms, predicate and individual constants have not been defined here. That is because we rely on TYPE information for distinctions which are not realized in one or more distinct attributes. Figure 3 illustrates the type hierarchy we assume. Of course, we can express the type hierarchy in the language of typed feature descriptions—and in this way obtain a specification fully equivalent to the BNF. For example, in Carpenter to appear 1992 we can specify that

$$\text{term}(x) \leftrightarrow \text{var}(x) \vee \text{const}(x)$$

For the purposes of this paper, we may rely on the informal presentation in Figure 3.

4.3 Use of Semantic Descriptions—Examples

Some simple examples of the kinds of metalinguistic specifications allowed are illustrated in Figure 4. These would be compiled in ways suggested by Figure 1; i.e., we imagine that construction principles (or grammatical rules) may have a semantic correlate which constrains

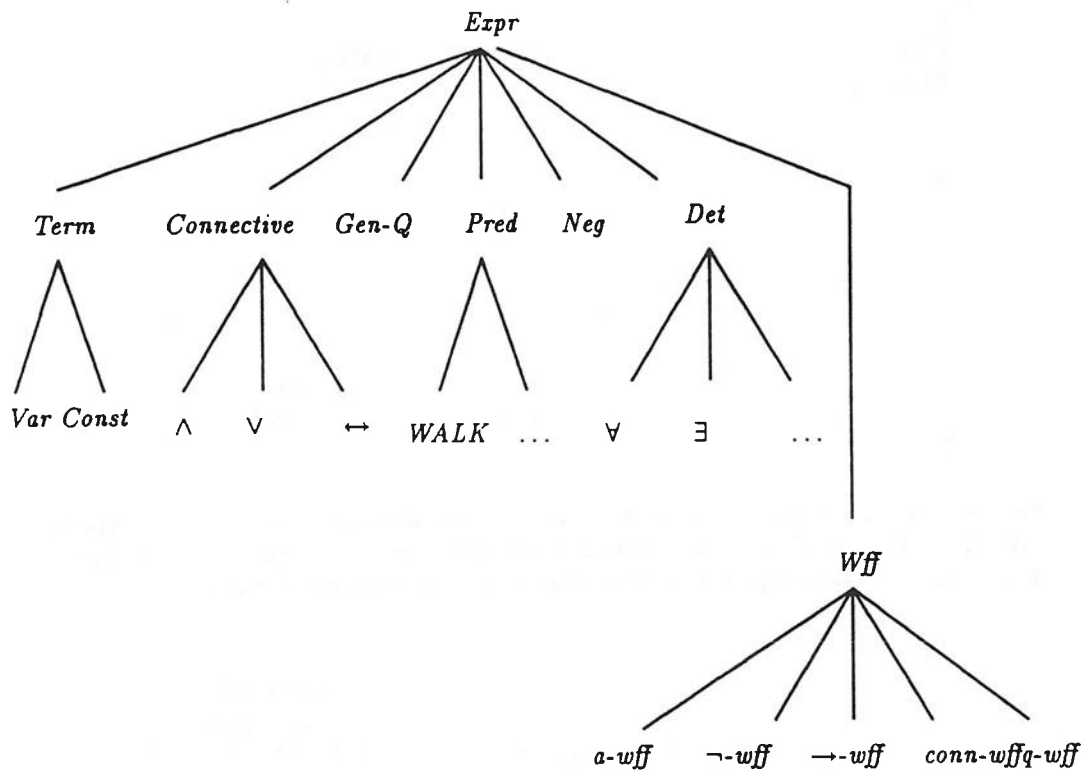


Figure 3: A type hierarchy for the domain of expressions in the language of generalized quantifiers as defined in the text. Carpenter 1992 provides the theory of typed feature structures within which a hierarchy such as this functions.

the semantic representation which is the meaning of the construction. Of course, as we noted above, there is no reason that only syntax should have the privilege of constraining meaning.

The use of feature structures as a metalinguistic level of semantic representation allows greater freedom in semantic explanations. This may be illustrated with respect to the representation of argument positions in atomic formulas and their specifications in the current scheme. The semantic representation language LGQ uses order-coding to represent which arguments are bound to which argument positions, while the metalanguage (feature structure descriptions) identifies this using features (FIRST, etc.). The first bit of freedom we might exercise concerns the identification of argument positions. Nothing would stand in the way of using more contentful-sounding role names to pick these out. For example, we might alter the feature specifications in such a way as to allow the following:

<i>atomic-wff</i> PRED send SOURCE <i>x</i> THEME <i>y</i> GOAL <i>z</i>
--

In this case the simplest generalization would appear to be that allowing any name to designate a semantic role. We would again represent roles as features, postponing any more detailed representation until it is motivated:

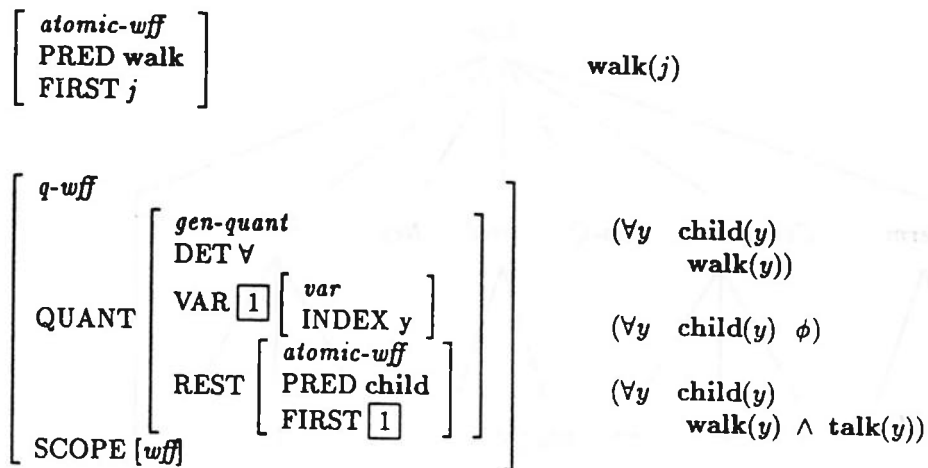


Figure 4: Feature structure descriptions used as metalinguistic descriptors of expressions in LGQ. Note the use of underspecification in the last example. The underspecified description is compatible with many, including semantically contradictory formulas.

$$\langle \textit{atomic-wff} \rangle ::= (\langle \textit{pred} \rangle \langle \textit{pair} \rangle^*)$$

$$\left[\begin{array}{l} \textit{atomic-wff} \\ \text{PRED } [\textit{pred}] \\ \text{ROLE}_1 [\textit{term}] \\ \vdots \\ \text{ROLE}_n [\textit{term}] \end{array} \right]$$

Although we will not make use of this representation below, it illustrates a degree of freedom allowed by the present semantic representation scheme but absent from simpler ones. It would moreover appear to suffice for all the syntactic purposes for which so-called thematic roles are deployed (cf. Jackendoff 1972, 29-46; Roberts 1991; Dowty 1991; Wechsler 1991).

On the other hand, there are purely semantic grounds for preferring keyword-coding to order-coding as a way of identifying argument positions. Two such reasons often adduced are (i) that the ORDER of arguments in relations is never used semantically, and thus that every alternative ordering leads to a perfectly equivalent logic, so that the keyword-coding suffices; and (ii) that the use of keyword coding allows us to make sense of anadic predication (i.e., using the same predicate with a variable number of arguments). On the latter point, see Creary and Pollard 1985.

5 A Richer Constraint Language

The use of feature descriptions above (ignoring the use of types) is of the fairly simple sort common to all feature systems, including, e.g., PATR-II (cf. Shieber et al. 1983). In general, this was conceived as a theory of linguistic categories (Gazdar et al. 1987), which complemented a “backbone” of context-free phrase-structure rules. Initially, this division of labor—feature-structures on the one hand and CF rules on the other—appeared justified by the need for recursion in the latter, but not in the former. But the treatment of long-distance dependence necessitates the use of some recursion even in the feature-structure component (cf. Kaplan and Zaenen 1988 and Kaplan and Maxwell 1988). In continuing below our investigations in constraint-based semantics, we shall employ the language of HPSG (Pollard and Sag 1987) because HPSG proceeds from the richer view of feature structure

description languages. Indeed, HPSG may be seen as the formulation of the feature-based work followed to its logical conclusion: here, feature structures provide not merely a theory of linguistic categories, but rather an alternative foundation for all linguistic theory. We rely on Carpenter to appear 1992 for formal development of the HPSG feature description language, consisting of feature terms to which additional CONSTRAINTS may apply, formulated using mechanisms such as:

types Cf. Carpenter to appear 1992. As above, we shall assume Carpenter's well-typing, i.e., types restrict which features are appropriate on their instances, and features restrict their values to appropriate types.

relations Devices for describing not only feature structures (attribute-value matrices), but also relations between them. An example commonly cited to show the need for specifying relations is the three-place relation *append*:

(2) $\text{append}(a, b, c) \leftrightarrow a = \Lambda \text{ and } b = c \quad \text{OR}$

(3) $a|\text{FIRST} = c|\text{FIRST} \text{ and } \text{append}(a|\text{REST}, b, c|\text{REST})$

where ' Λ ' denotes the empty element, and ' $\langle \text{value} \rangle | \text{path} \rangle$ ' is used to denote the value of $\langle \text{path} \rangle$ wrt $\langle \text{value} \rangle$. Lists are needed to describe linguistic phenomena which yield naturally to analyses as ordered sets, e.g., the daughters of a phrasal node or the subcategorization specifications of a head.

recursive type specifications These are type specifications on a type t which require an element of t' as value to a given attribute, where $t \subseteq t'$. For example, an unsaturated sign is defined in HPSG as one with a nonempty SUBCAT—where SUBCAT is restricted to being a list of signs.

$$\left[\begin{array}{l} \text{unsaturated-sign} \\ \text{SYN|LOC|SUBCAT } \langle [\text{sign}], \dots \rangle \end{array} \right]$$

Similarly, the type *list* may be defined recursively (we assume [FIRST, REST] form) as either the empty list or a structure in which the feature REST is restricted to be of type *list*. A further example is the definition of *tree* as a *lexical-sign* or a *sign* whose attribute DAUGHTERS is a list of *tree*'s. We employ recursive type specifications for the representation of scopally underdetermined relationships.

Of course, recursion is not limited to the simple case where an immediate attribute of t requires a value of $t' \supseteq t$. It suffices for any path $\text{ATTR}_1 | \dots | \text{ATTR}_n$ to impose such a restriction.

sets The motivation and technical development is provided in Pollard and Moshier 1990. One aspect of Pollard and Moshier's treatment is particularly significant below: a description set of n element descriptions can never hold of sets of $n + m$ elements, but it can hold of sets of $n - m$ individuals—this is possible where some of the element descriptions are compatible. (Rounds 1988 provides alternative developments of the notion 'set' in feature logic.) We employ the Pollard & Moshier notation of $\{ * \dots * \}$ as set delimiters.

6 Scope Underspecification

A fundamental problem in computational semantics is the support of DISAMBIGUATION, i.e., the resolution of semantic ambiguity and underspecification. Applications of feature-based techniques for predicate disambiguation is found in Moens et al. 1989 and Nerbonne 1991. Scope disambiguation is a more difficult problem both because the relevant factors are less well understood, and because the underspecified information is more complex. In this section we investigate the use of constraints which underspecify scope. To provide the flavor of the

system proposed, we first provide the (relatively simple) specifications in feature-description language for two common metalogical tasks—the definition of free variable, and that of closed formula.

6.1 Metalogical Task—Definition of Free Variables

We add to each type a feature **FREE-VARS**, which is constrained by type definitions. The feature specification mimics the usual recursive definition (cf. Ebbinghaus et al. 1978, p.30). A variable (occurrence) is free in atomic formulas, and free in boolean combinations under the obvious conditions, and finally free in quantified formulas where it is free in a component and not bound by the quantifier itself. We suppress the full definition (found in Nerbonne 1992) in the interest of saving space, and provide one sample clause:

$$\left[\begin{array}{l} \text{gen-quant} \\ \text{DET} \\ \text{VAR } \boxed{2} \\ \text{REST} \\ \text{FREE-VARS } \boxed{1} \end{array} \right] \quad \text{Condition: } x \in \boxed{1} \text{ iff } x \in \text{REST} \setminus \text{FREE-VARS} \text{ and } x \neq \boxed{2}$$

Given the definition of free variables, the definition of closed formula is straightforward:

$$\text{closed-wff} = \left[\begin{array}{l} \text{wff} \\ \text{FREE-VARS } \{ * \} \end{array} \right]$$

6.2 Overspecifying Logical Forms

Note that the means of defining free variables (above) depends on adding superfluous information to metalogical specifications. The **FREE-VARS** attribute provides information which is useful, but calculable from other information present—in the case it describes well-formed formulas, the feature is redundant. We exploit this possibility in what follows.

An interesting opportunity arises when we consider slightly less obvious descriptions of logical formulas. Suppose, for example, that the type *var* contains a field, not only for the variable name, but also for its scope, i.e., the quantified formula whose quantifier binds the variable:

$$\langle \text{var} \rangle ::= x_0 | x_1 | \dots \left[\begin{array}{l} \text{var} \\ \text{INDEX} \\ \text{BINDER } [q\text{-wff}] \end{array} \right]$$

Since the scope of a bound variable is unique, this is a well-defined feature (and something similar is common in implementations of theorem provers). It can now be exploited to represent scopally underspecified structures. For example, we can now represent formulas in which argument positions are bound to variable arguments, which in turn point to unique quantified formulas, whose scopes, however, needn't be specified. An example of a term in such a structure would be the following:

$$\text{"an employee"} \quad \boxed{1} \quad \left[\begin{array}{l} \text{var} \\ \text{INDEX} \\ \text{BINDER} \left[\begin{array}{l} \text{QUANT} \left[\begin{array}{l} \text{gen-quant} \\ \text{DET } \exists \\ \text{VAR } \boxed{1} \\ \text{REST} \left[\begin{array}{l} \text{atomic-wff} \\ \text{PRED employee} \\ \text{FIRST } \boxed{1} \end{array} \right] \end{array} \right] \\ \text{SCOPE} \end{array} \right] \end{array} \right] \end{array} \right]$$

The point of employing a representation such as this is, of course, that it encodes all of the information associated with the binding of a quantified term to a particular argument position, but none of the information associated with the scope the quantifier takes. It can thus serve as part of a representation for underspecified scopes.

Before demonstrating how this sort of representation might be employed, it will be useful to introduce an abbreviatory convention. As the feature representation of the quantified formulas above amply demonstrates, the standard logic representations are considerably more concise than the representations proposed here (in feature descriptions). Because of this, we shall abbreviate lengthy feature descriptions with the logic formulas they describe, wherever this can be done without danger of confusion. Thus for the structure above we shall write:

$$\text{"an employee"} \quad [1] \quad \left[\begin{array}{l} \text{var} \\ \text{INDEX } x \\ \text{BINDER} \left[\begin{array}{l} \text{q-wff} \\ \text{QUANT } [\exists x \text{ employee}([1])] \\ \text{SCOPE} \end{array} \right] \end{array} \right]$$

Where there is no possibility of confusion, we use the (more compact) logical notation to abbreviate AVM's describing the logic.

6.3 Quasi-Logical Form (QLF)

In this section we employ the representations proposed in the last to provide a level of representation in which scope is unspecified. First, we assume that the quantified formulas in the different VAR-BINDER positions have themselves unspecified scopes, in order to obtain a structure whose information content resembles the CLE's "quasi-logical forms" (Alshawhi and others July 1989), in that it shows which quantifiers bind which argument positions, but it says nothing about the relative scopes of the quantifiers. It is crucial to note here that the structure below does NOT actually describe any quantified formulas at all—it merely describes an atomic formula whose argument positions are filled by variables. The parallel to Schubert and Pelletier's and Alshawhi et al.'s structures is genuine—we have defined nonlogical structures from which genuine semantics are readily derived, just as they did.

An employee represents each department

$$(4) \quad \left[\begin{array}{l} \text{atomic-wff} \\ \text{PRED represent} \\ \text{FIRST } [1] \left[\begin{array}{l} \text{var} \\ \text{INDEX } x \\ \text{BINDER} \left[\begin{array}{l} \text{q-wff} \\ \text{QUANT } [\exists x \text{ employee}([1])] \end{array} \right] \end{array} \right] \\ \text{SECOND } [2] \left[\begin{array}{l} \text{var} \\ \text{INDEX } y \\ \text{BINDER} \left[\begin{array}{l} \text{q-wff} \\ \text{QUANT } [\forall y \text{ department}([2])] \end{array} \right] \end{array} \right] \end{array} \right]$$

Although this sort of representation is generally known under the CLE term "quasi-logical form" (hence: QLF), it is for all intents and purposes just Schubert and Pelletier 1982's "ambiguous logical syntax", or Hobbs and Shieber 1987's "wff's with complex terms", or Fenstad et al. 1987's "unscoped situation schemata". The idea of deriving (during parsing)

unscoped representations from which scoped formulas could be algorithmically derived is standard (and good) practice among theoretically minded computational semanticists.

But like all good practice, it invites theoretical questions as to WHAT it assumes and WHY it is effective. We have nothing to say definitively on the latter (agreeing with the widespread prejudice that it must have to do with the feasibility of ignoring or at least postponing scope disambiguation). The former question is particularly interesting here, because feature-based semantics can provide a general account of the nature of QLF: it simply represents one way of partially specifying semantic constraints. But QLF is theoretically suspect because it is an *ad hoc* representation. Here we can take a step further: given our general scheme of allowing the statement of constraints over logical forms, there is no need for a distinguished level of QLF; i.e., we can encode the information in QLF in the semantic metalanguage under development here.

Before showing how this is possible, it is worth noting that while the representation in (4) includes all the information in QLF, still it does not DESCRIBE the disambiguated scopings—instead, it stands in an algorithmic relation to them. That is, there is an algorithm which maps from representations such as these to scope disambiguations. The representation (4) thus only makes sense within a system with a level of QLF. What is distinctive about the feature-based systems under discussion here is that one can ELIMINATE the level of QLF, even while preserving all the information it contains. This is possible if the opportunity for underspecification in feature structures is exploited properly.

(4) therefore only suggests how QLF might be specified; we would prefer, however, to eliminate QLF in favor of variously (under)specified semantic feature structures. We seek a single representation which (in the sense of feature logics) subsumes various scope possibilities. These possibilities should be obtained by further specifications of feature values.

6.4 Underspecified Scopes

The information in a quasi-logical form (above) is NOT that a particular quantified-wff has a particular scope, but rather EITHER that a particular quantified-wff has a particular scope OR that its scope has a particular scope, OR that its scope's scope...etc. This motivates the addition of an apparently redundant feature:

$$\left[\begin{array}{l} q\text{-wff} \\ \text{QUANT} \\ \text{SCOPE } [1] \left[\text{SCOPE}^* [2] \right] \\ \text{SCOPE}^* \{ [1], [2] \} \end{array} \right] \text{ where '}' \text{ and '}' \text{ delimit feature description disjunctions.}$$

We require that the feature SCOPE^* is defined wherever SCOPE is (on $q\text{-wff}$), and that it always satisfy the disjunction here. Intuitively, SCOPE^* is the kernel, or nuclear scope of a complex formula; it functions as a regular path specification of the sort used by Kaplan and Maxwell 1988. Furthermore, *quantified-wff* is subject to a type restriction that SCOPE^* is either identical to the value of SCOPE or to the value of SCOPE^* in the scope.

This feature appears redundant, but it is only redundant when formulas are fully specified. It allows us to represent quasi-logical forms in a fashion that genuinely subsumes scopally unambiguous formulas. But to show this we likewise need to characterize the quantifiers involved, so that we add a second apparently ambiguous feature, with a similar type restriction:

$$\left[\begin{array}{l} q\text{-wff} \\ \text{QUANT} [1] \\ \text{SCOPE} \left[\text{QUANT}^* [2] \right] \\ \text{QUANT}^* [3] \end{array} \right] \text{ Condition: } [3] = \{ * [1] * \} \cup [2]$$

Starred braces denote set delimiters. We again require that $QUANT^*$ is defined wherever $QUANT$ is (on $q\text{-wff}$). Intuitively, $QUANT^*$ is the set of quantifiers involved in a formula. The condition in this case is again recursive (and no longer expressible as a regular path expression), viz., that $QUANT^*$ be the union of the $SCOPE$'s $QUANT^*$ together with the singleton set of the local $QUANT$, so that $QUANT^*$ is the set of quantifiers along the path to an atomic scope. Given these definitions, we can now specify a feature structure which subsumes the various scope disambiguities and which allows that the level of "quasi-logical form" be eliminated. We examine the feature structure description below, intended as a representation of the following sentence (where scope is underspecified):

An employee represents each department

$$\left[\begin{array}{l} q\text{-wff} \\ QUANT^* \{ * \exists x \text{ employee}(x), \forall y \text{ department}(y) * \} \\ SCOPE^* [\text{represent}(x, y)] \end{array} \right]$$

This structure describes formulas whose dominant operator is one of the two quantifiers in $QUANT^*$, and for which the atomic-wff $\text{represent}(x, y)$ occurs on the path $SCOPE^*$. The structure may be further specified e.g., by setting the scope feature (to be one of the two quantifiers involved).

Given the definitions above, this feature description can be true of exactly two formulas. Proof: $QUANT$ must have a value, and $QUANT^*$ must contain it (def.). Choose one of the two possibilities, the second will be the (parallel) second solution. We choose the existential quantifier first here. Then $SCOPE|QUANT^*$ must have other quantifier, and $SCOPE|SCOPE^*$ must be identical to $SCOPE^*$ (since otherwise type clash). Thus above reduces to:

$$\left[\begin{array}{l} q\text{-wff} \\ QUANT \exists x \text{ employee}(x), \\ SCOPE \left[\begin{array}{l} q\text{-wff} \\ QUANT^* \{ * \forall y \text{ department}(y) * \} \\ SCOPE^* \text{represent}(x, y) \end{array} \right] \end{array} \right]$$

We examine the value of $SCOPE$. It must define $QUANT$, which must therefore be equal to the element in the singleton $QUANT^*$ (since the latter had to contain the $QUANT$ value). Thus $SCOPE|SCOPE$ cannot be anything with a $QUANT$ value—and therefore nothing with a $SCOPE$ or $SCOPE^*$ value. But $SCOPE|SCOPE^*$ must be either $SCOPE|SCOPE$ or $SCOPE|SCOPE|SCOPE^*$. Since the latter cannot be defined, it must be the former. Thus:

$$\left[\begin{array}{l} q\text{-wff} \\ QUANT \exists x \text{ employee}(x), \\ SCOPE \left[\begin{array}{l} q\text{-wff} \\ QUANT \forall y \text{ department}(y) \\ SCOPE \text{represent}(x, y) \end{array} \right] \end{array} \right]$$

The second reading involves choosing the second quantifier first, but is otherwise parallel. $\square$

This indicates that the required subsumption relations indeed hold of this formula (and that no further logic formulas are described by this AVM). Thus scope disambiguation can be characterized at this level.⁴ It is clear that the characterization above crucially involved

⁴For the purposes of feature theory, it is worth noting that we genuinely need the set construct in order to describe the quantifiers involved in a logical form. Had we attempted to provide the specifications using, e.g., disjunction, we would have had insufficiently exact control over quantifiers in subformulas. In that case the disjunction could be satisfied, e.g., by setting the top-level $QUANT$ value equal to one of the quantifiers involved—leaving the others free to vary arbitrarily.

Pollard and Moshier set descriptions, which can never hold of sets with more elements than the set description has. For those who would prefer other notions of sets, the required restrictions can be obtained in interesting ways. For this we need further notions: first, a restriction to closed formulas, as defined above. Second, we need a way to ban vacuous quantification, as provided in Nerbonne 1992.

The scope ambiguity problem is classic, and the Schubert and Pelletier and Alshawi et al. solution to it is generally recognized to be the best available, and certainly one of the most practical. But it rests on the theoretically questionable foundation of "quasi-logical form". The point of demonstrating that one can obtain the same predictions as the "quasi-logical form" theory in a theoretically motivated fashion—through the use of feature structures as a formalized metalanguage for logic—is to show the capability of the feature-based approach as a foundation for semantics and disambiguation.

A second form of scope disambiguation involves "nested quantifiers", which are treated in Nerbonne 1992. Several further refinements are potentially interesting: first, we may generalize to account for scope ambiguity involving arbitrary operators (including, e.g., negation and modal operators) by creating a supertype of *quantified-wff*, say *operator-wff* and defining OPERATOR* and SCOPE* in analogy to QUANT* and SCOPE* above, respectively; second, we need to demonstrate how one could translate into the underspecified representation (cf. Nerbonne 1991); third, it would be most interesting to identify, represent and exploit disambiguating factors; fourth, it would be worth one's while to explore the semantic use of underspecified representations which are not actually subjected to disambiguation (cf. Poesio 1991 for a proposal for a semantic representation scheme in which scopes are unspecified); and fifth, the incorporation of syntactic restrictions on scope (e.g., scope islands) should be investigated.

7 Conclusions

The program of encoding semantics in the same formalism as syntax, found in a number of current computational linguistics frameworks, must face the issue of the expressive strength of the formalisms required. On the one hand, computational formalisms are preferably kept weak enough for descriptions (or programs) written in them to be guaranteed to be tractable; failing tractability, to be decidable; and failing even decidability, at the very least to be complete. Semantics formalisms, on the other hand, need to deal with the inherent higher-order facilities of natural language meanings—e.g., higher order quantification ('MOST'), type-raising, and even intensionality. These make complete logics impossible, and render chimerical goals such as decidability and tractability. Under these circumstances a strategy is appropriate which distinguishes formalisms by their computational properties, and provides for interfaces between them (rather than attempting to reduce all specification to the least tractable formalism).

In this context we propose to distinguish grammatical and semantic formalisms, to provide for an interface by using the former as a metalinguistic specification of the latter, and integrate (some) processing by compiling semantic descriptions even during grammatical processing. The strategy is appropriate not only for the syntax/semantics interface, but shows promise in dealing with nonsyntactic constraints on semantics, such as those arising in phonology and pragmatics. It furthermore allows the statement of generalizations which bridge syntax and semantics, such as those for which HPSG is admired.

An elaboration of this concept was introduced as illustration, and a sketch of semantic underspecification (relative quantifier scope) was provided, which moreover eliminates the need for auxiliary levels of logical form. Nerbonne 1991 provides further elaboration on the issue of disambiguation. On the negative side we must note that the program of genuinely

integrating syntax and semantics is not realized completely here, since some semantic processing, namely genuinely semantic inference, must be separate from the construction of semantic forms. This problem exists as well, e.g., in Montague's program, where syntax must first be mapped into logic for any inference to occur. The feature-based theories may have an advantage here in that it may be possible to define (weak) semantic consequence relations directly on the feature structures, but this is a topic for later work.

References

- Alshawhi, H., et al. July 1989. Research Programme in Natural Language Processing. Final report, Alvey Project No. ALV/PRJ/IKBS/105, SRI Cambridge Research Centre.
- Barwise, J. 1989. Conditionals and Conditional Information. In *The Situation in Logic*, ed. J. Barwise, 97-136. Stanford: CSLI.
- Barwise, J., and R. Cooper. 1981. Generalized Quantifiers and Natural Language. *Linguistics and Philosophy* 4(2):159-219.
- Barwise, J., and J. Perry. 1983. *Situations and Attitudes*. Cambridge: MIT Press.
- Blackburn, P. 1991. Modal Logic and Attribute Value Structures. In *Constraint Propagation, Linguistic Description, and Computation*, ed. M. Rosner, C.J. Rupp, and R. Johnson, 1-19. Lugano: Instituto Dalle Molle IDSIA.
- Carpenter, B. to appear, 1992. *The Logic of Typed Feature Structures*. Tracts in Theoretical Computer Science. Cambridge: Cambridge University Press.
- Carpenter, B., C. J. Pollard, and A. Franz. 1991. The Specification and Implementation of Constraint-Based Unification Grammar. In *Proceedings of the Second International Workshop on Parsing Technology*. Cancun.
- Creary, L. G., and C. J. Pollard. 1985. A Computational Semantics for Natural Language. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, 172-179.
- Dalrymple, M., S. M. Shieber, and F. C.N. Pereira. 1991. Ellipsis and Higher-Order Unification. *Linguistics and Philosophy* 14(4):399-452.
- Dowty, D. 1991. Thematic Proto-Roles and Argument Selection. *Language* 67(3).
- Ebbinghaus, H.-D., J. Flum, and W. Thomas. 1978. *Einführung in die Mathematische Logik*. Darmstadt: Wissenschaftliche Buchgesellschaft.
- Fenstad, J. E., P.-K. Halvorsen, T. Langholm, and J. van Benthem. 1987. *Situations, Language, and Logic*. Dordrecht: Reidel.
- Gawron, J. M., and S. Peters. 1990. *Anaphora and Quantification in Situation Semantics*. Stanford University: CSLI Lecture Notes.
- Gazdar, G., G. K. Pullum, R. Carpenter, E. Klein, T. Hukari, and R. D. Levine. 1987. Category Structures. Technical Report CSLI-87-102, CSLI.
- Hobbs, J. R., and S. M. Shieber. 1987. An Algorithm for Generating Quantifier Scopings. *Computational Linguistics* 13(1-2).
- Jackendoff, R. 1972. *Semantics Interpretation in Generative Grammar*. Cambridge: MIT Press.
- Johnson, M. 1988. *Attribute Value Logic and the Theory of Grammar*. Stanford: Center for the Study of Language and Information.
- Kaplan, D. 1979. On the Logic of Demonstratives. In *Contemporary Perspectives in the Philosophy of Language*, ed. P. A. French, J. Theodore E. Uehling, and H. K. Wettstein, 401-412. Minneapolis: University of Minnesota Press.
- Kaplan, R., and J. Maxwell. 1988. An Algorithm for Functional Uncertainty. In *Proceedings of Coling 1988*, 303-305. Budapest.
- Kaplan, R., and A. Zaenen. 1988. Long-Distance Dependencies, Constituent Structure, and Functional Uncertainty. In *Alternative Conceptions of Phrase Structure*, ed. M. Baltin and A. Kroch, xxx-yyy. Chicago: University of Chicago Press.

- Kasper, R. T., and W. C. Rounds. 1986. A Logical Semantics for Feature Structures. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, 257-266. Columbia University.
- Krifka, M. 1991. A Compositional Semantics for Multiple Focus Constructions. draft.
- Moens, M., J. Calder, E. Klein, M. Reape, and H. Zeevat. 1989. Expressing Generalizations in Unification-Based Grammar Formalisms. In *Proceedings of the 4th Meeting of the European Chapter of the Association for Computational Linguistics*, 174-181.
- Moore, R. C. 1981. Problems in Logical Form. In *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics*, 117-124.
- Moore, R. C. 1989. Unification-Based Semantic Interpretation. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, 33-41.
- Nerbonne, J. 1991. Feature-Based Disambiguation. In *Linguistic Description and Constraint Resolution*, ed. R. Johnson, M. Rosner, and C.J. Rupp, xxx-yyy.
- Nerbonne, J. 1992. A Feature-Based Syntax/Semantics Interface. In *Proceedings of the Second International Conference on the Mathematics of Language, ed. A. Manaster-Ramer and W. Zadrozny, xxx-yyy. Annals of Mathematics and Artificial Intelligence.*
- Poesio, M. 1991. Relational Semantics and Scope Disambiguation. In *Proceedings of the Second Conference on Situation Theory and Its Applications*, ed. J. Barwise, J. M. Gawron, and G. Plotkin. Stanford University: CSLI Lecture Notes.
- Pollard, C. 1989. The Syntax-Semantics Interface in a Unification-Based Phrase Structure Grammar. In *Views of the Syntax-Semantics Interface: Proceedings of the Workshop "GPSG and Semantics", Technische Universität Berlin, 22-24.Feb 1989*, ed. S. Busemann, C. Hauenschild, and C. Umbach, 167-184. Technische Universität Berlin: KIT FAST.
- Pollard, C., and D. Moshier. 1990. Unifying Partial Descriptions of Sets. In *Information, Language and Cognition*, ed. P. Hanson, Vol. 1 of Vancouver Studies in Cognitive Science, 167-184. Vancouver: University of British Columbia Press.
- Pollard, C., and I. Sag. 1987. *An Information-Based Theory of Syntax and Semantics, Vol. I.* Stanford: CSLI.
- Reape, M. 1991. An Introduction to the Semantics of Unification-Based Grammar Formalisms. Technical Report R3.2.A, DYANA, University of Edinburgh.
- Roberts, D. 1991. Linking Thematic Roles and Syntactic Arguments in HPSG. Master's thesis, Cornell University.
- Rooth, M. 1985. *Association with Focus*. PhD thesis, University of Massachusetts at Amherst.
- Rounds, W. C. 1988. Set Values for Unification-Based Grammar Formalisms and Logic Programming. Technical Report CSLI-88-129, Center for the Study of Language and Information, Stanford University.
- Schubert, L. K., and F. J. Pelletier. 1982. From English to Logic: Context-Free Computation of Conventional Logic Translations. *Journal of the Association for Computational Linguistics* 165-176.
- Shieber, S. 1986. *An Introduction to Unification-Based Approaches to Grammar*. Stanford University: Center for the Study of Language and Information.
- Shieber, S., H. Uszkoreit, J. Robinson, and M. Tyson. 1983. The Formalism and Implementation of PATR-II. In *Research on Interactive Acquisition and Use of Knowledge*. Menlo Park, Cal.: AI Center, SRI International.
- Smolka, G. 1988. A Feature Logic with Subsorts. Technical Report 33, WT LILOG-IBM Germany.
- von Stechow, A. 1989. Focusing and Backgrounding Operators. Arbeitspapier 6, Fachgruppe Sprachwissenschaft, Universität Konstanz.
- Wechsler, S. M. 1991. *Argument Structure and Linking*. PhD thesis, Stanford University.
- Zwicky, A. M., and J. Sadock. 1975. Ambiguity Tests and How to Fail Them. In *Syntax and Semantics, Vol. IV*, ed. J. Kimball, 1-36. New York: Academic Press.

A Dialectica Model of the Lambek Calculus

Valeria de Paiva
Computer Laboratory
University of Cambridge CB2 3QG

This note discusses the Lambek Calculus (Lambek [1958], van Benthem[1988]) and some variants from the point-of-view of Linear Logic (Girard [1987]). The Lambek Calculus was first introduced, with the name of Syntactic Calculus, by J. Lambek in 1958 as an explanation of the mathematics of sentence structure. After a long period of ostracism, around 1980, the Syntactic Calculus, now called the Lambek Calculus, was taken up by logicians interested in Computational Linguistics, especially in the area of Categorical Grammar. The calculus was considered “like a logical system” but much too weak to be taken seriously as a logic. To quote from van Benthem [1988],

...One fundamental system of this kind is the so-called ‘Lambek Calculus’, whose type-change rules show a close analogy with the inference rules of constructive propositional logic.

This situation could change drastically with the advent of Girard’s Linear Logic (Girard [1987]). Girard showed that there is a full embedding, preserving proofs, of Intuitionistic Logic into Linear Logic with modality ‘!’. That means, in loose terms, that Linear Logic is a logic¹ as expressive as Intuitionistic Logic and the claim is that it is more precise, i.e it makes finer distinctions. As the Lambek Calculus is the (multiplicative) fragment of non-commutative Intuitionistic Linear Logic, several interesting questions can be asked. One of them, posed by Morrill et al [1990] is whether we can extend the Lambek Calculus with a modality that does for the structural rule of (*exchange*) what the modality ‘!’ does for the rules of (*weakening*) and (*contraction*). A very preliminary proposal, which answers this question affirmatively, is set forward in this paper. But it must be said from the start that the ‘answer’ is only provided in semantical terms. The proof theory of the systems considered should be investigated in future work. Another warning is that the perspective of this note is basically from Category Theory as a branch of Mathematics, so words like categories and functors are always meant in their mathematical, rather than linguistic or philosophical sense.

In the first section we recall Linear Logic and provide the transformations to show that the Lambek Calculus L really is the multiplicative fragment of (non-commutative) Intuitionistic Linear Logic. In the second section we describe the usual String Semantics for the Lambek Calculus L and generalise it, using a categorical perspective. In the third section we describe a Dialectica-style model for the Lambek Calculus. In the last section we discuss (semantically valid) modalities and hint at some future work.

I would like to thank Jan van Eijck for inviting me to give the talk that became this note, thereby gently ‘forcing’ me to think about the subject, as well as, for his generous hospitality. I also would like to thank Martin Hyland, Harold Schellinx, Dirk Roorda, Mark Hepple, Glyn Morrill and Michael Moortgat for several useful discussions. Many of the ideas in this paper have been shaped by these discussions, but of course the mistakes are all mine. Finally I want to thank Jim Lambek for ‘putting me right’ about how completeness has nothing to do with the existence of two disjunctions, in the most friendly possible way.

1 From Linear Logic to the Lambek Calculus

Intuitionistic Linear Logic, henceforth **ILL**, was described in Girard and Lafont's [1987] TAPSOFT paper. One of the best ways of thinking about the system **ILL** is to start with Gentzen's Intuitionistic Calculus **LJ**, as for instance in Takeuti [1975] (page 19) and delete the structural rules of (*weakening*) and (*contraction*)

$$\frac{\Gamma \vdash A}{\Gamma, B \vdash A} \text{ (weakening)} \quad \frac{\Gamma, B, B \vdash A}{\Gamma, B \vdash A} \text{ (contraction)}$$

Without (*weakening*) and (*contraction*) the two possible formulations for the rule of conjunction,

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \quad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \wedge B}$$

give rise to two different connectives, which Girard calls, respectively, the 'additive' and the 'multiplicative' conjunctions. Recall the rules and axioms for **ILL** in a sequent-calculus formulation.

Intuitionistic Linear Logic (ILL)

Axioms:

$$\begin{array}{l} A \vdash A \text{ (identity)} \\ \vdash I \\ \Gamma \vdash 1 \quad \Gamma, 0 \vdash A \end{array}$$

Structural Rules:

$$\frac{\Gamma, A, B, \Gamma' \vdash C}{\Gamma, B, A, \Gamma' \vdash C} \text{ (permutation)} \quad \frac{\Gamma \vdash A \quad A, \Gamma' \vdash B}{\Gamma, \Gamma' \vdash B} \text{ (cut)}$$

Logical Rules:

Multiplicatives:

$$\begin{array}{l} \text{(unit)} \frac{\Gamma \vdash A}{\Gamma, I \vdash A} \\ \text{(\otimes}_l\text{)} \frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C} \quad \text{(\otimes}_r\text{)} \frac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma, \Gamma' \vdash A \otimes B} \\ \text{(\multimap}_l\text{)} \frac{\Gamma \vdash A \quad B, \Gamma' \vdash C}{\Gamma, A \multimap B, \Gamma' \vdash C} \quad \text{(\multimap}_r\text{)} \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B} \end{array}$$

Additives:

$$\begin{array}{l} \text{(\&}_r\text{)} \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \& B} \quad \text{(\&}_l\text{)} \frac{\Gamma, A \vdash C}{\Gamma, A \& B \vdash C} \quad \frac{\Gamma, B \vdash C}{\Gamma, A \& B \vdash C} \\ \text{(\oplus}_l\text{)} \frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \oplus B \vdash C} \quad \text{(\oplus}_r\text{)} \frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B} \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B} \end{array}$$

A difference between the sequent-calculus for traditional intuitionistic logic and the one above is that one thinks of Γ as a set in the traditional case, while here Γ means a multiset (or a bag) of assumptions.

Note that in the multiplicative fragment, we can put different contexts Γ and Γ' together - rules $(\otimes_r)$ and $(\multimap_l)$ - while in the additive fragment, we must have exactly the same context Γ - rules $(\&_r)$ and $(\oplus_r)$ - to apply a rule. Observe that, as the system **ILL** satisfies Cut-elimination, there is only one structural rule left, namely (*permutation*). Also in **ILL** we have only one (additive) disjunction ' $\oplus$ ', which contrasts with Classical Linear Logic (Girard [1987]) or Full Intuitionistic Linear Logic (Hyland and de Paiva [1990]).

If we delete the rules for the additives (and their axioms) from the system above, we have the system **LP** studied by van Benthem [1988] and Moorgat [1990], that is the Lambek Calculus with Permutation. Note that this system could also be called **ILL_m**, that is the multiplicative fragment of **ILL**. For the sake of completeness we list the axioms and rules of **LP**.

Lambek Calculus with Permutation (LP)

Axioms:

$$\frac{A \vdash A \text{ (identity)}}{\vdash I}$$

Structural Rules:

$$\frac{\Gamma, A, B, \Gamma' \vdash C}{\Gamma, B, A, \Gamma' \vdash C} \text{ (permutation)} \quad \frac{\Gamma \vdash A \quad A, \Gamma' \vdash B}{\Gamma, \Gamma' \vdash B} \text{ (cut)}$$

Logical Rules:

$$\begin{array}{ll} \text{(unit)} \frac{\Gamma \vdash A}{\Gamma, I \vdash A} & \\ \text{(\otimes}_l\text{)} \frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C} & \text{(\otimes}_r\text{)} \frac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma, \Gamma' \vdash A \otimes B} \\ \text{(\multimap}_l\text{)} \frac{\Gamma \vdash A \quad \Gamma', B \vdash C}{\Gamma, \Gamma', A \multimap B \vdash C} & \text{(\multimap}_r\text{)} \frac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma \vdash A \multimap B} \end{array}$$

Now if we remove the last structural rule of (*permutation*), which "means" the commutativity of $\otimes$, the rules for ' $\multimap$ ' can be given in two (now different) guises, eg

$$\text{(\multimap}_r\text{)} \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B} \quad \text{(\multimap}_l\text{)} \frac{A, \Gamma \vdash B}{\Gamma \vdash A \multimap B}$$

Thus we get two directional implications and the system is called the Lambek Calculus **L**. The rules and axioms of **L** are as follows:

Lambek Calculus (L)

Axioms:

$$\frac{A \vdash A \text{ (identity)}}{\vdash I}$$

Structural Rule:

$$\frac{\Gamma \vdash A \quad \Gamma', A, \Delta \vdash B}{\Gamma', \Gamma, \Delta \vdash B} \text{ (cut)}$$

Logical Rules:

$$\begin{array}{c} \text{(unit)} \frac{\Gamma \vdash A}{\Gamma', I, \Gamma'' \vdash A} \quad (\Gamma', \Gamma'' = \Gamma) \\ \\ \begin{array}{cc} \begin{array}{c} \text{(\otimes}_l\text{)} \frac{\Gamma, A, B, \Gamma' \vdash C}{\Gamma, A \otimes B, \Gamma' \vdash C} \\ \text{(\multimap}_l\text{)} \frac{\Gamma \vdash A \quad \Gamma', B, \Delta \vdash C}{\Gamma', A \multimap B, \Gamma, \Delta \vdash C} \\ \text{(\multimap}_l\text{)} \frac{\Gamma \vdash A \quad \Gamma', B, \Delta \vdash C}{\Gamma', \Gamma, A \multimap B, \Delta \vdash C} \end{array} & \begin{array}{c} \text{(\otimes}_r\text{)} \frac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma, \Gamma' \vdash A \otimes B} \\ \text{(\multimap}_r\text{)} \frac{\Gamma, \Gamma' \vdash A \otimes B}{\Gamma, A \vdash B} \\ \text{(\multimap}_r\text{)} \frac{\Gamma \vdash A \multimap B}{A, \Gamma \vdash B} \\ \text{(\multimap}_r\text{)} \frac{A, \Gamma \vdash B}{\Gamma \vdash A \multimap B} \end{array} \end{array}$$

Note that, due to the absence of permutation, Γ is not a multiset, but a sequence of assumptions. The lack of rule (*permutation*) splits the connective ' $\multimap$ ' into two implications ' $\multimap$ ' and ' $\multimap$ ', which one can think of as *over* and *under* or predictive and retroactive implication. For that reason, we call ' $\multimap$ ' *forward* and ' $\multimap$ ' *back*. Lack of (*permutation*) also accounts for the small modifications in the rules (*cut*), (*unit*) and ($\otimes_l$).

There are several variants of this calculus. The system above is called the *syntactic calculus* by Lambek in page 305 Lambek [1988] but in Lambek [1987] we have a version without the constant I . Moorgat [1990] presents a version without (explicit) tensors $\otimes$ (and I), which he calls the Lambek-Gentzen Calculus L. Moorgat also gives a Hilbert-style presentation of this logic, which he calls Z for Zielonka and says that the equivalence between Z and the Lambek-Gentzen system was shown by Buszkowski. Both Moorgat and Lambek consider briefly the system where the operation $\otimes$ - usually denoted by a dot ' $\cdot$ ' - is not associative, but we will not do this here.

In Lambek [1987] we also found the definition,

Definition 1 A categorial grammar of a language may be viewed as consisting of the syntactic calculus L freely generated from a finite set $\{S, N, \dots\}$ of basic types together with a dictionary which assigns to each word of the language a finite set of types composed from the basic types and I by the three binary operations, $\otimes$, $\multimap$ and $\multimap$.

but we will only mention the definition of a categorial grammar and not try anything closer to real Linguistics in this note.

Moorgat and van Benthem also argue that for certain linguistics applications (non-peripheral extraction is one of the examples) one would like to have some amount of permutation in the system. We will try and say something about that in the last section. Before going on to semantics we present a last sequent-calculus system, very natural from the Linear Logic viewpoint, namely the Lambek calculus with additives, which we call LA.

Lambek Calculus with Additives (LA)

Axioms:

$$\begin{array}{c} A \vdash A \text{ (identity)} \\ \vdash I \\ \Gamma \vdash 1 \quad \Gamma, 0, \Gamma' \vdash A \end{array}$$

Structural Rule:

$$\frac{\Gamma \vdash A \quad \Gamma', A, \Delta \vdash B}{\Gamma', \Gamma, \Delta \vdash B} \text{ (cut)}$$

Logical Rules:

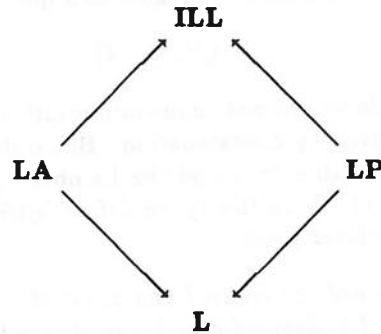
Multiplicatives:

$$\begin{array}{c} \text{(unit)} \frac{\Gamma, \Gamma' \vdash A}{\Gamma, I, \Gamma' \vdash A} \\ \\ \begin{array}{cc} \text{(\otimes}_l\text{)} \frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, A \otimes B, \Delta \vdash C} & \text{(\otimes}_r\text{)} \frac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma, \Gamma' \vdash A \otimes B} \\ \text{(\multimap}_l\text{)} \frac{\Gamma \vdash A \quad \Gamma', B, \Delta \vdash C}{\Gamma', A \multimap B, \Gamma, \Delta \vdash C} & \text{(\multimap}_r\text{)} \frac{\Gamma \vdash A \multimap B}{\Gamma, A \vdash B} \\ \text{(\multimap}_l\text{)} \frac{\Gamma \vdash A \quad \Gamma', B, \Delta \vdash C}{\Gamma', \Gamma, A \multimap B, \Delta \vdash C} & \text{(\multimap}_r\text{)} \frac{A, \Gamma \vdash B}{\Gamma \vdash A \multimap B} \end{array} \end{array}$$

Additives:

$$\begin{array}{ccc} \text{(\&}_r\text{)} \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \& B} & \text{(\&}_l\text{)} \frac{\Gamma, A, \Delta \vdash C}{\Gamma, A \& B, \Delta \vdash C} & \frac{\Gamma, B, \Delta \vdash C}{\Gamma, A \& B, \Delta \vdash C} \\ \text{(\oplus}_l\text{)} \frac{\Gamma, A, \Delta \vdash C \quad \Gamma, B, \Delta \vdash C}{\Gamma, A \oplus B, \Delta \vdash C} & \text{(\oplus}_r\text{)} \frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B} & \frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B} \end{array}$$

Note that the system **LA** could also be called non-commutative intuitionistic linear logic. That is **ILL**=**LP** + **A**. As a summary of the systems discussed we draw a diagram



But remember that to obtain **LP** from **L** we add an inference rule, namely (*permutation*), whereas to obtain **LA** from **L** we add the additive connectives.

2 Semantics

The usual model for the logical system L (Lambek [1958]) is given by sets of expressions from a (generally taken as finite) non-empty universe V^+ . This model is called the **String Semantics** for the Lambek Calculus L . To describe it, given subsets A and B of V^+ , define:

$$A \cdot B = \{xy \in V^+ \mid x \in A \text{ and } y \in B\}$$

$$C/B = \{x \in V^+ \mid \text{for all } y \in B, xy \in C\}$$

$$A \backslash C = \{y \in V^+ \mid \text{for all } x \in A, xy \in C\}$$

If you call $\mathcal{N}$ the set of all subsets of V^+ , $\mathcal{N}$ has a natural order induced by *set containment*. Concatenation, represented by the dot in $A \cdot B$, gives $\mathcal{N}$ a (non-commutative) monoidal structure. Concatenation is associative, has identity given by the empty sequence, and is, in general, non-commutative. Given $A, B \in \mathcal{N}$

- $A \cdot (B \cdot C) = (A \cdot B) \cdot C$ (associativity of concatenation);
- If we denote by I the empty sequence $A \cdot I = I \cdot A = A$ (identity);
- $A \cdot B \neq B \cdot A$, in general.

Now, we want to generalise the structure above, taking our clues from Category Theory (MacLane [1971]). To do that we first recall the definition of an ordered (non-commutative) monoid.

Definition 2 An ordered (non-commutative) monoid is a poset $(M, \leq)$ with a given compatible monoidal structure $(M, \circ, e)$. That is, a set M equipped with a binary relation ' $\leq$ ' satisfying:

- $a \leq a$ for all a in M (reflexivity)
- $a \leq b$ and $b \leq c \Rightarrow a \leq c$ (transitivity)
- $a \leq b$ and $b \leq a \Rightarrow a = b$ (antisymmetry)

together with a monoid structure $(\circ, e)$ consisting of a 'multiplication' $\circ: M \times M \rightarrow M$ and a distinguished object ' e ' of M , such that the following hold:

- $(a \circ b) \circ c = a \circ (b \circ c)$ (associativity)
- $a \circ e = e \circ a = a$ (identity)

The structures are compatible in the sense that, if $a \leq b$, we have

- $a \circ c \leq b \circ c$, for all c in M and $c \circ a \leq c \circ b$, for all c in M .

We write an ordered (non-commutative) monoid as a quadruple $(M, \leq, \circ, e)$. Note that

$$(\mathcal{N}, \leq, \cdot, I)$$

in the string semantics above is an ordered (non-commutative) monoid as the order $\leq$ is compatible with the multiplication given by concatenation. But ordered monoids are not the structure we are after, we need some structure to model the Lambek "slashes" (or $\leftarrow$ and $\rightarrow$ in our notation). Following the tradition of lattice theory we define *left(right)-pseudocomplements* and more importantly the notion of a *biclosed poset*.

Definition 3 Suppose M is a ordered monoid and $a, b \in M$. If there exists a largest $x \in M$ such that $a \circ x \leq b$ then this element is denoted $a \rightarrow b$ and it is called the left pseudocomplement of a wrt b . Similarly, if there exists a largest $x \in M$ such that $x \circ a \leq b$ then this element is denoted $a \leftarrow b$ and it is called the right pseudocomplement of a wrt b .

A biclosed poset is an ordered monoid $(M, \leq, \circ, e)$ such that $a \rightarrow b$ and $a \leftarrow b$ exist for all a and b in M . We write a biclosed poset as 6-tuple $(M, \leq, \circ, e, \rightarrow, \leftarrow)$.

Note that in a biclosed poset $(M, \leq, \circ, e, \multimap, \multimap)$ the binary operations $\multimap$ and $\multimap$ are antimonotonic in their first coordinate, monotonic in their second coordinate and such that:

$$a \circ b \leq c \text{ iff } a \leq b \multimap c \text{ in } M \text{ and } b \circ a \leq c \text{ iff } a \leq c \multimap b \text{ in } M$$

The notion of a biclosed poset is the natural simplification to the poset set up of the notion of a (non-symmetric) monoidal biclosed category, see Eilenber and Kelly [1965]. It is clear that $(\mathcal{N}, \leq, \cdot, I)$ with A/B and $A \backslash B$ defined for the string semantics is a biclosed poset. We now state some propositions, which are well-known facts:

FACT 1 *Any biclosed poset $(M, \leq, \circ, e, \multimap, \multimap)$ is a model for the Lambek Calculus L.*

By that we mean that if $\Gamma \vdash A$ is a derivable sequent in L, and $()^0$ assigns to each basic type A an element A^0 of the biclosed poset M , then $\Gamma^0 \leq A^0$.

Note that if the monoidal operation ' $\circ$ ' is commutative, a left pseudocomplement is also a right pseudocomplement, and the two implications ' $\multimap$ ' and ' $\multimap$ ' collapse back into a single one. The commutative structure corresponding to a biclosed poset is a *closed poset*, examples can be found in Lawvere [1973] and Flagg [1990]. Hence:

FACT 2 *Any (symmetric) closed poset $(M, \leq, \circ, e, \multimap)$ is a model for the Lambek Calculus with permutation LP.*

Recall that LA denotes the Lambek calculus with additives. The system LA satisfies cut-elimination, as the proof for ILL - with the modification made by Lambek to account for two implications - also work for LA. The calculus LA was given semantics by Morrill [1990], Moortgat [1990] and others. One way of adding additives to the string model above is defining:

$$A \cap B = \{x \in V^+ | x \in A \text{ and } x \in B\}$$

$$A \cup B = \{x \in V^+ | x \in A \text{ or } x \in B\}$$

To generalise the string semantics with additives, we could say:

FACT 3 *Any biclosed poset with (finite) meets and joins $(M, \leq, \circ, e, \multimap, \multimap, \wedge, \vee)$ is a model for the Lambek Calculus with additives LA.*

But note that adding booleans is stronger than adding additives, as for instance additives do not necessarily satisfy distributivity $A \& (B \oplus C) \not\equiv (A \& B) \oplus (A \& C)$ whereas booleans do (see Roorda [1991]). A biclosed poset with finite meets and joins is the natural simplification to the poset set up of the notion of a (non-symmetric) monoidal biclosed category with finite products and coproducts. For the sake of completeness we state:

FACT 4 *A (symmetric) closed poset with (finite) meets and joins $(M, \leq, \circ, e, \multimap, \wedge, \vee)$ is a model for the Lambek Calculus with additives and full permutation ILL.*

Note that a symmetric closed poset with meets and joins is almost a *quantale* as in Yetter [1990], i.e it is a quantale without completeness. Many people have considered quantales as models for Intuitionistic Linear Logic, but the (algebraic) completeness of a quantale is not necessary to model the propositional logic and it distracts from the essential structure.

Two examples

In this short subsection we show that mathematical structures, which look superficially very different from the traditional models, can model L as well as strings and residuated semigroups. In the first example we calculate with matrices.

Matrices as a Model for L

Consider the set $\mathcal{N}$ of invertible matrices with entries from $\mathbb{N}$ the set of natural numbers,

$$\mathcal{N} = \{\text{invertible square matrices } k \times k \text{ with elements in } \mathbb{N}\}$$

We use Greek letters α, β for our matrices. Of course we have the usual operation of matrix multiplication, denoted

$$\otimes: \mathcal{N} \times \mathcal{N} \rightarrow \mathcal{N}$$

Recall that matrix multiplication is a non-commutative monoidal operation on $\mathcal{N}$. That is

- There is an identity for matrix multiplication I , simply the identity matrix of dimension k ,
- For any matrix α , $\alpha \otimes I = I \otimes \alpha = \alpha$,
- Associativity of matrix multiplication $(\alpha \otimes \beta) \otimes \gamma = \alpha \otimes (\beta \otimes \gamma)$ holds and
- The matrix product is, in general, non-commutative $\alpha \otimes \beta \neq \beta \otimes \alpha$.

We can define a pre-order on $\mathcal{N}$, saying that

$$\alpha \leq \beta \quad \text{iff} \quad \alpha^{-1} \otimes \beta \leq I$$

Given any pre-order we have an induced partial order by taking equivalence classes, the usual construction. The order induced on $\mathcal{N}$ is compatible with the matrix multiplication. Thus $(\mathcal{N}, \leq, \otimes, I)$ is another non-commutative ordered monoid. Moreover, we can use the invertibility of our matrices to define two operations that make $\mathcal{N}$ a biclosed poset. We define two binary operations on elements of $\mathcal{N}$, $[(-) \multimap (-)]$ and $[(-) \multimap (-)]$, by saying that $\alpha \multimap \beta = \alpha^{-1} \otimes \beta$ and $\alpha \multimap \beta = \beta^{-1} \otimes \alpha$.

Note that these are operations of the form $\mathcal{N}^{\text{op}} \times \mathcal{N} \rightarrow \mathcal{N}$. By that we mean that the operations are antimonotonic in the first coordinate and monotonic in the second. Hence:

Proposition 1 *The ordered monoid of invertible $k \times k$ matrices over $\mathbb{N}$, $(\mathcal{N}, \leq, \otimes, I)$ with the operations of matrix multiplication ' $\otimes$ ', and ' $\multimap$ ' and ' $\multimap$ ' defined above constitutes a biclosed poset.*

Thus $(\mathcal{N}, \leq, \otimes, I, \multimap, \multimap)$ is another model for the Lambek Calculus and there are several other models like modules of rings, etc. According to Lambek the idea for the Syntactic Calculus came from algebraic models. Actually it is easy to see that if one has a(n ordered) group, i.e the possibility of inverting elements of an (non-symmetric) ordered monoid, we will have a biclosed poset.

$\mathbb{N}$ as a Model for LP

If we want to produce algebraic models for LP, the calculus with permutation, instead of the Lambek Calculus, we have many more 'familiar' structures, like, for instance, the set of natural numbers or the positive reals.

The set of the natural numbers $\mathbb{N}$ is clearly a (symmetric) *closed poset*, where conjunction ' $\circ$ ' is given by addition of natural numbers, the identity for this conjunction is 0 and the internal-hom is given by truncated subtraction, as in Lawvere [1973], (also in de Paiva [1991]). It is easy to prove the adjunction and another easy argument on Lindenbaum algebras (following Girard's proof) proves completeness of the models above²

In the next section we describe another model for the Lambek Calculus, but this time a categorical one, that is, one where we take in consideration different proofs of the same entailment.

3 A Dialectica Construction

In this section we show that we can construct a Dialectica³category model of the Lambek Calculus in a straightforward way. The construction is meant both as evidence of the flexibility and powerfulness of the dialectica-like constructions and also as a way of, semantically, showing that we can define a modality that adds restricted permutation to the Lambek Calculus. The construction is given without all the details, hence it might look too complicated at first reading. The reader should be warned that some familiarity with Category Theory is assumed in this section.

Now suppose $\mathbf{C}$ is the category of sets and usual maps, so, in particular, $\mathbf{C}$ is a monoidal closed - actually cartesian closed - category with finite products and coproducts. Since $\mathbf{C}$ has finite products, it has a terminal object 1 , any singleton set. It also has an initial object 0 , the empty set. We write - as usual - $A \times B$ for the categorical product and $A + B$ for the coproduct in $\mathbf{C}$. Consider as well a biclosed poset $(\mathcal{N}, \leq, \otimes, e, \multimap, \multimap)$ in $\mathbf{C}$, you could think about it as the string semantics model above.

Note that the partial order on $\mathcal{N}$ induces an order on the hom-set $\mathbf{C}(X, \mathcal{N})$ given by

$$\text{for } f, g: X \rightarrow \mathcal{N} \quad f \leq g \quad \text{iff} \quad \forall x \in X \quad f(x) \leq g(x)$$

Using the data above we can define a (dialectica) category $D_{\mathcal{N}}\mathbf{C}$ as follows.

Definition 4 Given a biclosed poset $(\mathcal{N}, \leq, \otimes, e, \multimap, \multimap)$ and $\mathbf{C}$ the category of Sets and usual functions, the category $D_{\mathcal{N}}\mathbf{C}$ consists of:

- Objects of $D_{\mathcal{N}}\mathbf{C}$ are triples (U, X, α) where α is a map in $\mathbf{C}$ of the form

$$U \times X \xrightarrow{\alpha} \mathcal{N}$$

We call the object above A , because it is determined by the function α ; and write it as $A = (U \multimap X)$.

- Morphisms in $D_{\mathcal{N}}\mathbf{C}$ from an object A to an object B of the form $V \times Y \xrightarrow{\beta} \mathcal{N}$ correspond to pairs of maps in $\mathbf{C}$, $f: U \rightarrow V$ and $F: Y \rightarrow X$ such that in the following diagram

$$\begin{array}{ccc} U \times Y & \xrightarrow{U \times F} & U \times X \\ f \times Y \downarrow & & \downarrow \alpha \\ V \times Y & \xrightarrow{\beta} & \mathcal{N} \end{array}$$

we have $\alpha \circ (U \times F) \leq \beta \circ (f \times Y)$ as morphisms in $\mathbf{C}(U \times Y, \mathcal{N})$.

We write morphisms in $D_{\mathcal{N}}\mathbf{C}$ as diagrams:

$$\begin{array}{ccc} U & \xleftarrow{\alpha} & X \\ f \downarrow & \Downarrow & \uparrow F \\ V & \xleftarrow{\beta} & Y \end{array} \quad \forall u \in U, \forall y \in Y \quad \alpha(u, Fy) \leq \beta(fu y)$$

Note that checking that the data above really defines a category is not totally trivial. The category $D_{\mathcal{N}}\mathbf{C}$ above is a variant of the construction in the author's thesis (de Paiva [1988]) which is related to Chu's construction (Barr [1979]) and to GAME_k , (Lafont and Streicher [1991]).

Proposition 2 *The description above defines a category $D_{\mathcal{N}}\mathbf{C}$.*

Composition of morphisms (f, F) and (g, G) is given by the morphism (gf, FG) . Identities are identities in each coordinate and associativity is given by the associativity of function composition in $\mathbf{C}$, that is $\mathbf{Sets}$.

Having defined the category we want to use as a model for the Lambek Calculus, we now embark in a series of boring and relatively easy calculations to show that this category is monoidal biclosed. We start by defining a monoidal structure on $D_{\mathcal{N}}\mathbf{C}$.

Definition 5 *Given two objects $(U \xrightarrow{\alpha} X)$ and $(V \xrightarrow{\beta} Y)$ in $D_{\mathcal{N}}\mathbf{C}$ we define $A \otimes B$ their tensor product as follows:*

$$A \otimes B = (U \otimes V \xleftarrow{\alpha \otimes \beta} X^V \times Y^U)$$

The morphism " $\alpha \otimes \beta$ " is given intuitively by $\alpha \otimes \beta(u, v, f, g) = \alpha(u, fv) \otimes \beta(v, gu)$, where the second $\otimes$ refers to the (non-commutative) monoidal operation in $\mathcal{N}$.

To give a formal definition of $\alpha \otimes \beta$ consider the map

$$U \times V \times X^V \times Y^U \xrightarrow{("ev, ev") } U \times Y \times V \times X \xrightarrow{\alpha \times \beta} \mathcal{N} \times \mathcal{N} \xrightarrow{\otimes} \mathcal{N}$$

Note that we are using the cartesian closed - hence commutative - structure of $\mathbf{Sets}$ - to write X^V and Y^U - and the monoidal structure of $\mathcal{N}$ to define $\alpha \otimes \beta$.

This operation clearly defines a bifunctor, which is a tensor product. Associativity is straightforward and the object $I = (1 \xrightarrow{I} 1)$ - the map $1 \times 1 \xrightarrow{I} \mathcal{N}$, which only picks the identity I in $\mathcal{N}$ - is the identity for this tensor product.

Note that the 'carrier' of the tensor product is the same for $A \otimes B$ and $B \otimes A$, due to commutativity of products in $\mathbf{C}$, but as objects in $D_{\mathcal{N}}\mathbf{C}$ they are different, so the tensor product in $D_{\mathcal{N}}\mathbf{C}$ is not commutative.

One observation is that if one tries to get a (Dialectica) model for the Lambek calculus using a (non-symmetric) monoidal closed category $\mathbf{C}$ that leads into 'braiding' problems studied by Joyal and Street [1986], which seems worth investigating.

Actually this odd-looking tensor product, analogous to the one in GC cf. de Paiva [1989], is the right one to prove monoidal-biclosedness of $D_{\mathcal{N}}\mathbf{C}$ with respect to the two internal-homs. To define the internal-homs:

Definition 6 *Given two objects $(U \xrightarrow{\alpha} X)$ and $(V \xrightarrow{\beta} Y)$ in $D_{\mathcal{N}}\mathbf{C}$ we define $[A, B]_l$ their left-respectively right - internal hom as follows:*

$$[A, B]_l = (V^U \times X^Y \xleftarrow{\alpha \multimap \beta} U \times Y)$$

$$[A, B]_r = (V^U \times X^Y \xleftarrow{\alpha \multimap \beta} U \times Y)$$

The morphisms " $\alpha \multimap \beta$ " and " $\alpha \multimap \beta$ " are given intuitively by $(\alpha \multimap \beta)(f, F, u, y) = \alpha(u, Fy) \multimap \beta(fu, y)$, respectively, $(\alpha \multimap \beta)(f, F, u, y) = \alpha(u, Fy) \multimap \beta(fu, y)$, where ' $\multimap$ ' and ' $\multimap$ ' are the corresponding 'internal-homs' or 'slashes' in $\mathcal{N}$.

Having defined internal homs and a tensor product we state and prove the desired theorem.

Theorem 1 *The category $D_{\mathcal{N}}\mathbf{C}$ is a (non-symmetric) monoidal biclosed category.*

The proof is very simple, one has to verify the natural isomorphisms

$$\text{Hom}_{D_{\mathcal{N}}\mathbf{C}}(A \otimes B, C) \cong \text{Hom}_{D_{\mathcal{N}}\mathbf{C}}(A, [B, C]_l)$$

$$\text{Hom}_{D_{\mathcal{N}}\mathbf{C}}(B \otimes A, C) \cong \text{Hom}_{D_{\mathcal{N}}\mathbf{C}}(A, [C, B]_r)$$

This can be done by looking at the diagrams

$$\begin{array}{ccc} U \times V & \xleftarrow{\alpha \otimes \beta} & X^V \times Y^U \\ \downarrow f & & \downarrow \langle F_1, F_2 \rangle \\ W & \xleftarrow{\gamma} & Z \end{array} \quad \begin{array}{ccc} U & \xleftarrow{\alpha} & X \\ \downarrow \langle f, F_2 \rangle & & \downarrow F_1 \\ W^V \times Y^Z & \xleftarrow{\beta \multimap \gamma} & V \times Z \end{array}$$

and

$$\begin{array}{ccc} V \times U & \xleftarrow{\beta \otimes \alpha} & Y^U \times X^V \\ \downarrow f & & \downarrow \langle F_1, F_2 \rangle \\ W & \xleftarrow{\gamma} & Z \end{array} \quad \begin{array}{ccc} U & \xleftarrow{\alpha} & X \\ \downarrow \langle f, F_2 \rangle & & \downarrow F_1 \\ W^V \times Y^Z & \xleftarrow{\gamma \multimap \beta} & V \times Z \end{array}$$

and calculating (the hard bit) the sums.

If the morphism $(f, \langle F_1, F_2 \rangle)$ is in $\text{Hom}(A \otimes B, C)$, then we know $(\alpha \otimes \beta) \leq \gamma$, that is

$$(\alpha \otimes \beta)(u, v, F_1 z, F_2 z u) \leq \gamma(f(u, v), z)$$

This means that $\alpha(u, F_1 z v) \otimes \beta(v, F_2 z u) \leq \gamma(f(u, v), z)$ in $\mathcal{N}$. But $\mathcal{N}$ is a biclosed poset, which implies that $\alpha \leq \beta \multimap \gamma$ or $\alpha(u, F_1 z v) \leq \beta(v, F_2 z u) \multimap \gamma(f(u, v), z)$. To show that the corresponding morphism $((f, \overline{F_2}), \overline{F_1})$ is in $\text{Hom}(A, [B, C]_l)$ we have to show $\alpha(u, F_1(v, z)) \leq (\beta \multimap \gamma)((f, F_2)u, \langle v, z \rangle)$ that means $\alpha(u, F_1(v, z)) \leq (\beta \multimap \gamma)(fu, F_2 u, v, z) = \beta(v, F_2 u z) \multimap \gamma(fu, v, z)$ we know what we need, if transposing is allowed in the category $\mathbf{C}$. Similarly for the other isomorphism. $\square$

That is all we need to show that the category $D_{\mathcal{N}}\mathbf{C}$ with the structure described is a model of the Lambek Calculus.

FACT 5 A monoidal biclosed category $\mathbf{M}$ is a model for the Lambek Calculus $\mathbf{L}$.

The notion of a monoidal biclosed category reduces to the notion of a biclosed poset in the category of partial orders. Thus fact 5 is a generalisation of fact 1.

FACT 6 The category $D_{\mathcal{N}}\mathbf{C}$ is a sound model for the Lambek Calculus $\mathbf{L}$, meaning that if a sequent $\Gamma \vdash A$ is proved in $\mathbf{L}$, there is a morphism $f: |\Gamma| \rightarrow |A|$ in the category.

Atomic formulae are objects of $D_{\mathcal{N}}\mathbf{C}$, $\otimes$ is the non-commutative conjunction and ' $\multimap$ ' and ' $\multimap$ ' are the left and right internal-homs. The existence of the adjunction shows the validity of the inference rules for $\multimap$ and $\multimap$. $\square$

The model $D_{\mathcal{N}}\mathbf{C}$ above is an easy model to calculate with as it is made out of sets, but it is not the syntactical traditional one. Recall that as we mentioned before there are lots of other algebraic models. But the reason to give the construction above - admittedly not the easiest algebraic model - is that we can define a non-trivial of course ' $!$ ' operator in $D_{\mathcal{N}}\mathbf{C}$, more about that in the next section.

Moreover we can define additives $A \& B$ and $A \oplus B$ very easily and independently of the $\mathcal{N}$ chosen.

Proposition 3 *The category $D_N\mathbf{C}$ has binary (hence finite) products and coproducts given by*

$$A \& B = (U \times V \xleftarrow{\alpha \& \beta} X + Y)$$

$$A \oplus B = (U + V \xleftarrow{\alpha \oplus \beta} X \times Y).$$

Note that $\alpha \& \beta$ and $\alpha \oplus \beta$ actually only use either ' α ' or ' β ', as the coproduct forces us to choose, say between x and y , in the product case.

FACT 7 *The category $D_N\mathbf{C}$ is a model for the Lambek Calculus with additives **LA**.*

The attentive reader will have noticed that we could state three other (well-known) facts, to the effect that symmetric monoidal closed categories are categorical models for **LP**, monoidal biclosed categories with (finite) products and coproducts are categorical models for **LA** and symmetric monoidal closed categories with products and coproducts (*linear categories* of Girard and Lafont [1987]) are categorical models for **ILL**.

4 Modalities

We start our discussion of modalities by recalling the rules for the modality ' $!$ ' in usual *commutative* Intuitionistic Linear Logic. These are:

$$\begin{array}{cc} \frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B} \text{ (dereliction)} & \frac{\Gamma \vdash B}{\Gamma, !A \vdash B} \text{ (weakening)} \\[10pt] \frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B} \text{ (contraction)} & \frac{! \Gamma \vdash A}{! \Gamma \vdash !A} \text{ (!)} \end{array}$$

As observed by several people, the four rules above fall neatly into two pairs. One pair, namely (*weakening*) and (*contraction*), has to do with putting back into the logic, in a controlled way, contraction and weakening for specific formulae and the other pair - (*dereliction*) and (!) - make ' $!$ ' look like the (necessitation) $\Box$ modal operator of **S4**.

In categorical models one usually interprets ' $!$ ' as a functor with a natural comonad structure, while the objects $!A$ have a natural comonoid structure with respect to the tensor product of the category. The comonad structure of ' $!$ ' is used to interpret the **S4**-like rules for the modality $!$, but this interpretation also seems to use the fact that there are additives in the logic. Thus if one knows that

$$!(A \& B) \cong !A \otimes !B \text{ and } !0 \cong 1$$

and that there exists a comonad ' $!$ ', we can reason as follows. There is always a map $!A \xrightarrow{\delta} !!A$ and if $!G \xrightarrow{f} A$, then we can apply the functor ' $!$ ' to f , to get $!!G \xrightarrow{!f} !A$ and precompose it with $!G \xrightarrow{\delta} !!G$ to get $!G \rightarrow !A$, which shows the rule (!) is satisfied. For this we are assuming that $! \Gamma \vdash A$ corresponds to a morphism $!G \rightarrow A$, and that

$$! \Gamma = !G_1 \otimes !G_2 \otimes \dots \otimes !G_k \cong !(G_1 \& G_2 \& \dots \& G_k).$$

Thus we are using that 'commas' really are instances of the connective ' $\otimes$ ' and that the sequence $!G_1, !G_2, \dots, !G_n$ can be transformed into ' $!$ ' applied to the object $G_1 \& G_2 \dots \& G_n$.

The reasoning above seems to hint at the usefulness of treating additives as well as multiplicative, if one wants to re-introduce the structural rules into the system. It may not be necessary, but it certainly makes it semantically easier, as we shall see.

Some of the work done in non-commutative Linear Logic, Yetter [1990], Lincoln et al [1990] has kept a (weaker) notion of commutativity given by circular permutations. For the linguistic purposes envisaged for the Lambek calculus, the idea of allowing circular permutations is very odd. But for linguistical purposes no one is interested in having a single dualising element, admittedly at least in Yetter's paper, the main reason for the circular permutations rule.

It seems clear that formulas of the type $!A$, that we are allowed to contract and weaken over, should commute with any other formula, not only with some special ones. Also additives commute.

Hence what follows is a preliminary proposal, based mainly on Barry and Morrill [1990] and Yetter's [1990] ideas. We start with the calculus **LA** the Lambek calculus with additives, we add a modality like Yetter's ' κ ' - *kappa* - with its rules and call the new calculus **LA $_{\kappa}$** , for Lambek calculus with additives and (very restricted) permutation κ . Then we add Girard and Lafont's [1987] four rules for the modality ' $!$ ' and call the resulting system **LA $_{\kappa}!$** .

The reason for doing that is merely semantical, as we can interpret these modalities κ and $!$ in a variant of the Dialectica category $D_{\mathcal{N}}C$ described in the previous section.

Lambek Calculus with Additives and κ (**LA $_{\kappa}$**)

Axioms:

$$\begin{array}{c} A \vdash A \quad (\text{identity}) \\ \vdash I \\ \Gamma \vdash 1 \quad \Gamma, 0, \Gamma' \vdash A \end{array}$$

Structural Rule:

$$\frac{\Gamma \vdash A \quad \Gamma', A, \Delta \vdash B}{\Gamma', \Gamma, \Delta \vdash B} \quad (\text{cut})$$

Logical Rules:

Multiplicatives:

$$\begin{array}{c} (\text{unit}) \frac{\Gamma \vdash A}{\Gamma', I, \Gamma'' \vdash A} \\ \\ (\otimes) \frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, A \otimes B, \Delta \vdash C} \quad (\otimes) \frac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma, \Gamma' \vdash A \otimes B} \\ (\multimap_l) \frac{\Gamma \vdash A \quad \Gamma', B, \Delta \vdash C}{\Gamma', A \multimap B, \Gamma, \Delta \vdash C} \quad (\multimap_r) \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B} \\ (\multimap_l) \frac{\Gamma' \vdash A \quad \Gamma, B, \Delta \vdash C}{\Gamma, \Gamma', A \multimap B, \Delta \vdash C} \quad (\multimap_r) \frac{A, \Gamma \vdash B}{\Gamma \vdash A \multimap B} \end{array}$$

Additives:

$$\begin{array}{c} (\&r) \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \& B} \quad (\&l) \frac{\Gamma, A, \Delta \vdash C}{\Gamma, A \& B, \Delta \vdash C} \quad (\oplus_r) \frac{\Gamma, B, \Delta \vdash C}{\Gamma, A \& B, \Delta \vdash C} \\ (\oplus_l) \frac{\Gamma, A, \Delta \vdash C \quad \Gamma, B, \Delta \vdash C}{\Gamma, A \oplus B, \Delta \vdash C} \quad (\oplus_r) \frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B} \quad (\oplus_l) \frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B} \end{array}$$

Rules for κ (*kappa*)

$$\begin{array}{c}
\frac{\Gamma, A, \kappa B, \Gamma' \vdash C}{\Gamma, \kappa B, A, \Gamma' \vdash C} \quad (e1) \quad \frac{\Gamma, \kappa A, B, \Gamma' \vdash C}{\Gamma, B, \kappa A, \Gamma' \vdash C} \quad (e2) \\
\frac{\Gamma, A, \Delta \vdash C}{\Gamma, \kappa A, \Delta \vdash C} \quad (der_{\kappa}) \quad \frac{\kappa \Gamma \vdash C}{\kappa \Gamma \vdash \kappa C} \quad (\kappa)
\end{array}$$

Note that the difference between the systems $(\mathbf{LA}_{\kappa})$ and $\mathbf{ILL}$ is that permutation is restricted in $(\mathbf{LA}_{\kappa})$ to the formulas marked with a kappa. Also note that the rules for kappa are like **S4**-rules.

Recall that $D_{\mathcal{N}}\mathbf{C}$ models $\mathbf{LA}$ for any biclosed poset $\mathcal{N} = (N, \circ, \leq, e, \multimap, \multimap)$. In particular, we can take as $\mathcal{N}$ a (non-commutative) quantale Q . That is not really necessary, but simplifies parts of the exposition, as we can quote from Yetter's paper. Recall that by proposition 1.9 in page 44 of Yetter [1990], if Q is a quantale, there is a unique maximal central modality in Q , denoted $c_m: Q \rightarrow Q$. Use this modality c_m to define a modality κ in $D_{\mathcal{N}}\mathbf{C}$ as follows:

Definition 7 Given a category $D_{\mathcal{N}}\mathbf{C}$, where $\mathcal{N}$ is a quantale Q we define

$$\kappa: D_{\mathcal{N}}\mathbf{C} \rightarrow D_{\mathcal{N}}\mathbf{C}$$

which acts on carriers of objects of $D_{\mathcal{N}}\mathbf{C}$ as the identity endofunctor. Thus

$$\kappa(U \xrightarrow{\alpha} X) = (U \xrightarrow{c_m \alpha} X)$$

where $U \times X \xrightarrow{\alpha} Q$, a morphism in $\mathbf{C}$, is taken by κ to the composition

$$U \times X \xrightarrow{\alpha} Q \xrightarrow{c_m} Q$$

and c_m is the corresponding unique maximal central modality in Q .

Moreover it is very easy to see that:

Proposition 4 The endofunctor $\kappa: D_{\mathcal{N}}\mathbf{C} \rightarrow D_{\mathcal{N}}\mathbf{C}$ has a natural structure of comonad.

The natural transformations are induced by the 'natural transformations' in Q . All of this can be made much more concrete if we one takes a quantale of relations (cf. Brown and Gurr [1991] for instance). The point here is that, the same way the completeness of a quantale is *not* necessary to model propositional Linear Logic, the completeness of a quantale is not necessary to model the existence of an of course '!' operator. But completeness is an added bonus (extra structure), which makes some calculations simpler. Because the modality is central is makes the kappa rules sound, cf. Yetter. Thus we have a proposition:

Proposition 5 The category $D_{\mathcal{N}}\mathbf{C}$ - where $\mathcal{N}$ is a quantale Q - together with the comonad κ defined above is a categorical model for the Lambek Calculus with additives and restricted permutation or system $\mathbf{LA}_{\kappa}$.

Now we add to the system $\mathbf{LA}_{\kappa}$ the Girard's rules for the modality '!'.

Lambek Calculus with Additives, κ and '!' ($\mathbf{LA}_{\kappa}!$)

Axioms:

$$\begin{array}{c}
A \vdash A \quad (\text{identity}) \\
\vdash I \\
\Gamma \vdash 1 \quad \Gamma, 0, \Gamma' \vdash A
\end{array}$$

Structural Rule:

$$\frac{\Gamma \vdash A \quad \Gamma', A, \Delta \vdash B}{\Gamma', \Gamma, \Delta \vdash B} \text{ (cut)}$$

Logical Rules:
Multiplicatives:

$$\text{(unit)} \frac{\Gamma \vdash A}{\Gamma', I, \Gamma'' \vdash A}$$

$$\begin{array}{ll} (\otimes) \frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, A \otimes B, \Delta \vdash C} & (\otimes) \frac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma, \Gamma' \vdash A \otimes B} \\ (\multimap_l) \frac{\Gamma \vdash A \quad \Gamma', B, \Delta \vdash C}{\Gamma', A \multimap B, \Gamma, \Delta \vdash C} & (\multimap_r) \frac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma, A \vdash B} \\ (\multimap_l) \frac{\Gamma \vdash A \quad \Gamma', B, \Delta \vdash C}{\Gamma', \Gamma, A \multimap B, \Delta \vdash C} & (\multimap_r) \frac{\Gamma \vdash A \multimap B}{A, \Gamma \vdash B} \end{array}$$

Additives:

$$\begin{array}{ll} (\&r) \frac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma \vdash A \& B} & (\&l) \frac{\Gamma, A, \Delta \vdash C}{\Gamma, A \& B, \Delta \vdash C} \\ (\oplus_l) \frac{\Gamma, A, \Delta \vdash C \quad \Gamma, B, \Delta \vdash C}{\Gamma, A \oplus B, \Delta \vdash C} & (\oplus_r) \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \oplus B} \end{array}$$

Rules for κ (kappa)

$$\begin{array}{ll} \frac{\Gamma, A, \kappa B, \Gamma' \vdash C}{\Gamma, \kappa B, A, \Gamma' \vdash C} \text{ (e1)} & \frac{\Gamma, \kappa A, B, \Gamma' \vdash C}{\Gamma, B, \kappa A, \Gamma' \vdash C} \text{ (e2)} \\ \frac{\Gamma, \kappa B, A, \Gamma' \vdash C}{\Gamma, A, \Delta \vdash C} \text{ (der}_\kappa) & \frac{\Gamma, B, \kappa A, \Gamma' \vdash C}{\kappa \Gamma \vdash C} \text{ (\kappa)} \\ \Gamma, \kappa A, \Delta \vdash C & \kappa \Gamma \vdash \kappa C \end{array}$$

Rules for ! (of course!)

$$\begin{array}{ll} \frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B} \text{ (dereliction)} & \frac{\Gamma \vdash B}{\Gamma, !A \vdash B} \text{ (weakening)} \\ \frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B} \text{ (contraction)} & \frac{! \Gamma \vdash A}{! \Gamma \vdash !A} \text{ (!)} \end{array}$$

Rules connecting κ and !⁴

$$\frac{\Gamma, !A, \Gamma' \vdash B}{\Gamma, \kappa A, \Gamma' \vdash B} \text{ (!}\kappa_l) \quad \frac{\Gamma \vdash \kappa C}{\Gamma \vdash !C} \text{ (!}\kappa_r)$$

The next step is to define a modality

$$! : D_{\mathcal{N}}\mathcal{C} \rightarrow D_{\mathcal{N}}\mathcal{C}$$

which given an object $A = (U \xrightarrow{\alpha} X)$ returns an object $(U \xrightarrow{! \alpha} X^{*U})$. This can be shown to be an endofunctor with a natural comonad structure inherited from the monoidal structure of X^{*U} , cf. de Paiva [1991]. There are lots of details to check, but as the category $\mathcal{C}$ is **Sets**, which is cartesian closed, the constructions above work and give you a comonad and comonoids in $D_{\mathcal{N}}\mathcal{C}$.

After checking all the details, we should have:

Proposition 6 *The category $D_{\mathcal{N}C}$ is a categorical model for the Lambek Calculus with additives, restricted permutation κ and Girard modality $!$ (of course!).*

As we mentioned before, this is all done semantically, it would be very interesting to check what happens with the Proof Theory of the systems $(\mathbf{LA}_{\kappa})$ and $(\mathbf{LA}_{\kappa!})$. On the one hand, the proof theory should not be too complicated, as we do not have the *par* operator interacting with linear implication, but on the other hand, people interested in proof-nets find weak forms of commutativity like (*cycling*) extremely natural.

Conclusions

If one takes the linguistics motivation seriously, the modality κ above does not seem exactly what is wanted. There might be some use in having an operator κ , but perhaps it would be more useful to have a modality which did a *very* limited amount of permutation. In particular it would be interesting to produce rules for a modality that applied to two formulas A and B said something like *It is known that A and B commute*. The modality κ above says something like κA *commutes with everything*.

There were some other attempts at less constraining permutative modalities and the reader should consult Barry et al for several possibilities. But the interesting aspect of the modality κ is that we can model it without too much difficulty. Also interesting is that it makes a sort of “bridge” between the commutative world and the non-commutative one. This bridge is of the same type as the bridge one has (“of course!”) between resource-aware and traditional systems. Maybe one should investigate κ ’s proof theory, but our main concern at the moment is with *term calculi*, similar to the λ -calculus, but more accurate, that would model “exactly” other logical systems in the spirit of Linear Logic.

Notes

1. The whole issue of what a logic is *philosophically* is being eschewed here though.
2. I recently became aware of Abrusci’s ([1990]) paper where starting from the same perspective (Linear Logic) he explains the Lindenbaum algebra construction.
3. Dialectica categories were introduced by the author in dePaiva [1987] as an internal categorical version of Gödel’s Dialectica Interpretation. They turned out to be also an interesting family of models of Intuitionistic Linear Logic.
4. The necessity of rule $(! \kappa_1)$ to (eventually perhaps) obtain cut-elimination was explained to me by Dirk Roorda.

References

- Abrusci, V.M.: 1990, 'A Comparison between Lambek Syntactic Calculus and Intuitionistic Propositional Linear Logic', *Zeitschr. f. Math, Logik und Grundlagen d. Math* 36 11-15
- Bar, M.: 1979, **-Autonomous Categories* LNM 752, Springer-Verlag
- Bellin, G.: 1990, *Mechanizing Proof-Theory: Resource-Aware Logics and Proof-transformations to Extract Implicit Information* Ph. D thesis, Stanford Computer Science Dept. TR STAN-CS-90-1319
- Benthem, J. van: 1988, 'The Lambek Calculus', *Categorical Grammars and Natural Language Structures*, eds R. Oehrle, E. Bach and D. Wheeler
- Barry, G and G. Morrill.: 1990, *Studies in Categorical Grammar* Edinburgh Working Papers in Cognitive Science, vol 5
- Brown, C. and D. Gurr.: 1991, 'A Representation Theorem for Quantales', manuscript
- Buszkowski, W.: 1986, 'Completeness Results for Lambek Syntactic Calculus' *Zeitschr. f. Math, Logik und Grundlagen d. Math* 32 13-28
- Eilenberg, S. and M. Kelly: 1965, 'Closed Categories' *Proc. Conf. on Categorical Algebra, La Jolla* 421-562
- Flagg, R.C.: 1990, 'A Generalised Logic', manuscript
- Girard, J-Y. and Y. Lafont: 1987, 'Linear Logic and Lazy Computation' *TAPSOFT-87*
- Girard, J-Y.: 1987, 'Linear Logic' *Theoretical Computer Science* 501-102
- Hyland, J.M.E. and V.C.V de Paiva: 1990, 'Lineales', manuscript
- Joyal, A. and R. Street: 1986, 'Braided Monoidal Categories', *Macquarie Mathematics Report* 860081
- Lafont, Y. and T. Streicher: 1991, 'Games Semantics for Linear Logic', *LICS 1991*
- Ketonen, J. and R. Weyrauch: 1986, 'A Decidable Fragment of Predicate Calculus' *Theoretical Computer Science* 32 297-307
- Lambek, J.: 1958, 'The Mathematics of Sentence Structure' *Amer. Math. Monthly* 65 154-169
- Lambek, J.: 1988, 'Categorical and Categorical Grammars' in *Categorical Grammars and Natural Language Structures*, (eds) R. Oehrle, E. Bach and D. Wheeler
- Lambek, J.: 1987, 'Multicategories Revisited' in *Categories in Computer Science and Logic - 87* eds. J. Gray and A. Scedrov, 1989
- Lawvere, F.W.: 1973, 'Metric Spaces, Generalized Logic, and Closed Categories' in *Rendiconti del Seminario Matematico e Fisico di Milano*, 43
- Lincoln, P. MITCHELL, J, SCEDROV, A and N. SHANKAR: 1990, 'Decision Problems for Propositional Linear Logic' in *Proc. of FOCS - 90*
- MacLane, S.: 1971, 'Categories for the Working Mathematician' Springer-Verlag
- Moorgat, M.: 1990, 'The Logic of Discontinuous Type Constructors', Unpublished Manuscript.
- Morril, G.: 1990, 'Grammar and Logical Types' in *Studies in Categorical Grammar* Edinburgh Working Papers in Cognitive Science, vol 5
- Morril, G. Leslie, N. Hepple, M and G. Barry: 1990, 'Categorical Deductions and Structural Operations' in *Studies in Categorical Grammar* Edinburgh Working Papers in Cognitive Science, vol 5
- de Paiva, V. C. V.: 1987 'The Dialectica Categories' in *Categories in Computer Science and Logic*, Proc. Summer Research Conf. AMS 92, Boulder, CO, June 1987. Published as vol. 92 of AMS Publications in 1989.
- de Paiva, V. C. V.: 1989 'A Dialectica-like Model of Linear Logic' *Categorical Logic in Computer Science*, Proc. Manchestre 1989, LNCS vol 389, Springer-Verlag
- de Paiva, V. C. V.: 1988 'The Dialectica Categories' Ph. D Thesis DPMMS, University of Cambridge, published as Technical Report 213 from the Computer Lab, January 1991
- Roorda, D.: 1991 'Resource Logics: Proof-theoretical Investigations' Ph. D Thesis, University van Amsterdam, September 1991
- Takeuti, G.: 1975 *Proof theory* Studies in Logic and the Foundations of Mathematics vol 81, North-Holland
- Yetter, D.: 1990 'Quantales and (non-commutative) Linear Logic' in *Journal of Symbolic Logic* 55

THE ROLE OF VARIABLES IN THE FORMAL DESCRIPTION OF NATURAL LANGUAGE

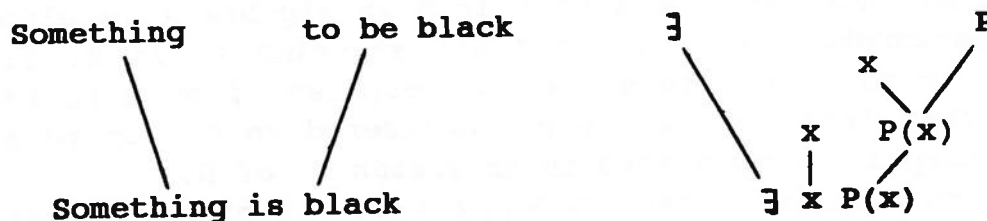
Jaroslav Peregrin^o

Institute for Theoretical and Computational Linguistics
Faculty of Philosophy, Charles' University,
Prague, Czechoslovakia.

1 What Do Variables Stand For?

The languages of formal logic have become an indispensable tool of semantic analysis of natural language: formal expressions function as 'regimentations' of natural ones suppressing their irrelevant features and bringing to light relevant regularities¹. But the direct correspondence between the schematizing formal language and the schematized natural one is limited: besides constants and, more generally, closed expressions, which can be considered as standing directly for words and expressions of natural language there are also variables and open expressions which have no such natural language counterparts².

Let us consider the grammatical structure of the sentence *something is black* and that of its logical schematization $\exists xB(x)$.



We may equate *something* with $\exists$ and *to be black* with B , but there is nothing to be equated with x and $B(x)$. So the question is: what x and $B(x)$ of the right side,

which do not have any direct counterpart on the left side, in fact are?

2 Interpretations and Derivers

The main thesis of this paper is that addition of variables to a language can be adequately considered as a disguised addition of grammatical rules. This can be well elucidated within the framework introduced by Janssen(1983), in the framework identifying language with many-sorted algebra and semantic interpretation with homomorphism.

What is usually considered as a semantic interpretation becomes in such a framework a homomorphism from the many-sorted algebra N , capturing the grammatical structure of natural language, into a (many-sorted) denotational algebra D . The interpretation is, however, usually decomposed into two subsequent homomorphisms: (i) a homomorphism from a logical language L into D and (ii) a homomorphism from N into an extension L^* of L . We shall call (i) *denotation* and (ii) *regimentation*³. To decompose interpretation into regimentation and denotation is a kind of trick: it allows for D being considerably "simpler" (in the sense of having fewer operators) than N . Such a trick, however, clearly works only when a special kind of relationship obtains between L and L^* : when any homomorphism from L into an algebra A is also a homomorphism from L^* into an extension A^* of A . If this is the case, then the homomorphisms from N to L^* and that from L to D may be considered to add up to a homomorphism from N into an extension D^* of D .

Let us call two algebras, which have the property that any homomorphism from one of them is in the same time a homomorphism from the other, (*congruentially*) *equivalent*⁴. It is clear that as long as we restrict

our interest to interpretations and as long as we consider interpretation as homomorphism, we need not make any difference between congruentially equivalent algebras. What we need decomposing interpretation into regimentation and denotation is that the range of regimentation, L^* , be congruentially equivalent with the domain of denotation, L .

A deriver, a method of obtaining algebras from algebras, is called safe by Janssen, if $D(A)$ is congruentially equivalent with A for every algebra A . So what we need is that the range of regimentation is the result of applying a safe deriver to the domain of denotation, that $L^* = D(L)$ for a safe deriver D . What Janssen proves is that addition of polynomial functions as new operators to an algebra is a safe deriver.

What we are going to show are two things: first, that addition of variables (together with some form of lambda abstraction) to an algebra is equivalent to adding certain kind of operators; and, second, that addition of operators of this kind is a deriver, which, although not safe in the sense of Janssen, is safe in a weaker, but sufficient sense. These two facts then clearly document that addition of variables can be plausibly considered a peculiar type of addition of operators.

3 Applicativity

It seems to be clear that not every algebra is acceptable as the target of denotation. But if this is the case, then there is no need to consider all possible homomorphisms as denotations; it is enough to restrict our attention to homomorphisms the target algebras of which fall into a restricted class. Let us call a deriver D safe w.r.t. a class C of algebras iff for any algebra A , any algebra $B \in C$ and any homomorphism from A to B there exists an algebra B' such that h is a

homomorphism from $D(A)$ to B' : then the property of a deriver we have to require is not safety, but only safety w.r.t. the class of "acceptable" algebras.

Let us call an n -ary function F *applicator* iff there exists an n , $0 < i \leq n$, such that for every $\langle x_1, \dots, x_n \rangle$ from the domain of F it holds that x_i is an $(n-1)$ -ary function such that

$F(x_1, \dots, x_n) = x_i(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$. Now the history of modern logic and formal semantics displays a clear tendency towards denotational algebras the operators of which are applicators. This is immediately obvious in the case when the intervening logical language is a propositional calculus (not only the classical one)⁵; however, if we accept the view of variables proposed here, it turns to be adequate also for predicate calculi. Let us call the algebras all the operators of which are applicators *applicative*.

We are now going to define a generalization of the notion of polynomial function, called *abstractive function*, and we are going to show that addition of abstractive operators is safe w.r.t. the class of applicative algebras. Then we shall show that addition of variables with the apparatus of lambda abstraction is equivalent to addition of abstractions.

4 The Framework

Let us summarize the framework. It is essentially due to Janssen(1983), although the terminology adopted here does not follow this author in all details (we, for example, prefer to call sort what Janssen calls *carrier of sort*, etc.).

DEFINITION 4.1. Let $S = \langle S_i \rangle_{i \in I}$ be a collection of sets. If $i_1, \dots, i_n, i_{n+1}$ all belong to I , then the set of all the functions the domains of which equal $S_{i_1} \times \dots \times S_{i_n}$ and the ranges of which are included in $S_{i_{n+1}}$ will be called a *type over S* ; it will be denoted by $[S_{i_1} \times \dots \times S_{i_n} \rightarrow S_{i_{n+1}}]$.

DEFINITION 4.2. A many-sorted algebra (msa) is an ordered pair $A = \langle C, O \rangle$, where (i) $C = \langle C_i \rangle_{i \in I}$ is a family of pairwise disjoint sets (the sorts or categories of A); and (ii) $O = \langle O_j \rangle_{j \in J}$ is a family of functions (the operators or rules of A) such that O_j is a function of a type over C for every $j \in J$. The union of C will be called the carrier of A.

DEFINITION 4.3. Let $A = \langle \langle C_i \rangle_{i \in I}, \langle O_j \rangle_{j \in J} \rangle$ and $A' = \langle \langle C'_i \rangle_{i \in I}, \langle O'_j \rangle_{j \in J} \rangle$ be msa's. Let σ be a bijective assignment of sorts of A' to those of A and τ a bijective assignment of operators of A' to those of A. A and A' are called $\langle \sigma, \tau \rangle$ -similar iff $\tau(O_j)$ is of the type $[\sigma(C_{i_1}) \times \dots \times \sigma(C_{i_n}) \rightarrow \sigma(C_{i_{n+1}})]$ whenever O_j is of the type $[C_{i_1} \times \dots \times C_{i_n} \rightarrow C_{i_{n+1}}]$. A and A' are called similar iff they are $\langle \sigma, \tau \rangle$ -similar for some σ and τ .

DEFINITION 4.4. Let $A = \langle \langle C_i \rangle_{i \in I}, \langle O_j \rangle_{j \in J} \rangle$ and $A' = \langle \langle C'_i \rangle_{i \in I}, \langle O'_j \rangle_{j \in J} \rangle$ be $\langle \sigma, \tau \rangle$ -similar msa's. A function F is called a $\langle \sigma, \tau \rangle$ -homomorphism from A to A' if F is the union of a collection $\langle F_i \rangle_{i \in I}$ of functions such that F_i is from C_i to $\sigma(C_i)$ for every $i \in I$, and, moreover, for any O_j and for any $a_1, \dots, a_n$ from the domain of O_j it holds that $F(O_j(a_1, \dots, a_n)) = \tau(O_j)(F(a_1), \dots, F(a_n))$. If F is a $\langle \sigma, \tau \rangle$ -homomorphism from A to A', then $\sigma(C_i)$ will be also denoted by $[C_i]_F$. F is a homomorphism from A to A' (schematically: $F \in \text{Hom}(A, A')$) if it is a $\langle \sigma, \tau \rangle$ -homomorphism from A to A' for some σ and τ . A surjective homomorphism F from A to A' is an epimorphism from A to A' ($F \in \text{Epi}(A, A')$).

DEFINITION 4.5. Let $A = \langle C, O \rangle$ be a msa. An n-ary function F will be called projection over A iff F is of a type over C and if there exists an i such that $F(a_1, \dots, a_n) = a_i$ for every $\langle a_1, \dots, a_n \rangle$ from the domain of F. F will be called constant function over A iff it is of a type over C and if there exists an e such that $F(a_1, \dots, a_n) = e$ for every $\langle a_1, \dots, a_n \rangle$ from the domain of F.

DEFINITION 4.6. Polynomial functions over A are functions defined as follows:

- (i) if P is a projection over A, then P is a polynomial function over A;
- (ii) if P is a constant function over A, then P is a polynomial function over A;
- (iii) if O is an operator of A of a type $[C_1 \times \dots \times C_n \rightarrow C]$ and if $P_1, \dots, P_n$ are polynomial functions over A of the respective types $[C'_1 \times \dots \times C'_k \rightarrow C_1], \dots, [C'_1 \times \dots \times C'_k \rightarrow C_n]$, then the function P of the type $[C'_1 \times \dots \times C'_k \rightarrow C]$ such that $P(a_1, \dots, a_k) = O(P_1(a_1, \dots, a_k), \dots, P_n(a_1, \dots, a_k))$ for every $\langle a_1, \dots, a_k \rangle \in C'_1 \times \dots \times C'_k$ is a polynomial function over A.
- (iv) the functions characterized by (i)-(iii) are the only polynomial functions over A.

A msa which results from an addition of operators polynomial over A to A will be called a *polynomial extension* of A.

That every operator of A is a polynomial over A follows from the fact that $O(a_1, \dots, a_n) = O(P_1(a_1, \dots, a_n), \dots, P_n(a_1, \dots, a_n))$ for the projections $P_1, \dots, P_n$ such that $P_i(a_1, \dots, a_n) = a_i$ for $i=1, \dots, n$.

DEFINITION 4.7. A *deriver* is a method of obtaining algebras from algebras. A deriver D is called *safe* iff for every A, every B, and every $h \in \text{Epi}(A, B)$ there is a unique algebra B^* such that $h \in \text{Epi}(D(A), B^*)$. If C is a class of algebras, then a deriver D is called *C-safe* iff for every A, every $B \in C$, and every $h \in \text{Epi}(A, B)$ there is a unique algebra B^* such that $h \in \text{Epi}(D(A), B^*)$.

THEOREM 4.8. If $h \in \text{Epi}(A, B)$ and A^* is a polynomial extension of A, then there exists a unique extension B^* of B such that $h \in \text{Epi}(A^*, B^*)$.

PROOF. Let H be a homomorphism from A to B and let P be a polynomial function over A. Let A' be A with P added. To prove the theorem it is evidently enough to prove that there exists a unique extension B' of B such that $h \in \text{Epi}(A', B')$. But to prove this it is in turn evidently enough to prove that the kernel of H is a congruence on A' , i.e. that if $H(a_1) = H(b_1), \dots, H(a_k) = H(b_k)$, then $H(P(a_1, \dots, a_k)) = H(P(b_1, \dots, b_k))$. Let us prove this by induction. Basic cases (i)-(ii) are clear. Let now P be as in (iii) and let $H(a_1) = H(b_1), \dots, H(a_k) = H(b_k)$. Let σ and τ be such that H is a $\langle \sigma, \tau \rangle$ -homomorphism. $H(P(a_1, \dots, a_k))$ thus equals $H(O(P_1(a_1, \dots, a_k), \dots, P_n(a_1, \dots, a_k)))$ and this, in turn, equals $\tau(O(H(P_1(a_1, \dots, a_k)), \dots, H(P_n(a_1, \dots, a_k))))$. But $H(P_1(a_1, \dots, a_k)) = H(P_1(b_1, \dots, b_k))$, $\dots$, $H(P_n(a_1, \dots, a_k)) = H(P_n(b_1, \dots, b_k))$ according to the inductive assumption, and so this in turn equals $\tau(O(H(P_1(b_1, \dots, b_k)), \dots, H(P_n(b_1, \dots, b_k))))$ and hence $H(O(P_1(b_1, \dots, b_k), \dots, P_n(b_1, \dots, b_k)))$ and finally $H(P(b_1, \dots, b_k))$.
[.]

COROLLARY 4.9. (Janssen) Polynomial extension is a safe deriver.

5 Applicative Algebras

An applicative algebra is an algebra the sorts of which form a functional hierarchy and the operators of

which consist in the application of the function-elements to other elements.

DEFINITION 5.10. A set is called *ground* if its members are not functions. An algebra A is called *applicative* iff every its sort is either ground or a type over the sorts of A , and if, moreover, for any operator O of A of the type $[C_1x \dots xC_n \rightarrow C_{n+1}]$ there exists an i such that C_i is $[C_1x \dots xC_{i-1}xC_{i+1}x \dots xC_n \rightarrow C_{n+1}]$ and $O(a_1, \dots, a_n) = a_i(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$ for every $\langle a_1, \dots, a_n \rangle \in C_1x \dots xC_n$. C_i is then called the *functor sort* of O .

DEFINITION 5.11. A homomorphism $h \in \text{Hom}(A, A')$ is called *applicative* iff A' is an applicative algebra. If H is an applicative homomorphism from A , then for any operator O of the type $[C_1x \dots xC_n \rightarrow C]$ there by definition exists an $i \in \{1, \dots, n\}$ such that $H(O(a_1, \dots, a_n)) = H(a_i)(H(a_1), \dots, H(a_{i-1}), H(a_{i+1}), \dots, H(a_n))$ for all $\langle a_1, \dots, a_n \rangle \in C_1x \dots xC_n$; we shall call C_i the *H-functor sort* of O .

Let us consider the algebra $L = \langle \langle C_i \rangle_{i \in \{s, o1, o2, p, t, q\}}, \langle R_j \rangle_{j \in \{o1, o2, p, q\}} \rangle$ where R_{o1} is of the type $[C_{o1}xC_s \rightarrow C_s]$, R_{o2} of the type $[C_{o2}xC_sxC_s \rightarrow C_s]$, $R_p [C_pxC_t \rightarrow C_s]$ and $R_q [C_qxC_p \rightarrow C_s]$. The sorts $C_s, C_{o1}, C_{o2}, C_t, C_p, C_q$ can be informally considered as sentences, one-place sentential operators, two-place sentential operators, terms, (unary) predicates and quantifiers; the operators as sentential modification, sentential junction, predication and quantification. Let us further, consider the algebra $D = \langle \langle D_i \rangle_{i \in \{s, o1, o2, p, t, q\}}, \langle R'_j \rangle_{j \in \{o1, o2, p, q\}} \rangle$ where R'_{o1} is of the type $[C_{o1}xC_s \rightarrow C_s]$, R'_{o2} of the type $[C_{o2}xC_sxC_s \rightarrow C_s]$, $R'_p [C_pxC_t \rightarrow C_s]$ and $R'_q [C_qxC_p \rightarrow C_s]$; $D_s = \{T, F\}$, $D_{o1} = [D_s \rightarrow D_s]$, $D_{o2} = [D_sxD_s \rightarrow D_s]$, $D_p = [D_t \rightarrow D_s]$, $D_q = [D_p \rightarrow D_s]$; and such that $R'_{o1}(o1, s1) = o1(s1)$, $R'_{o2}(o2, s1, s2) = o2(s1, s2)$, $R'_p(p, t) = p(t)$, $R'_q(q, p) = q(p)$ for every $o1 \in O1$, $o2 \in O2$, $s1, s2 \in S$, $p \in P$, $t \in T$. The sorts $D_s, D_{o1}, D_{o2}, D_t, D_p, D_q$ can be considered as truth values, unary truth-functions, binary truth-functions, individuals, classes of individuals, classes of classes of individuals. D is evidently an applicative algebra; the functor sort of R'_j is D_j (for $j \in \{O1, O2, P, Q\}$). L and D are $\langle \sigma, \tau \rangle$ -similar, where $\sigma(C_i) = D_i$ and $\tau(R_j) = R'_j$.

If H is a $\langle \sigma, \tau \rangle$ -homomorphism from L into D , then the H -functor category of the rule R_j is C_j (for $j \in \{01, 02, P, Q\}$).

If A is applicative, then there is a founded partial ordering of its sorts such that the functor sort of every operator is greater than both every other sort constituting the domain of the operator, and the sort including the range of the operator. This follows from the fact that the relation of set membership is founded. Any assignment of sorts to operations having this property will be called *stratification*, and an algebra for which there is a stratification will be called *stratified*.

DEFINITION 5.12. Let F be an injective function which to any operator O of A assigns a category C of A ; let there, moreover, exist a founded partial ordering $>$ of the categories of A such that for any operator O of the type $[C_1 x \dots x C_n \rightarrow C_{n+1}]$ it holds that $F(O) = C_i$ for some $i \in \{1 \dots n\}$ and $C_i > C_j$ for $j \in \{1 \dots, i-1, i+1, \dots, n, n+1\}$. Then F will be called a *stratification* of A . If there exists a stratification of A , then A will be called *stratified*. If $H \in \text{Hom}(A, B)$ for an applicative B , then the stratification S for which $S(O)$ is the H -functor category of O for every O will be called *H-stratification*; H will be called *S-applicative*.

Not only is every applicative algebra stratified, but clearly also every algebra which has an applicative interpretation, i.e. which is similar to an applicative algebra. As there surely exist algebras which are not stratified, the existence of an applicative homomorphism from A (in contrast to existence of a homomorphism *simpliciter*) tells us something about the structure of A . Stratification can be considered a kind of a generalization of applicativity; if we have a stratification of an algebra, we can think of its operators as "applications".

Let us return to the algebra L . The function S_1 which assigns the category C_j to the rule R_j (for $j \in \{01, 02, P, Q\}$) is a stratification of A ; A is hence stratified. H is S_1 -applicative. Having S_1 in mind, we can speak of R_P as an "application" of a member of C_P .

to a member of C_T ; it is, however, not an application in the set-theoretical sense. (R'_P , on the contrary, is an application in the set-theoretical sense.) S_1 is, however, not the only stratification of A ; another one is, for example, the function S_2 which assigns C_s to R_{01} , C_{02} to R_{02} , C_T to R_P and C_P to R_Q .

An operator of an algebra, or, more generally, a polynom over the algebra, applied to arguments $a_1, \dots, a_n$ yields a value e . We can imagine this to be decomposed into two steps: we first use some of the $a_1, \dots, a_n$ to yield an intermediary value, and then, in the second step, we use this value together with those of $a_1, \dots, a_n$ which have not been used so far to yield e . The first step can be considered as performing the original operation with leaving some "holes" in the result, and the second step as filling these holes. If the second step consists in the "application" of the intermediary value to the remaining elements, then the first step is what we are going to call an abstractive function over A . As the concept of "application" is relative to stratification, so too will be the concept of abstractive function will be.

DEFINITION 5.13. Let A be a msa and S its stratification. S -abstractive functions over A are functions defined as follows:

- (i) if P is a projection over A , then P is an S -abstractive function over A ;
- (ii) if P is a constant function over A , then P is an S -abstractive function over A ;
- (iii) if O is an operator of A of the type $[C_1x \dots xC_n \rightarrow C]$ and if $P_1, \dots, P_n$ are S -abstractive functions over A of the respective types $[C'_1x \dots xC'_k \rightarrow C_1], \dots, [C'_1x \dots xC'_k \rightarrow C_n]$, then the function P of the type $[C'_1x \dots xC'_k \rightarrow C]$ such that $P(a_1, \dots, a_k) = O(P_1(a_1, \dots, a_k), \dots, P_n(a_1, \dots, a_k))$ for every $\langle a_1, \dots, a_k \rangle \in C'_1x \dots xC'_k$ is an S -abstractive function over A .
- (iv) if O is an operator of A of the type $[C'_1x \dots xC'_k \rightarrow C]$, $S(O) = C'_1$, and P' is an S -abstractive function of the type $[C'_1x \dots xC'_{i-1}xC_1x \dots xC_nxC'_{i+1}x \dots xC'_k \rightarrow C]$ over A , then any function P of the type $[C_1x \dots xC_n \rightarrow C'_1]$ such that for every $\langle b_1x \dots xb_{i-1}xa_1x \dots xa_nxb_{i+1}x \dots xb_k \rangle \in C'_1x \dots xC'_{i-1}xC_1x \dots xC_nxC'_{i+1}x \dots xC'_k$ it holds that $O(b_1, \dots, b_{i-1}, P(a_1, \dots, a_n), b_{i+1}, \dots, b_k) =$

$P'(b_1, \dots, b_{i-1}, a_1, \dots, a_n, b_{i+1}, \dots, b_k)$ is an S-abstractive function over A.
 (v) the functions characterized by (i)-(iv) are the only S-abstractive functions over A.
 A msa which results from an addition of operators S-abstractive over A to A will be called an (S-)abstractive extension of A.

A function which to any $o \in C_{O1}$, $p \in C_P$, $t \in C_T$ assigns the value $R_{O1}(o, R_P(p, t))$ is an S-abstractive function over L for every stratification S of L; it is, moreover, a polynomial function over L. Let R_{O1} be a function which to any $o \in C_{O1}$ and $p \in C_P$ assigns such $p' \in C_P$ that $R_P(p', t) = R_{O1}(o, R_P(p, t))$ for every $t \in T$; then R_{O1} is an S1-abstractive function over L; it is not an S2-abstractive function over A; it is not a polynomial function over A.

Now we are going to prove a theorem parallel to the theorem 4.8; the theorem indicates that the role of abstractive functions w.r.t. applicative interpretations is the same as that of polynomial functions w.r.t. interpretations in general.

THEOREM 5.14. If $H \in \text{Epi}(A, B)$, B is applicative, and A^* is an abstractive extension of A, then there exists a unique extension B^* of B such that $H \in \text{Epi}(A^*, B^*)$.
PROOF. Let S be a stratification of A and $H \in (A, B)$; let, moreover, P be an S-abstractive function over A. To prove the theorem it is again evidently enough to prove that if $H(a_1) = H(b_1), \dots, H(a_k) = H(b_k)$, then $H(P(a_1, \dots, a_k)) = H(P(b_1, \dots, b_k))$. Proof of the cases (i)-(ii) is again evident; proof of the case (iii) is analogous to that of theorem 4.8. Let thus P be as in (iv) and let $H(a_1) = H(b_1), \dots, H(a_n) = H(b_n)$. As $S(O) = C'_1$ and as O is of the type $[C'_1, \dots, C'_k \rightarrow C]$, $[C'_1]_H = [[C'_1]_H]_H \times \dots \times [C'_{i-1}]_H \times [C'_{i+1}]_H \times \dots \times [C'_k]_H \rightarrow [C]_H$, and hence $H(P(a_1, \dots, a_n))$ and $H(P(b_1, \dots, b_n))$ are identical iff they have the same values for all $\langle h_1, \dots, h_{i-1}, h_{i+1}, \dots, h_k \rangle$ from $[C'_1]_H \times \dots \times [C'_{i-1}]_H \times [C'_{i+1}]_H \times \dots \times [C'_k]_H$. So let $\langle h_1, \dots, h_{i-1}, h_{i+1}, \dots, h_k \rangle$ be arbitrary from $[C'_1]_H \times \dots \times [C'_{i-1}]_H \times [C'_{i+1}]_H \times \dots \times [C'_k]_H$. As H is a surjection, there exist some $c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_k$ such that $H(c_1) = h_1, \dots, H(c_{i-1}) = h_{i-1}, H(c_{i+1}) = h_{i+1}, \dots, H(c_k) = h_k$, and $H(P(a_1, \dots, a_n))(\langle h_1, \dots, h_{i-1}, h_{i+1}, \dots, h_k \rangle)$ thus equals $H(P(a_1, \dots, a_n))(H(c_1), \dots, H(c_{i-1}), H(c_{i+1}), \dots, H(c_k))$. This is in turn equal to $H(O(c_1, \dots, c_{i-1}, P(a_1, \dots, a_n), c_{i+1}, \dots, c_k))$ and hence to $H(P'(c_1, \dots, c_{i-1}, a_1, \dots, a_n, c_{i+1}, \dots, c_k))$. However, this

is, according to the inductive assumption, equal to $H(P'(c_1, \dots, c_{i-1}, b_1, \dots, b_n, c_{i+1}, \dots, c_k))$, hence to $H(O(c_1, \dots, c_{i-1}, P(b_1, \dots, b_n, c_{i+1}, \dots, c_k)))$, further to $H(P(b_1, \dots, b_n))(H(c_1), \dots, H(c_{i-1}), H(c_{i+1}), \dots, H(c_k))$ and hence to $H(P(b_1, \dots, b_n))(h_1, \dots, h_{i-1}, h_{i+1}, \dots, h_k)$.
 $[]$.

COROLLARY 5.15. Abstractive extension is a deriver safe w.r.t. the class of applicative algebras.

We have seen above that the function R_{O1} is $S1$ -abstractive. Also the following two functions are evidently $S1$ -abstractive: R_s of the type $[S \rightarrow P]$ such that for every $s \in S$ $R_s(s) = p$ such that for every $t \in T$ $R_P(p, t) = s$; and R_{O2} of the type $[O2 \times P \times P \rightarrow P]$ such that for every $o \in O2$, $p_1, p_2 \in P$ $R_{O2}(o, p_1, p_2) = p'$ such that for every $t \in T$ $R_P(p', t) = R_{O2}(o, R_P(p_1, t), R_P(p_2, t))$. So the

$\begin{matrix} a & l & g & e & b & r & a \end{matrix}$

$L^A = \langle \langle C_i \rangle_{i \in \{S, O1, O2, P, T, Q\}}, \langle R_j \rangle_{j \in \{O1, O2, P, Q, S, O1', O2'\}} \rangle$ is an abstractive extension of L , and hence is every applicative homomorphism from L also a homomorphism from L^A .

6 Variables and Abstraction

To be able to show that an extension of a language by means of variables can be identified with the addition of abstractive rules we must first articulate what the addition of variables means. It seems that to be able to speak about an extension of an algebra s about an addition of variables, at least the following two conditions must be fulfilled: (i) the extension is free; (ii) the new elements can be substituted for by the old ones. Let us first turn our attention to (i).

DEFINITION 6.16. Let $A = \langle \langle C_i \rangle_{i \in I}, \langle O_j \rangle_{j \in J} \rangle$ be a msa and let $X = \langle X_i \rangle_{i \in I}$ be a collection of sets such that $C_i \cap X_i = \emptyset$ for every $i \in I$. The free extension A_X of A generated by X (i.e. the algebra freely generated by $\langle \langle C_i \cup X_i \rangle_{i \in I}, \langle O_j \rangle_{j \in J} \rangle$), will be called a variable extension of A . If A_X is a variable extension of A , then those elements of its carrier which are elements

of the carrier of A will be called *close*, the other elements *open*. Members of X_i (for every $i \in I$) are called *variables*. Note that any open element has an "unique analysis" in the sense that there exists at most one operator O and one n -tuple of elements $a_1, \dots, a_n$ such that $e = O(a_1, \dots, a_n)$ (if e is not a variable, then there exists precisely one, if it is, then no).

The algebra L^V freely generated by $\langle \langle C_i \rangle_{i \in \{s, o_1, o_2, p, t, q\}}, \langle R_j \rangle_{j \in \{o_1, o_2, p, q\}} \rangle$, such that $C_T = C_T \{x\}$ is a variable extension of the algebra L ; its carrier contains in addition to the elements of the carrier of L also the variable x and other new ("open") elements: the results of rules for arguments including x .

DEFINITION 6.17. Let A_x be a variable extension of A . A *substitution* is an sort-preserving assignment of closed elements to (not necessarily all) variables. A substitution which assigns a_i to x_i for $i=1, \dots, n$ will be denoted by $x_1/a_1, \dots, x_n/a_n$. For any element e of the carrier of A_x and any substitution S we define the S -instance $e[S]$ of e as follows:
 (i) if e is closed, then $e[S] = e$;
 (ii) if e is a variable, then $e[S] = S(e)$ when e is in the domain of S , otherwise $e[S] = e$;
 (iii) if e is open and if it is not a variable, then $e[S] = O(a_1[S], \dots, a_n[S])$ for such O and $a_1, \dots, a_n$ that $e = O(a_1, \dots, a_n)$.

The only nontrivial kind of substitution of the algebra L^V is the assignment of an element to the variable x . Some examples of substitutional instances: $e[x/b] = x$ for every b if e is closed; $x[x/b]$ is b ; $R_P(p, x)[x/t] = R_P(p, x[x/t]) = R_P(p, t)$.

What is essential for variables, is evidently that they can be substituted for by constants. This means that for any substitution S there must be an operator which assigns to an element its S -instance. An operator which substitutes its arguments for some given variables will be called S -closure.

DEFINITION 6.18. Let S be a stratification of A , A_x a variable extension of A , O a operation of A_x of the type $[C_1 x \dots x_{C_n} \rightarrow C_{n+1}]$ such that $S(O) = C_1$, and F a function of the type $[C_{n+1} \rightarrow C_1]$ over A_x . If for every $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n$ of the respective types $C_1, \dots, C_{i-1}, C_{i+1}, \dots, C_n$ and for every e of the type C_{n+1} it holds that $O(a_1, \dots, a_{i-1}, F(e), a_{i+1}, \dots, a_n) =$

$e[x_1/a_1, \dots, x_{i-1}/a_{i-1}, x_{i+1}/a_{i+1}, \dots, x_n/a_n]$, then F will be called S -closure for $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ and it will be sometimes denoted by $O_{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n}$. An algebra which originates from A_x by addition of S -closures will be called S -closure variable extension of A .

As we have only the variable x in L^V , we can have only closures for x ; an S_1 -closure for x cannot be evidently of a type other than $[S \rightarrow P]$. An F of this type is an S_1 -closure for x iff $R_P(F(s), t) = s[x/t]$. Let us denote the algebra L^V extended with one such function as L^{Vc} , and let us denote the restriction of L^{Vc} to the carrier of L as $L^{(Vc)}$.

What we now want to show is that the variable extension of an algebra partialized to the original algebra is in fact an abstractive extension of the algebra, in the case of our example that $L^{(Vc)}$ is equivalent with L^A . To prove this we first prove some auxiliary lemmas.

DEFINITION 6.19. Let A^* be an S -closure variable extension of A . A function F over A^* will be called *closed* if it has closed values for closed arguments, i.e. if $F(a_1, \dots, a_n)$ is from the carrier of A whenever all $a_1, \dots, a_n$ are. If F is closed, then by $F^{(A)}$ we shall denote the partialization of F to the carrier of A .

LEMMA 6.20.

Let F be a closed function over A and let $F(a_1, \dots, a_n) = O(F_1(a_1, \dots, a_n), \dots, F_k(a_1, \dots, a_n))$ for some functions $F_1, \dots, F_k$ over A for every $a_1, \dots, a_n$ from the domain of F . Then $F_1, \dots, F_k$ are all closed.

PROOF. Let us suppose that F_1 is not closed. Then $F_1(a_1^*, \dots, a_n^*)$ is open for some closed $a_1^*, \dots, a_n^*$. However, as the variable extension is free, the value of O must be open whenever one of its arguments is, and hence $O(F_1(a_1^*, \dots, a_n^*), \dots, F_k(a_1^*, \dots, a_n^*))$ is open; hence F is not closed.

[].

LEMMA 6.21. Let A^* be an S -closure variable extension of A . Let F be a polynomial function of the type $[C_1 x \dots x_{C_n} \rightarrow C]$ over A^* . Let F' be such a function of the type $[C_1 x \dots x_{C_n} x_{C_1}' x \dots x_{C_k}' \rightarrow C]$ over A^* that $F'(a_1, \dots, a_n, b_1, \dots, b_k) = F(a_1, \dots, a_n)[x_1/b_1, \dots, x_k/b_k]$ for all $\langle a_1, \dots, a_n, b_1, \dots, b_k \rangle \in C_1 x \dots x_{C_n} x_{C_1}' x \dots x_{C_k}'$. Then if F' is closed, then $F^{(A)}$ is an S -abstractive function over A .

PROOF. If F is a projection, then $F'(a_1, \dots, a_n, b_1, \dots, b_k) = F(a_1, \dots, a_n)[x_1/b_1, \dots, b_k/x_k] = a_i[x_1/b_1, \dots, x_k/b_k]$; hence $F'(A)(a_1, \dots, a_n, b_1, \dots, b_k) = a_i$ and $F'(A)$ is a projection over A . If F is a constant function, then $F'(a_1, \dots, a_n, b_1, \dots, b_k) = F(a_1, \dots, a_n)[x_1/b_1, \dots, x_k/b_k] = a[x_1/b_1, \dots, x_k/b_k]$. If F is closed, then either if $a \in A$, or a is x_i for some $i \in \{1, \dots, k\}$; in the former case $F'(A)(a_1, \dots, a_n, b_1, \dots, b_k) = a$ and $F'(A)$ is thus a constant function over A , in the latter case $F'(A)(a_1, \dots, a_n, b_1, \dots, b_k) = b_i$ and $F'(A)$ is a projection over A . Let now $F(a_1, \dots, a_k) = O(F_1(a_1, \dots, a_k), \dots, F_n(a_1, \dots, a_k))$ for some operator O and some polynomial functions $F_1, \dots, F_n$. So $F'(a_1, \dots, a_n, b_1, \dots, b_k) = O(F_1(a_1, \dots, a_n), \dots, F_n(a_1, \dots, a_n))[x_1/b_1, \dots, x_k/b_k]$. Let us first assume that O is not a closure. Then, by definition of substitutional instance, $F'(a_1, \dots, a_n, b_1, \dots, b_k) = O(F_1(a_1, \dots, a_n)[x_1/b_1, \dots, x_k/b_k], \dots, F_n(a_1, \dots, a_n)[x_1/b_1, \dots, x_k/b_k])$. As F' is closed, $F_1, \dots, F_n$ are according to the previous lemma closed, and hence, according to the inductive assumption, if we denote $F_i(a_1, \dots, a_n)[b_1/x_1, \dots, b_k/x_k]$ by $F_i'(a_1, \dots, a_n, b_1, \dots, b_k)$, then $F_i'(A)$ are S-abstractive over A . Hence as $F'(A)(a_1, \dots, a_n, b_1, \dots, b_k) = O(F_1'(a_1, \dots, a_n, b_1, \dots, b_k), \dots, F_n'(a_1, \dots, a_n, b_1, \dots, b_k))$, $F'(A)$ is S-abstractive over A . Now let us suppose that O is a S-closure for $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$. Then $n=1$ and there is a operation O such that $O(b_1, \dots, b_{i-1}, F(a_1, \dots, a_k), b_{i+1}, \dots, b_m) = F_1(a_1, \dots, a_k)[x_1/b_1, \dots, x_{i-1}/b_{i-1}, x_{i+1}/b_{i+1}, \dots, x_m/b_m]$. This implies that the function $F'(a_1, \dots, a_n, b_1, \dots, b_k) = F_1(a_1, \dots, a_k)[x_1/c_1, \dots, x_{i-1}/c_{i-1}, x_{i+1}/c_{i+1}, \dots, x_m/c_m]$ is closed, and hence is, according to the inductive assumption, $F'(A)$ S-abstractive over A . But as $O(c_1, \dots, c_{i-1}, F(A)(a_1, \dots, a_k), c_{i+1}, \dots, c_m) = F'(A)(c_1, \dots, c_{i-1}, a_1, \dots, a_k, c_{i+1}, \dots, c_m)$ for every $c_1, \dots, c_{i-1}, a_1, \dots, a_k, c_{i+1}, \dots, c_m$, $F'(A)$ is S-abstractive over A .
[].

THEOREM 6.22. Let A^* be an S-closure variable extension of A . If F is a closed polynomial function over A^* , then $F'(A)$ is an S-abstractive function over A .

PROOF. The theorem is a special case of the previous lemma for $k=0$.
[].

That every polynomial over $L^{(vc)}$ is a polynomial over L^A follows from the following equations:

$$R_{\lambda x}(s) = R_s(s) \text{ for every } s \in S$$

$$R_{\lambda x}(R_P(p, x)) = p \text{ for every } p \in P$$

$R_{\lambda x}(R_{01}(o, s)) = R_{01} \cdot (o, R_{\lambda x}(s))$ for every $o \in O1, s \in S$
 $R_{\lambda x}(R_{02}(o, s1, s2)) = R_{02} \cdot (o, R_{\lambda x}(s1), R_{\lambda x}(s2))$ for every
 $o \in O1, s1, s2 \in S$.

However, also the converse is true, namely that every polynomial over L^A is a polynomial over $L^{(vc)}$. This follows from the following:

$R_s(s) = R_{\lambda x}(s)$ for every $s \in S$
 $R_{01} \cdot (o, p) = R_{\lambda x}(R_{01}(o, R_P(p, x)))$ for every $o \in O1, p \in P$
 $R_{02} \cdot (o, p1, p2) = R_{\lambda x}(R_{02}(o, R_P(p1, x), R_P(p2, x)))$ for
 every $o \in O1, p1, p2 \in P$.

7 Discussion

That the predicate calculus of the first order can be reformulated without the use of variables is nothing new; it has been stated as early as in 1960 by Quine⁶, and it has been later further discussed by Quine himself as well as by other authors⁷. That the predicate calculus of any order can be reformulated without the use of variables follows from the fact of the equivalence of the theory of types and combinatory logic⁸; various ways of elimination of variables in intensional logic have been discussed in the recent book of Cresswell⁹. What has been proved here is, however, even a bit more general: that any "reasonably interpretable" language with variables can be reformulated as a language without variables.

The other thesis we have proven aims at a characterization of rules that are added via addition of variables. These rules are shown to be those which can be added, roughly speaking, without disturbing semantics. This means that addition of these rules does not mean a real change of an algebra; the addition can be considered as a mere making explicit of something that is implicit. Let us explain this with the help of one more example.

Let us imagine someone asks us to specify the algebra of natural numbers. We could answer that it is the algebra N with the carrier constituted by the numbers and with two binary operations, addition and multiplication. But what if he objects that we can add or multiply not only two numbers, but any number of numbers, and that the right algebra is N' with n -ary addition and multiplication for every natural n ? We would probably answer that there is no contradiction between his proposal and our one, that N and N' are only two different forms of the algebra.

This means that there is a viewpoint from which we identify algebras with certain of their extensions; what we put forward is that the basis for such an identification is the sameness of semantic character, i.e. the coincidence of interpretations. If an algebra is identified with its extension, then what is explicit in the extension must be considered as at least implicit in the original algebra; if N' is the same as N , then N must be considered as implicitly containing all the additions and multiplications of the arities greater than 2. To extend N to N' thus means to realize some kind of "dispositions". And in this sense is also an abstractive extension and hence addition of variables a realization of a disposition, or making explicit what is implicit.

The question which remains to be answered is why are the abstractive rules not added directly, why the roundabout of variables is used. The answer is quite simple: to add variables is much more economical. This is not completely clear in our oversimplified example where we have only three abstractive rules, but if we take the whole predicate calculus, then the difference is unmistakable.

8 Conclusion

The relation between natural language and its logical formalization is not a completely transparent matter: the correspondence between elements of the formalization and those of the formalized language is not one-to-one. This suggests to consider formalization as revealing something which is normally "hidden" or "latent". But if this is so, then it is not reasonable to consider variables as standing for "latent expressions", but rather as codifying "latent rules".

Notes

0. Work on this paper has been kindly supported by the Alexander von Humboldt-Stiftung. The author is grateful to P.Sgall, P.Stekeler-Weithofer and B.Partee for stimulating discussions.

1. That logical formulas are understood as "standing for" sentences or expressions of natural language is essential not only when we use logic as a tool of semantic analysis: just this is what makes logic irreducible to a branch of algebra.

2. An inadequate understanding of this discrepancy may impede the cooperation between language-oriented logicians and empirical linguists in the sense of Sgall et al.(1986, 5.1).

3. Janssen calls (i) *interpretation* and (ii) *translation*.

4. The term is substantiated: sameness of homomorphisms clearly coincides with sameness of congruences.

5. Each of the grammatical rules of the language of a propositional calculus combines an n-ary operator with n propositional formulas into a propositional formula; the denotation of the result of such a rule is then always considerable as the application of the denotation of the operator to the denotations of the formulas.

6. Quine(1960).

7. Quine(1971; 1972).

8. Curry and Feys(1957).

9. Cresswell(1990).

REFERENCES

- Cresswell, M.J.: 1990, *Entities and Indices*, Kluwer, Dordrecht
- Curry, H.B. and Feys, R.: 1957, *Combinatory Logic*, North Holland, Amsterdam
- Janssen, T.M.V.: 1983, *Foundations and Applications of Montague Grammar*, Ph.D.thesis, Mathematisch Centrum, Amsterdam
- Quine, W.V.O.: 1960, Variables explained away, *Proceedings of the American Philosophical Society* 104, 343-347; reprinted in Quine: *Selected Logic Papers*, Random House, New York, 1966
- Quine, W.V.O.: 1971, Algebraic Logic and Predicate Functors, in R.Rudner and I.Scheffler (eds.), *Logic and Art: Essays in Honor of Nelson Goodman*, Bobbs-Merill; reprinted in Quine: *The Ways of Paradox*, Random House, New York, 1976.
- Quine, W.V.O.: 1972: The Variable, in R.Parikh (ed.), *Logic Colloquium*, Springer; reprinted in Quine: *The Ways of Paradox*, Random House, New York, 1976.
- Sgall, P., Hajičová, E., Panevová, J.: 1986, *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*, Academia, Praha.

WH-IN-SITU: AN APPARENT PARADOX¹

Tanya Reinhart, Tel Aviv University

Two approaches have been proposed, in the syntactic frameworks, regarding the assignment of wide scope to the wh-in-situ in multiple questions. (I will be concerned here only with the standard cases where the wh-in-situ is assigned wide scope, as in (1).) The one, most commonly assumed, is that they undergo movement at LF, to some clause-initial position, where their scope is correctly captured, as illustrated for the Surface-Structure (1) in (2a). The other, originating in Baker (1977), is that each question-sentence contains an abstract Q-morpheme, and wh-in-situ are bound directly to Q. Recently, this view has regained popularity with the work of Pesetsky (1987) and others, who argue that at least in certain cases, wh-in-situ are bound in situ by Q. This line makes use of the notion of unselective quantification developed in Heim (1982), following Lewis (1975): the Q operator unselectively binds all the variables in the wh-NPs which have not moved. The LF derived this way for (1) is (2b).²

- 1) Which lady₂ [e₂ read which book₁]?
 - 2 a) [Which book₁ [which lady₂ [e₂ read e₁]]]
 - b) Q<_{1,2}>[which lady₂ [e₂ read which book₁]]
- 3 a) {P | (E x) (E y) (lady (y) & book (x) & P = (y read x) & true(P))}
- b) {P | (E <x,y> (lady (y) & P = (y read x & book (x)) & true(P))}

Although not much attention is given in the syntactic LF framework to the interpretations of these LFs, it seems unproblematic to translate them into standard representations. E.g., assuming Karttunen's (1977) semantics for questions, their respective translations are given in (3).³ On this view, wh-NPs are, essentially, existential NPs; the question denotes the set of propositions which are true answers to it. Thus, (3a), which is the translation of (2a), denotes the set of true propositions P such that there is a lady y and a book x, about which P asserts that y read x. (3b), the translation of (2b), differs from (3a) only in where the book-restriction occurs in the representation. For brevity, I will omit the truth - clause (true (P)) in subsequent examples.

A problem which has received surprisingly little attention in the LF framework is that, in fact, given standard semantics, neither approach can capture correctly the (full range of the) interpretation of wh-in-situ.

¹I would like to thank Irene Heim, Hans Kamp, Mats Rooth and, especially, Remko Scha for discussion and comments. Research on this paper was supported by a grant from the Netherlands Organization for the Advancement of Research (NWO), through the Department of Theoretical Philosophy of the University of Amsterdam.

²Both of these proposed LFs are motivated not just by the problem of assigning scope and interpretation to wh-in-situ, but also by syntactic considerations restricting their occurrence, which I discuss in Reinhart (1992).

³I chose Karttunen's framework since it lends itself easily to the type of solution I propose for the problems below. I leave it open whether the same can be stated also in the framework of Groenendijk and Stokhof (1982) 483

1. An interpretation paradox.

1.1. Let us focus first on the case where the N-restriction of the wh stays in situ. Is it, in fact, true that the variable in this N-set can be unselectively bound in situ, as exemplified in (3b). Given standard semantics (assumed, e.g., in the LF framework), the answer appears to be no:

- 4
 - a) Who will be offended if we invite which philosopher?
 - b) $Q_{\langle 1,2 \rangle} [\text{who}_2 [e_2 \text{ will be offended if we invite which philosopher}_1]]$
 - c) $\{ \text{Pl } (E \langle x,y \rangle \ \& \ P = \wedge ((\text{we invite } y \ \& \ \text{philosopher } (y)) \rightarrow (x \text{ will be offended}))) \}$
 - d) Lucie will be offended if we invite **Donald Duck**
- 5
 - a) $[\text{which philosopher}_1 [\text{who}_2 [e_2 \text{ will be offended if we invite } e_1]]]$
 - b) $\{ \text{Pl } (E x) (E y) (y \text{ is a philosopher}) \ \& \ P = \wedge ((\text{we invite } y) \rightarrow (x \text{ will be offended}))) \}$
- 6
 - a) Which linguist read every book by which philosopher?
 - b) $\{ \text{Pl } (E \langle x,y \rangle \ \& \ P = \wedge (\text{Az } (z \text{ is a book by } y \ \& \ y \text{ is a philosopher}) \rightarrow (x \text{ reads } z))) \}$
 - c) All linguists read every book by **Nancy Reagan**.

If in (4a), we bind the index of which philosopher to Q, while leaving the N-restriction in-situ, as in the LF (4b), the final logical representation will be that in (4c). Now let us consider the set of possible answers to the question under this representation. It turns out that the value for y can be anything in the world, since its restriction occurs in the antecedent clause of an implication. Suppose, e.g. we chose Donald Duck as a value for y in (4b). Since he is not a philosopher, the antecedent clause is false, and the implication is true for this value. So, if (4c) was the correct representation for (4a), (4d) should have been an appropriate answer.⁴ Similarly, if we allow unselective Q-quantification in (6a), we obtain the representation (6b), under which it turns out that a necessarily true answer is e.g. (6c), since it is true for every linguist x that if Nancy Reagan is a philosopher, then x read every book by her.

The representation yielding the correct set of answers in such cases is that in which the restriction is pulled out of the implication. E.g. for (4a), if the wh-in-situ is raised, yielding the LF (5a), the derived representation (5b) correctly allows the values for y to be all and only those individuals who are philosophers and for whom the implication is true.

⁴Technically speaking, it is not, in fact, fully clear that the representation (4c) should allow the relevant answer to be the proposition expressed in (4d). It allows Donald Duck as a value for y, but the proposition in the denotation set may have to be (i). If this is so, however, (4c) would also disallow (ii) as a possible answer, while equally allowing both (i) and (iii) as answers, which is sufficiently wrong.

- i) Lucie will be offended if Donald Duck is a philosopher and we invite him.
- ii) Lucie will be offended if we invite Kripke.
- iii) Lucie will be offended if Kripke is a philosopher and we invite him.

1.2. More generally, the mechanism of unselective binding, under its current formulations, is extremely powerful, and it is not clear that the full implications of allowing it in Universal Grammar have been sufficiently studied. It may seem, then, that our problem would be solved if we just give up unselective binding, and the idea of ever leaving wh-expressions in situ at LF. But is it really so that wh-in-situ must move at LF? The anaphora puzzle below, which was discovered by Engdahl (1986), suggests the opposite.

- 7 a) Who remembers which patient₁ had what (type of) fantasies about himself₁
 b) Answer: **Dr Razi** remembers which patient had **war-hero fantasies about himself**, **Dr Zira** remembers which patient had **Don Juan fantasies about himself**...
 c) *[what fantasies about himself₁]₃ [who₂ [e₂ remembers [which patient₁ e₁ had e₃]]]
 d) {Pl (E x) (E y) (phantasies about z (y) & P=[^](x remembers {P'! (E z) patient (z) & P'=[^]z had y})))}

There are two ways to answer question (7a), one is to assign a value just to the top who (e.g. with the answer: 'Dr Razi'). On this construal, the lower wh-in-situ is not being questioned, i.e. its scope is that of the embedded clause. The other way is illustrated in (7b). This answer provides the values of two wh-expressions: The top who and the wh-in-situ what fantasy about himself. To obtain this answer, the wh-in-situ must have wide (matrix) scope, but which patient has scope only over the embedded clause, as witnessed by the fact that no value is provided for it by the answer. If wide scope must be captured by LF movement, the wh-in-situ must be raised to form the LF in (7c). Now, this LF is ill-formed, since the anaphor himself is not bound. The problem here is not just syntactic. Once the variables and operators are translated, as in (7d), it becomes clear that the pronoun is not in the scope of the operator which is supposed to bind it (which z). So, the curious property which such structures seem to display is that the antecedent must have a narrower scope than the pronoun it binds.

Of course, the problem can be solved if the wh-in-situ is reconstructed back to its original SS position. But this would be precisely the interpretation obtained with no LF movement and unselective binding, and we have just seen that this too is impossible.

Since in the specific examples above, an interpretation in-situ will, nevertheless, yield the correct truth conditions for the sentence, it might be worth it to stretch our patience to the mental exercise below, which combines the two contexts we have examined:

- 8 a) Who remembers which lady₁ will be offended if we invite which of her₁ philosophical rivals.
 b) Answer: I remember which lady will be offended if we invite **her Deconstructionist Rival**.
- 9 a) {Pl (E x) (E y) (P = [^](x remembers {P'! (E z) (lady (z) & P'=[^]((we invite y & (z's philosophical rivals) (y)) ---> (z will be offended)))))})
 b) Answer: I remember which lady will be offended if we invite **Donald Duck**.
- 10 a) {Pl (E x) (E y) ((z's philosophical rivals)(y) & P = [^](x remembers {P'! (E z) (lady (z) & P'=[^]((we invite y) ---> (z will be offended)))))})

Although (8a) may not be the first question we would think about asking in any actual discourse, it is still not beyond our processing capacity, as witnessed by the fact that it can be straightforwardly answered, as, e.g. in (8b). But how could this answer be derived? So far we have precisely two options of analyzing the question with wide (matrix) scope to the *wh-in-situ* (which of her philosophical rivals): The one is to bind it unselectively in situ, as in (9a). Here all variables are appropriately bound, but we run into the Donald Duck problem we observed in (4): Since the restriction occurs inside an antecedent clause of an implication, there is no way to block (9b) as an appropriate answer to (8a). If, to avoid this problem, we choose to raise the *wh-in-situ*, obtaining (10a), we run into Engdahl's problem again. (10a) is nonsensical, since the variable *z* is outside the scope of its binding operator. So, neither approach to questions seems able, as stated, to explain why (8a) is, nevertheless, a well-formed question which can be answered.

In sum, then, if we change nothing in our standard semantics,⁵ we run into a contradiction: *Wh-in-situ* both cannot, and must, be interpreted in-situ.

2. A solution: Existential quantification over choice functions.

2.1. The interpretive problem we encountered was how to assign wide scope to *wh*-expressions which, otherwise, show properties of remaining in situ. Now, suppose, for the sake of analogy, that we had some reason to desire to assign the existential some book in (11a) wide scope, without pulling its restriction out.

- 11 a) Every lady read some book
b) $(\exists f) (A z) (\text{lady}(z) \rightarrow z \text{ read } f(\text{book}))$

One way to do that is to allow existential quantification over choice functions, i.e. functions applying to a set and yielding an individual member of the set. In (11) such a function applies to the set of books. The function variable can be bound by an existential operator arbitrarily far away. Note that $f(\text{book})$ here is an argument (of *read*), which corresponds to the fact that its NP stayed in argument position, and it denotes the value of the function *f* (i.e. a given book)

It is more interesting to check how the same procedure applies when the N-restriction occurs in the antecedent clause of an implication, since these are the contexts which pose problems to unselective binding.

- 12 a) Max will be offended if we invite some philosopher
b) $(\exists f) (\text{we invite } f(\text{philosopher}) \rightarrow \text{Max will be offended})$
c) $(\exists x) (\text{philosopher}(x) \ \& \ (\text{we invite } x \rightarrow \text{Max will be offended}))$
- 13 a) Lucie read every book by some philosopher.
b) $(\exists f) (A y) ((\text{book by } f(\text{philosopher}))(y) \rightarrow (\text{Lucie read } y))$

Although the N-restriction (philosopher) in (12) has stayed in situ, inside the *if* clause, the representation in (12b) correctly captures its truth conditions, and it is equivalent to the standard representation of wide scope in (12c). Similarly, (13b) correctly captures the (wide existential scope) reading of (13a), the

⁵ Engdahl offers, of course, a solution to her paradox, but I will explore here a line which, I believe, is somewhat simpler and more strictly compositional. Another motivation for this alternative solution is that it can solve some syntactic problems of scope assignment, which I discuss in Reinhart (1992).

implication being that for all y , if y is a book by the philosopher selected by the given function, then Lucie read y .

Note that we are considering here only the interpretation of an existential with a wide scope (which is not, independently, a particularly interesting problem). Hence the functions used here are the relatively trivial choice functions, which lack the complexity of the more familiar Skolem functions, employed to capture narrow scope of existentials, where the value they yield varies with the value of some other variable.⁶

2.2. While in the case of the standard existentials above, quantifying over choice-functions may be viewed as nothing more than a logical game, in the case of which-N, we have, as we saw, substantial reasons to allow them to stay in-situ sometimes. Crucially, in Karttunen's approach, which I have assumed here, wh-NPs are always translated into existential quantifiers. Hence, if desired, we can apply straightforwardly the same mechanism of quantifying over choice functions:

In (14), which book moved at SS, so it is no longer in argument position. Hence it cannot serve as an argument (of the form $f(\text{book})$), and it is interpreted as the restriction of the existential operator, just as before.

- 14 a) Which book₁ did Lucie read e_1 ?
 b) $\{P \mid (\exists x) (\text{book}(x) \text{ and } P = \wedge \text{Lucie read } x)\}$
- 15 Which book₁ did which lady read e_1 ?
- 16 a) LF: which lady₂ [which book₁ [e_1 read e_1]]
 b) $\{P \mid (\exists x) (\exists y) (\text{lady}(y) \text{ and } \text{book}(x) \text{ and } P = \wedge y \text{ read } x)\}$
- 17 $\{P \mid (\exists x) (\exists f) (\text{book}(x) \text{ and } P = \wedge f(\text{lady}) \text{ read } x)\}$

However, in (15), one wh-NP (which lady) stayed in situ at SS. So we have now two options. If it moves at LF, as in (16a), we would obtain the standard Karttunen representation in (16). If it stays in-situ, it can be interpreted as (17),

⁶ The more complex issue of capturing narrow wh scope in the functional reading of questions (e.g. Q: Who does every one love? A: His mother), may well lend itself to treatment with the equivalent of such Skolem functions, as in proposed in Chierchia (1991).

As far as I know, existential quantification over choice functions, to capture wide scope, was not used in linguistic semantics before. This is mainly because to capture just the fact that existentials can have freely wide scope, as in contexts above, constant Skolem functions are sufficient. The same will not be true when I apply this analysis to wh in situ. Since this use of choice functions is new, more attention must be given to their precise formal properties. I return directly to further specification of their domains, relevant for intensional contexts.

⁷ Of course, which book can be reinterpreted as an argument, if we create a Lambda predicate for that purpose, as in (ia), translated as in (ib). But since I am not aware of any reason to want to do that, this option need not be dwelled on.

- i a) Which book $(\lambda x (\text{Lucie read } x))$
 b) $\{P \mid (\exists f) (P = \wedge f(\text{book}) (\lambda x (\text{Lucie read } x)))\}$

which is the set of propositions each stating for some book x and for some function f that x read the book selected by f . This is precisely equivalent to the set defined in (16b).

Let us see now how this analysis handles the problems we observed in section 1.2. for binding wh-in-situ unselectively. In (4), repeated in (18), the problem was that if we leave the philosopher restriction in-situ, as in (18b) we would allow any non-philosopher to be a possible value for the question variable. However, under the present analysis, if the wh stays in situ, the representation will be (19) where the values permitted are, correctly, only those selected from the philosophers set, as we saw already in the parallel (12), with the existential some philosopher.

- 18 a) Who will be offended if we invite which philosopher?
 b) $\{Pl (E \langle x, y \rangle \ \& \ P = \wedge ((\text{we invite } y \ \& \ \text{philosopher } (y)) \rightarrow (x \text{ will be offended})))\}$
- 19 $\{Pl (E x) (E f) (P = \wedge ((\text{we invite } f(\text{philosopher})) \rightarrow (x \text{ will be offended})))\}$

Similarly, (6a), repeated in (20a), is analogous to (13). Using a choice function, as in (20b), rather than unselective binding of an individual variable, does not allow Nancy Reagan as one of the values of the answer, since the set selected from must be that of philosophers.

- 20 a) Which linguist read every book by which philosopher?
 b) $\{Pl (E x) (E f) (\text{linguist } (x) \ \& \ P = \wedge ((A y) (y \text{ is a book by } f(\text{philosopher}) \rightarrow x \text{ reads } y)))\}$

For both (18a) and (20a) a derivation with LF-movement of the wh-in-situ is equally possible. (This was illustrated for (18a) in (5).) But the anaphora problem of section 1.2. was that the N-restriction of the wh-in situ contained a variable bound by a wh antecedent with narrower scope, as in (7), repeated in (21a) (where what we are considering is the wide scope of what phantasies about himself). So the only way to get a well formed interpretation was to leave the wh in situ.

- 21 a) $Who_2 [e_2 \text{ remembers } [which \text{ patient}_1 [e_1 \text{ had what (type of) phantasies about himself}_1]]]$
 b) $\{Pl (E x) (E f) (P = \wedge (x \text{ remembers } \{P' (E z) \text{ patient } (z) \ \& \ P' = \wedge (z \text{ had } f(\text{phantasies about } z))\}))\}$

This can be easily done, with the mechanism proposed here, as in (21b). The variable z is appropriately bound by the existential corresponding to which patient, in the embedded clause. The N set left in situ is now (that denoted by) phantasies about z , and the function bound by the wide scope existential selects an individual from that set.

Finally, we may turn to the anaphora paradox in (8), repeated in (22).

- 22 a) $Who_2 [e_2 \text{ remembers which lady}_1 \text{ will be offended if we invite which of her}_1 \text{ philosophical rivals.}]$
 b) $\{Pl (E x) (E f) (P = \wedge (x \text{ remembers } \{P' (E z) (\text{lady } (z) \ \& \ P' = \wedge ((\text{we invite } f(z's \text{ philosophical rivals})) \rightarrow (z \text{ will be offended})))\}))\}$

Crucially, which of her philosophical rivals must be assigned wide scope here, which gives rise to the anaphora problem. As we saw, the anaphora requires leaving it in situ, but doing this with unselective binding gives the wrong

interpretation, enabling Donald Duck to be a possible value for this wh. In (22b) none of these problems arise. The interpretation is captured by quantifying over functions which select a value out of the set of z's philosophical rivals. The variable z in this set is appropriately within the scope of the binding existential operator, so the representation is both well formed, and gives the correct interpretation.

2.3. I have not been fully explicit yet on the formal characterization of the quantification I assume. It is intended to capture strictly the wide scope of existentials. To guarantee that it is equivalent to their standard wide-scope interpretation we must make sure that the given functions select always only from the extension of the N-set in the actual world (even when the N-restriction originates in an intensional context). This can be captured by defining the range of quantification for f, as in (23). This means that the precise representation of e.g. the wh-in-situ of (18a), repeated in (24a), should be (24b), rather than the simpler version I used so far. (f is defined to belong to the set in (23), and its argument is an intension.)

23) $G = \{f \mid \exists P (f(P) \in \forall P)\}$
P of type $\langle s, \langle e, t \rangle \rangle$

24 a) Who will be offended if we invite which philosopher?
b) $\{P \mid (\exists x) (\exists f \in G) (P = \wedge((\text{we invite } f(\wedge^{\text{philosopher}})) \rightarrow (x \text{ will be offended})))\}$

All instances of quantification over function variables above, should be read in the same way.

3. How is the interpretation derived.

What we saw so far, is that using existential quantification over choice functions solves the interpretation problem we encountered: It allows wh expressions to be interpreted in situ, without running into the problems of unselective binding of individual variables. But for this to be a feasible solution, rather than an exercise in logic, we need to know how it relates to the syntax of natural language. There are, in fact, two questions here: First, can this interpretation be compositionally derived for the given syntactic structures (in a way consistent with what we know about UG)? Next, can we make sure that whatever mechanism we use in the derivation will not allow also the wrong interpretations we wanted to exclude?

25 a)
$$\begin{array}{c} N'' \\ \swarrow \quad \searrow \\ \text{Det} \quad N \\ | \quad | \\ \text{some/wh} \quad \text{woman (i)} \end{array}$$

b) $(\exists x)$ $(\text{woman } (x))$

c) f $\{x \mid \text{woman } (x)\}$

Starting with the first question, let us look at a standard existential NP such as some woman, whose structure is given in (25a). Following Higginbotham (1983), N is generated with an index-argument, which must be bound ('discharged' in his terms, though for what we are concerned with here, nothing hinges on⁴⁸⁹ the specific formulation of the NP-structure). One way the variable can be bound is

if the determiner some is translated, itself, as an existential operator, as in (25b). However, one of the basic insights emphasized in the DRT framework is that indefinite determiners, or more generally weak determiners, do not necessarily correspond to an operator (unlike the strong determiners)⁸. An available alternative, then, is to bind the variable by forming a set as the translation of N, as in (25c). Now, the determiner is, in any case, a function, so an option (in the case of weak NPs) is to let it serve as a (choice-) function variable, applying to the given set. This is how f(woman), used in our formulae before, is derived.

The next procedure is the binding of the function variable. For this, existential closure applies, introducing an existential operator, along the lines proposed in Heim (1982) (except that this operator binds here a function variable).⁹

In the case of wh expressions, such as which woman, under the semantics we assumed for them all along, they are viewed just as standard existentials, hence at the local NP-level they can be analyzed just as in (25c). However, they differ from the other existentials in that their binding existential operator must be inserted in a predetermined position in the scope of the question-formation operator (which forms the set of propositions denoted by the question).

When applying existential closure, there are, in principle, two options. One is to introduce an existential operator for each free f-variable (which is what I have done in the formulae I used here), and the other is to let them all be bound by one operator, along the lines of unselective binding. In the cases under consideration here, the two ways are equivalent, and which one we chose depends on whether unselective binding is, independently, assumed to be an operation available in UG, a question which I will not get into here.¹⁰

So the interpretations I assumed here can be derived, and we may turn to the next question of whether we are not allowing too much. Specifically, I assumed, following the insights of DRT, that indefinite (weak) NPs are not necessarily closed. However, in that framework (e.g. Heim (1982)), this is taken to mean that an NP like (25a) can be treated as an open formula (woman(x)), whose variable is, then, existentially bound outside the NP. I.e. existential closure is allowed to bind directly the N-variable. If we allow that, and also allow wh to stay in situ, then we are back precisely where we started. In a sentence like (4) (Who will be offended if we invite which philosopher?), if we leave which

⁸ In slight variance from DRT I am assuming here that the determiner can also be directly interpreted as an existential operator, though it does not have to.

⁹ Technically, Heim uses the term 'existential closure' only for the introduction of an existential operator in the nuclear scope of another operator. I use it here in the broader sense of introducing this operator anywhere (arbitrarily far away). However, the restrictions on this operation are precisely the same as stated by Heim. Specifically, this cannot be done in the restrictive term of another operator.

¹⁰ We may note, though, that allowing just unselective existential closure is a much more restrictive assumption than that currently assumed, which, as far as I can see, would not run into the potential problems of the current liberal application of this procedure (though it would not, of course, handle everything attempted by the current application). In the specific case of wh-in-situ, this is an attractive option, since the location of the existential operator must be syntactically marked. We could assume that the existential operator is introduced by the one wh-NP which must move to COMP (visibly in English, invisibly at the LF of Chinese). It is, then, this same operator which bind all the function-variables marked syntactically with a wh.

philosopher in situ, and introduce an existential at the top-level to bind it, we obtain an interpretation identical to (4b), which allows Donald Duck to serve as a value for the answer. For the solution of the wh-paradox to work, it is essential that existential closure can bind only function variables.

The relevant restriction can be drawn from Higginbotham (1985). He argues, essentially, that the N-variable must be discharged inside the NP. This constraint could not, of course, follow from principles of logic, but it is, most likely, a universal constraint of natural language. If true, this entails that the options illustrated in (25) are the only ones available: either the variable is bound by the NP's determiner or by a set formation operator. This result does not effect the major insight in DRT, that weak NPs are not necessarily closed (since they may still contain a free function variable), but it entails a restriction on the type of existential closure allowed. E.g. variables bound by a discourse existential must be function variables. If this is so, this means that quantification over choice-functions is a crucial linguistic device and its precise formal properties should be studied in much greater depth than what I was able to do here.¹¹

References.

- Baker, C.L. (1970), 'Notes on the description of English questions
The role of an abstract question morpheme' Foundations of Language.
Chierchia, G. (1991), "Functional WH and weak crossover", ms. To appear in
the WCCFL proceedings.
Engdahl, E. (1986), Constituent Questions, Reidel, Dordrecht
Groenendijk, J. and M. Stokhof (1982), "Semantic analysis of wh-complements",
Linguistics and Philosophy Vol 5 no 2: 175-234
Higginbotham, J. (1985) 'On semantics' Linguistic Inquiry 16: 547-593
Heim, I. (1982), The semantics of definite and indefinite Noun Phrases,
PhD dissertation, University of Massachusetts, Amherst, Published in
1989 by Garland, New York.
Huang, J. (1982), Logical relations in Chinese and the theory of grammar,
PhD dissertation, MIT, Cambridge.
Karttunen, L. (1977), 'The syntax and semantics of questions', Linguistics
and Philosophy 1: 1-44.
Pesetsky, D. (1987), 'Wh-in-situ: Movement and unselective binding', in E.
Reuland and A. ter Meulen, eds. The representation of (in)definites,
MIT Press, Cambridge, Mass.
Reinhart, T (1992), 'Interpreting wh-in-situ', manuscript, Tel Aviv
University.

¹¹ Assuming this restriction would also have direct implications for the issues of unselective binding. E.g. The interpretation of (i) in (ib) is not allowed.

- i a) An old fashioned critic always hates a modern piece.
 b) (A <x,y>) ((old fashioned critic (x) & modern piece (y)) -->
 (x hates y))

ii) (A <f,g>) (f(old fashioned critic) hates g(modern piece))

Rather, the only interpretation of unselective binding allowed here would be (ii). At present, it is not even obvious how to determine whether the two representations may be equivalent, since much work is still needed on what¹⁹¹ the value of f or g is if the N set is empty.

Dealing with Ambiguities by Underspecification: A First Order Calculus for Unscoped Representations

Uwe Reyle
University of Stuttgart

Abstract

In this paper we develop a theory of language meaning that represents scope ambiguities by underspecified structures. The emphasis is on the development of a deductive component for these structures. The set of possible meanings of a sentence, or text is determined by a set of meta level constraints that restricts the class of semantic representations appropriately. Thus the way ambiguities are represented does not correspond to any of the usual concepts of formalizing ambiguities by means of disjunctions (of completely specified structures). A sound and complete proof theory is provided that relates these structures directly, without considering cases.

1 Introduction

Utterance interpretation is based on a relation between the linguistic form of the utterance and its meaning. Current approaches to natural language understanding assume that the linguistic form of an utterance is given by some syntactic analysis and that the relation between this form and its meaning is characterised by a translation process into some semantic representation structure. In almost all cases this relation is not functional. Whenever semantical ambiguities arise there is a set of meanings associated with a single form and to decide whether some other sentence logically follows from this form it has to be shown that it follows from each of these associated meanings. It is, therefore, the notion of *disjunction* on which such a theory of meaning relies.

In this paper we develop a theory of language meaning that represents scope ambiguities by underspecified structures. The translation into semantic form will thus be functional. The way ambiguities will be represented does not correspond to any of the usual concepts of formalizing ambiguities by means of disjunctions (of completely specified structures). A proof theory is provided that relates these structures directly, without considering cases.

Consider the following argument.

- (1) Many a problem about the environment preoccupies every politician.
Every politician who many a problem about the environment preoccupies proposes a solution.

Every politician proposes a solution.

The first sentence of (1) has two readings. The first reading is the one where the scope relation of its two NP's corresponds to their linear order. The other reading gives **every politician** wide scope over **many a problem about the environment**. In the second sentence, **many a problem about the environment** cannot have wide scope over **every politician**, because the scope of (proper) quantifiers is bound to their local domain – in this case to the relative clause. So, its ambiguity depends only on the way the indefinite **a solution** is interpreted: either as specific indefinite, or as dependent on **every politician**. Therefore the second sentence has two readings. Also for the same reason the conclusion of

(1) is two times ambiguous. Thus there are four possible readings of the premise set of (1), call them $P_1, \dots, P_4$, and two readings of the conclusion, G_1 and G_2 .

How are we going to relate these two readings of the conclusion to the set of readings of the premises? What are the inferential properties of ambiguous representations? And can they be characterized by one and only one notion of logical consequence?

We are aware of the fact that there is a whole range of logical consequence relations to be investigated. For instance, a proof of the argument (1) could be represented by (2) or (3).

$$(2) \quad P_1 \vee \dots \vee P_4 \vdash G_1 \vee G_2$$

$$(3) \quad P_1 \vee \dots \vee P_4 \vdash G_1 \wedge G_2.$$

If we adopt the consequence relations underlying (2) and (3), then we follow the policy to reckon with the worst as regards the premises: to accept the argument we have to prove that its conclusion follows from each of the readings of its premises. With respect to the estimation of the conclusion, (3) pursues the same policy, whereas (2) is less ambitious, in that it accepts the argument if there is at least one reading of the conclusion that can be proved.¹ In this paper, we will focus on the relation given in (2). One reason for this is that it is the most conservative relation in the sense that it satisfies the usual meta-mathematical properties, namely reflexivity, monotonicity and transitivity. ((3) does not satisfy reflexivity.)

The main advantage of this approach is to come closer to a meaning representation which has been asked for in different areas especially within the field of cognitive science. A recipient of an ambiguous sentence seems to use a kind of representation that is underspecified with respect to the disambiguations of its possible scope relationships.

2 Representing Scope Ambiguities

How are we going to represent the meaning of sentences without specifying the scope relations between their quantifiers? There are quite a few proposals in the literature. (See for example [Schubert/Pelletier], [Fenstad et. al.], [Hobbs/Shieber], [Nerbonne].) What all these proposals have in common is the idea of deriving unscoped representations which then may be transformed algorithmically into the set of disambiguated representations. If the algorithm is simple and effective, there is certainly a benefit to all these approaches. But effective as the algorithm might be, it has the disadvantage to be needed at all. Even though there is an unscoped representation for (1), it has to be translated into (2) or (3) for the purpose of deductive manipulation. As a consequence each of the representations $P_1, \dots, P_4$ of the different readings of the premise set overspecifies its meaning – and this overspecification has then to be compensated for by taking the disjunction of $P_1, \dots, P_4$.

But there is a further disadvantage of the mentioned representations. Consider the sentence

(4) Every professor who recommends a book is admired.

which we may represent à la [Schubert/Pelletier] by

(5) $admired(\forall x(professor(x) \wedge recommend(\exists ybook(y), x)))$

What the papers cited have in common is that they do not have a dynamic representation of the meaning of the indefinite a **book** which accounts for the fact that it is interpreted

as universally quantified if it has narrow scope with respect to **every professor** and that the other reading assigns it an existentially quantified meaning. This means that the disambiguation algorithm must deal with the problem of choosing the correct quantification type when creating the different meanings.²

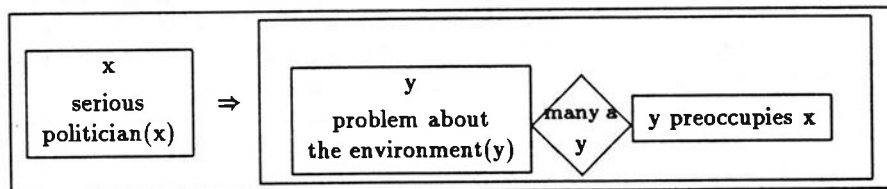
Thus in the framework of unscoped representations, sentences such as (4) cause the same problem as donkey sentences do. The problem is that the indefinite article is regarded as expressing existence. In DRT, this problem does not occur because the existential import of the indefinite in the cases where it has wide scope in (4) is not a consequence of the meaning of the NP as such, but rather of the way truth is characterized.

The base for our unscoped representations is the separation of information about the structure of a semantic form and of the content of the information bits the semantic form combines. Structural information will be expressed by a language with one predicate $\leq$ that relates individual constants l , called *labels*. The constants are names for DRS's. They are also used to position DRS-conditions at the right place in the hierarchy. This is done by writing $l:\gamma$ for an occurrence of a DRS-condition γ in a DRS named l .

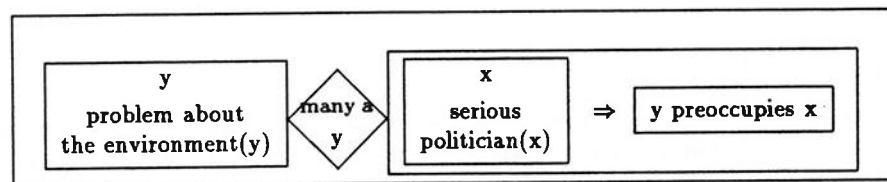
Given such a separation of structural information and purely linguistic content we are able to indicate that, for example, proper names always end up in the top-level DRS. Assume that the label of the top-level DRS is l_T . Then we can specify the target position of any proper name π in the lexicon by writing $l_T:\pi$. The scope potential of indefinite descriptions – like **a book** in (4) – is also dealt with in the lexicon: suppose that the meaning component of the indefinite is given by (a set of conditions of the form) $l':\gamma$, then we can express the fact that the indefinite may take arbitrarily wide scope by adding a $l \leq l'$, where l represents the minimal position l' can occupy.³

The construction of a meaning representation for a given sentence will then consist in relating names that show up in conditions associated with the phrases to be combined. Take the first sentence of (1). Its two readings are represented by the DRS's (6) and (7).⁴

(6)



(7)



To get an ambiguous representation of (1), we must to some extent encode the ways in which the meaning components contributed by the verb and NP's are combined.

- (8)
1. $l_1: \Rightarrow(l_{11}, l_{12}), \quad l_{11}: x, \quad l_{11}: \text{serious politician}(x)$
 2. $l_4: \text{many a } y \ (l_{41}, l_{42}), \quad l_{41}: y, \quad l_{41}: \text{problem about the environment}(y)$
 3. $l_2: y \text{ preoccupies } x$

What are the minimal restrictions to the possible ways of combination? The least we know is that the possible combinations are restricted by the semantic requirement that all discourse referents are properly bound. This is true iff the meaning of the verb is subordinate to each of the meanings of its arguments. More precisely, the verb must be subordinate to the scope of its argument meanings. This is captured by the following conditions.

(9) $l_2 \leq l_{12}, \quad l_2 \leq l_{42}$

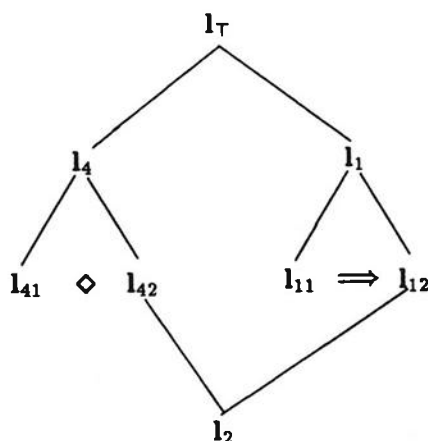
Furthermore, we know that all argument meanings must be subordinate to the DRS into which the meaning of the whole sentence (or clause) will be incorporated. The label of this DRS is given by the label of the local domain, $l_{max_{dom}}$, which for simple sentences will be equal to l_T . This gives us

(10) $l_1 \leq l_{max_{dom}} \quad l_4 \leq l_{max_{dom}}$

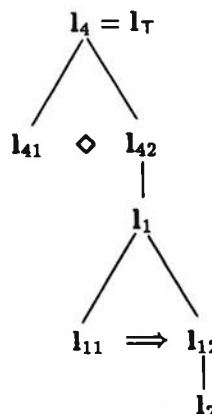
These additional conditions together with (8) specify all of what is common to (6) and (7). What they do not specify is the scope of the two NP's relative to each other. Nothing is said about l_i being strictly subordinate to l_j , or vice versa. And this is as we want it – except that there are syntactic restrictions that limit the range of scope ambiguities. To say that, for example, the subject of a sentence has to have wide scope over its object we enrich the structural information built up so far by the additional formula $l \leq l'$, where l is the label associated with the meaning of the subject and l' is the label associated with the object. This process of enrichment is characteristic for the construction of meaning representation: information from different sources (syntactic and semantic knowledge as well as knowledge about the world) may be added in a monotonic manner to narrow down the possible range of readings. In case of (8) we would add (11).

(11) $l_1 \leq l_{42}$

(11) removes all ambiguities in the structure determined by (8), (9) and (10) and completely specifies a structure which is homomorphically embeddable onto the DRS (7). It is convenient to represent the "structural" information given by the subordination relation and by conditions of the form $l: \Rightarrow(l_1, l_2), \quad l: \neg(l_1), \quad l: \text{quant } x \ (l_1, l_2)$ graphically. (12) represents the structural information given in (8), (9) and (10). The result of adding (11) is represented in (13).



(12)



(13)

Thus the set of possible meanings of a sentence (that arises from scope ambiguities) is determined by a set of meta level constraints that restricts the class of semantic representations appropriately. This set of constraints together with the meaning components of the NP's and the verb – as they are given in (8) above – will then be *the underspecified semantic representation* of the sentence.

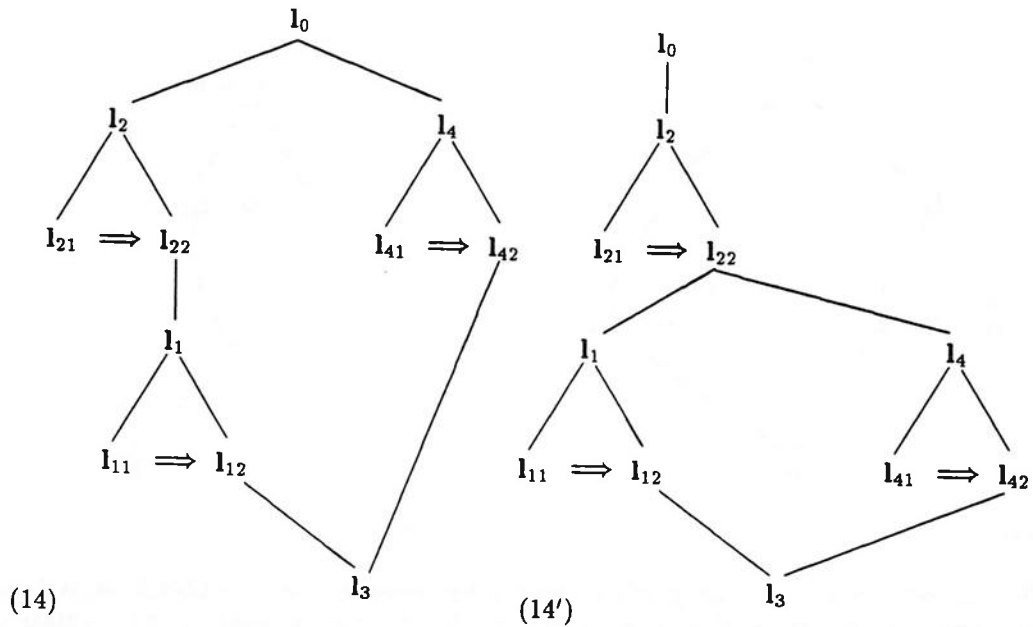
3 Meaning

We will assume that an ambiguous expression is true in a given situation in case one of its disambiguations is. The first sentence of (1), or, its underspecified DRS given in (12) (together with (8)), is true with respect to some model M iff one of (6) and (7) is true in M . The definition of truth will, therefore, rely on a disambiguation algorithm. A precise formulation of this algorithm is given in [Reyle 92]. For reasons of space, we give only an informal description here, which, however, will suffice to justify our rules. Correctness and completeness are also proved in [Reyle 92]. Note that the deduction rules themselves do *not* rely on any disambiguation.

We have already mentioned that in order to translate the structure given in (12) (together with (8)), into (7), it is sufficient to augment the structural information in (12) with $l_1 \leq l_{12}$. Analogously we get a structure corresponding to the DRS in (6) if we add $l_4 \leq l_{12}$.

In more complex examples, this operation has to be applied recursively. The structure in (14), e.g., allows for four readings, which we can characterize by the following orderings of its scope bearing elements: $\langle l_4, l_2, l_1 \rangle$, $\langle l_2, l_4, l_1 \rangle$, and $\langle l_2, l_1, l_4 \rangle$.

All these readings will be produced by a disambiguation procedure that starts at the top node and decides if l_4 shall have scope over l_2 or l_2 over l_4 . The former choice gives us the reading $\langle l_4, l_2, l_1 \rangle$. The latter choice yields a structure that is underspecified with respect to the relative scope of l_1 and l_4 . Applying the disambiguation procedure to l_{22} in (14') gives the two other readings.



Suppose we had – as in (1) – more than one ambiguous sentence. Then we must guarantee that the quantifiers of sentence S_i cannot get scope over some quantifier of a sentence S_j , where $i \neq j$. To exclude readings like this we introduce a new type of condition of the form $\sigma(l_1, \dots, l_n)$. The disambiguation procedure is restricted by these conditions such that a given node l can get scope over l' only if both are arguments of the same σ -condition. Thus we add $\sigma(l_1, l_4)$ to (12) (or (8)) and a corresponding condition for the meaning of second sentence to be incorporated.

As long as we are dealing with first order fragments, we do not allow structures containing conditions $l:\Rightarrow(l_1, l_2)$ and $l:\Rightarrow(l_3, l_4)$ together with some label l' such that $l' \leq l_2$ and $l' \leq l_4$. (Such structures would correspond to languages that allow partially ordered quantifier prefixes.)

To formulate further restrictions to the wellformedness of underspecified structures (which we will henceforth call UDRS's) let us introduce some terminology. Suppose that in some UDRS $\mathcal{K}$ l_1 and l_2 are daughters of l . Then l_1 and l_2 are called *parts* of l , if $\mathcal{K}$ contains a condition of the form $l:\Rightarrow(l_1, l_2)$. The same for $\neg$. The subset of labelled conditions of the form $l:\phi$ is called a *node* of the UDRS, denoted by D_l . *scope* is a function from the set of labels L to L defined as follows.

- (i) $scope(l) = l_1$, if $D_l = \{l:\neg(l_1)\}$
- (ii) $scope(l) = l_2$, if $D_l = \{l:\Rightarrow(l_1, l_2)\}$
- (iii) otherwise $scope(l) = l$

We say that a node D_l is *scope bearing* if it has one of the forms (i) to (ii), i.e. if $scope(l) \neq l$.⁵ If D_l is a scope bearing node, we call l a *scope bearing label*. For convenience we define $res(l)$ to be the set of daughters of l that is disjoint from $scope(l)$, if $scope(l) \neq l$; otherwise $res(l)$ is empty.

Suppose l_1 and l_2 are daughters of l but not parts of l . Then l_1 may take scope over l_2 , or vice versa. We can achieve that l_1 actually does have scope over l_2 if we add $l_2 \leq \text{scope}(l_1)$. But suppose that l_1 is scope bearing and that there is some l_3 such that $l_3 \leq l_2$ and $l_3 \in \text{res}(l_1)$. Suppose further that D_{l_1} contains free variables x and y which are declared in $\text{res}(l_1)$ and l_2 , respectively. By adding $l_2 \leq \text{scope}(l_1)$ we get a structure which does not correspond to a DRS in any natural way. Therefore, if some scope bearing node l_1 is assigned scope over some sister l_2 we only allow l_2 to be subordinate to $\text{scope}(l_1)$, if there are no l', l'' such that $l' \leq l_2$, $l' \leq l''$ and $l'' \in \text{res}(l_1)$.

Definition An *underspecified DRS* (UDRS) is a structure $\mathcal{K} = \langle \mathcal{L}^{\mathcal{K}}, D^{\mathcal{K}}, \text{scope}, \text{res} \rangle$, where

- (i) $\mathcal{L}^{\mathcal{K}}$ is an upper semi-lattice of labels $\langle \mathcal{L}^{\mathcal{K}}, \leq, 1_{\top} \rangle$ with one-element $1_{\top}$;
- (ii) $D^{\mathcal{K}}$ is a set of labelled conditions of the following types
 $l:x, l:P(x_1, \dots, x_n), l:\neg l_1, l:\Rightarrow(l_1, l_2), l:\sigma(l_1, \dots, l_n)$.
Here $l, l_1, \dots, l_n$ are labels, $x_1, \dots, x_n$ discourse referents, and P is an n -place predicate;
- (iii) scope and res are determined as above.

$\mathcal{L}^{\mathcal{K}}$ and $D^{\mathcal{K}}$ are subject to the following restrictions:

- (i) if $l:\neg l' \in D^{\mathcal{K}}$, then $l \neq l'$, $l' \leq l \in \mathcal{L}^{\mathcal{K}}$ and l' is a daughter of l in $\mathcal{L}^{\mathcal{K}}$;
- (ii) if $l:\Rightarrow(l_1, l_2) \in D^{\mathcal{K}}$, then $l_1 \neq l_2$ and l_1 and l_2 are daughters of l in $\mathcal{L}^{\mathcal{K}}$;
- (iii) if $l:\sigma(l_1, \dots, l_n) \in D^{\mathcal{K}}$, then $l_i \neq l_j$ for all $i, j = 1, \dots, n$ with $i \neq j$, and l_1 to l_n are daughters of l in $\mathcal{L}^{\mathcal{K}}$;
- (vi) Clause identification:
Suppose $l:\sigma(l_1, \dots, l_n) \in D^{\mathcal{K}}$, $k:\sigma(k_1, \dots, k_m) \in D^{\mathcal{K}}$ and $k_i \neq l_j$ for some i, j . Then
 - $\{l_1, \dots, l_n\} \cap \{k_1, \dots, k_m\} = \{\}$, and
 - l has a daughter l' which is not part of l iff l' occurs as argument of some σ -condition
- (v) First order property:
suppose $l:\phi$ and $l:\phi'$ are negative or implicative conditions such that for some daughter l_i of $l:\phi$ and l'_j of $l:\phi'$ there is an infimum k ; then $l:\phi$ and $l:\phi'$ must be scope bearing;
- (vi) Goodness:
For no l there is an l' such that $l \leq \text{scope}(l')$ and $l \leq \text{res}(l')$.

There is an obvious translation from DRS's to UDRS's. We therefore will feel free to talk of DRS's as if they were represented in the language of UDRS's. The translation induces a number of syntactic notions connected with UDRS's. First, each label l of a UDRS has its set of *locally declared discourse referents*, $U_l^{\mathcal{K}} = \{x : l:x \in D^{\mathcal{K}}\}$, which we also call its *local universe*, and its set of (*DRS-*) *conditions*, $C_l^{\mathcal{K}} = \{\gamma : l:\gamma \in D^{\mathcal{K}}\}$. The set $D_l^{\mathcal{K}} = U_l^{\mathcal{K}} \cup C_l^{\mathcal{K}}$ is called *node*. Second, we generalize these notions to sets of discourse referents and DRS-conditions of some l which do not only include the universe and conditions local to l but also all universes and conditions local to any l' which is subordinate to l .

- $\underline{U}_l^{\mathcal{K}} = \bigcup_{l' \leq l} U_{l'}^{\mathcal{K}}$ is the set of *discourse referents subordinate to l* ,
- $\underline{C}_l^{\mathcal{K}} = \bigcup_{l' \leq l} C_{l'}^{\mathcal{K}}$ is the set of *conditions subordinate to l* ,

- $\underline{D}_1^K = \underline{U}_1^K \cup \underline{C}_1^K$ (Hence $D^K = \underline{D}_{1_\tau}^K = \underline{U}_{1_\tau}^K \cup \underline{C}_{1_\tau}^K$),
- $\underline{L}_1^K = \langle \underline{L}_1^K, \leq_{\underline{L}_1^K}, 1 \rangle$, where $\underline{L}_1^K = \{l' : l' \leq 1\}$,
- $\mathcal{K}_1 = \langle \underline{L}_1^K, \underline{D}_1^K \rangle$ is the *sub-UDRS of $\mathcal{K}$ dominated by 1*.⁶

4 Deduction Rules for the Fragment with $\Rightarrow$ and $\neg$

The general architecture of the proof system for UDRS's is borrowed from the DRS-Calculus presented in [Kamp/Reyle 91]. In both systems, the most basic type of proof consists of a series of applications of *inference rules*, which permit us to extend the representation of the premises, but which do not involve any sub-goals. These proofs are called *direct proofs*. They involve one *rule of proof*, according to which a goal is proved if its representation is embeddable into the (extended) premise set. We first develop an idea of how direct proofs are to be performed and then consider other rules of proof, which will allow us to prove goals by establishing intermediate goals.

4.1 Inference Rules

Rule of Direct Proof

Suppose $\mathcal{K}'$ is embeddable in $\mathcal{K}$. Then $\mathcal{K} \vdash \mathcal{K}'$ succeeds.

An almost trivial application of the rule of direct proof is provided by the proof of $(14') \vdash (14)$. Here (14) is embeddable into (14') because $(14) \subseteq (14')$. (Note that in any standard approach we have to consider two subproofs.)

We assume that an embedding f is one-to-one when f is restricted to the set of discourse referents occurring in proper sub-universes of an UDRS $\mathcal{K}$. Discourse referents occurring in the universe associated with 1_τ might be identified by f . We do not assume that the restriction of f to the set of labels is also one-to-one. But f must preserve $\neg$ and $\Rightarrow$.

Definition Let f^U be a function from $\underline{U}_1^K$ to $\underline{U}_{1'}^{K'}$ such that f^U is 1-1 on $\underline{U}_1^K \setminus \underline{U}_{1_\tau}^K$. Let f^L be a function from $\underline{L}_1$ to $\underline{L}_{1'}$, such that $f^L(\underline{L}_1) = \underline{L}_{1'}$, and

- if $l \Rightarrow (l_1, l_2)$ occurs in K' , then $f^L(l) \neq f^L(l_1) \neq f^L(l_2)$,
- if $l : \neg l_1$ occurs in K' , then $f^L(l) \neq f^L(l_1)$.

Let Σ be the set of conditions of the form $k : \sigma(k_1, \dots, k_n)$ and f be $f^U \cup f^L$.

- If $f(\underline{D}_1 \setminus \Sigma) \subseteq \underline{D}_{1'}$, then f is called an *embedding of $\mathcal{K}_1$ in $\mathcal{K}'_1$* .
- If f is 1-1 on $\underline{U}_1^K$, then $\mathcal{K}'_1$ is called the *alphabetic variant of $\mathcal{K}_1$ according to f* .

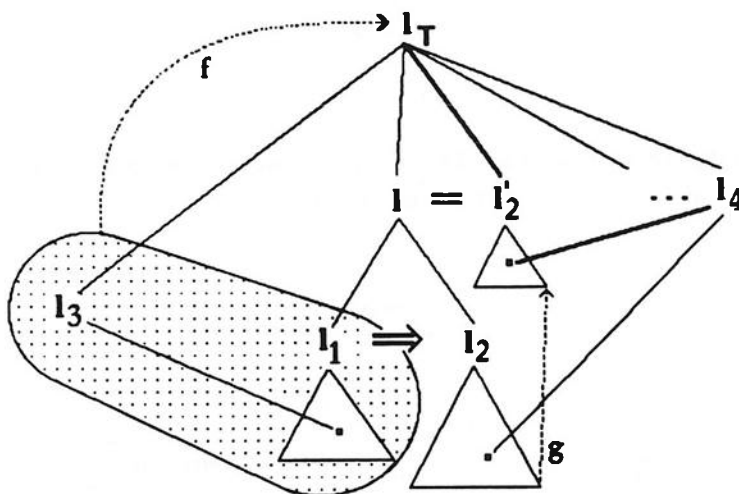
In case $l = 1_\tau$ and $l' = 1'_\tau$ we omit the subscript and talk of an embedding from $\mathcal{K}$ in $\mathcal{K}'$. The same holds for alphabetic variants.

In this paper, we consider the fragment without disjunction and identity. The first inference rule we need is non-empty universe (NeU). NeU allows one to add any finite collection of discourse referents to a DRS universe. It reflects the assumption that there is of necessity one thing, i.e. that we consider only models with non-empty universes.

Non-empty Universe:

May add to $\mathcal{K}_1$ any (finite) collection of conditions of the form $l:x$, where x is a new discourse referent.

The second inference rule – called DET1 – is derived from the familiar DRS-rule of DET(achment), which is a generalization of Modus Ponens. Graphically, DET1 can be represented as follows.



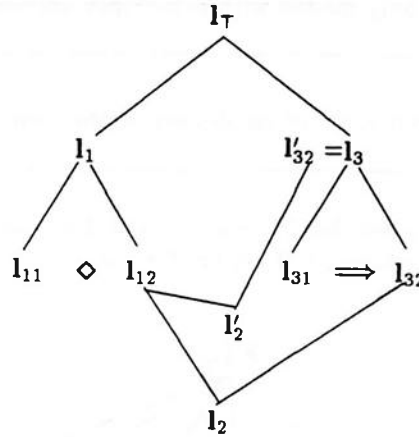
The operation is this: Suppose a UDRS $\mathcal{K}$ contains a condition of the form $l:\Rightarrow(l_1, l_2)$ as shown in the graph. Suppose further that there is an embedding f of $\mathcal{K}_{l_1} \cup \bigcup_{l_3} \mathcal{K}_{l_3}$ in $\mathcal{K}_{l_T}$, where $\bigcup_{l_3} \mathcal{K}_{l_3}$ is the union of all $\mathcal{K}_{l_3}$, where l_3 is such that there are l', l'' with $l' \leq l_3$, $l' \leq l''$ and $l'' \in res(l)$. Then we may add an alphabetic variant $\mathcal{K}_{l'_2} = g(f(\mathcal{K}_{l_2}))$ to $\mathcal{K}$, together with $l'_2 = l$ and $l'_2 \leq l_4$ for all labels l_4 to which l_2 is subordinate (except l).

We have to mention three points in order to justify the rule. First, why must f embed not only $\mathcal{K}_{l_1}$ but also $\bigcup_{l_3} \mathcal{K}_{l_3}$? Cases where such an l_3 occurs result from indefinite NPs in relative clauses, as in **Every politician who a problem preoccupies proposes a solution**. Here the meaning of the NP **a problem** can be introduced as being subordinate to the restrictor of the quantifier **every politician**, or it can be introduced at top level. The formulation of DET1 ignores the latter case. But this is sufficient because – according to our underlying consequence relation – the conclusion follows from the premises if it can be verified in each of their readings. In the case under consideration the reading where **a problem** is read non-specifically implies the other one. Thus the formulation of DET1 is sufficient in this respect.

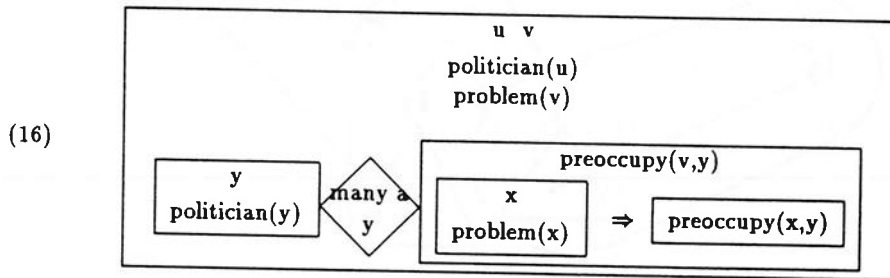
The second point of justification concerns the fact that DET1 is applied to a position that is subordinate to l_T . But as long as the (copy of the) antecedent is not lifted up to much, we are on the safe side: Consider the representation (12) of the first sentence of (1), and

suppose we have the additional information that there is a serious politician (given by $l_T:u$ and $l_T:\text{serious politician}(u)$). Then DET1 transforms (12) into (15),

(15)



with $l'_2:\text{preoccupy}(u,y)$ and empty $D_{l_{32}}$. This transition corresponds to the result (16) of applying a so-called extended DET to the DRS (7),



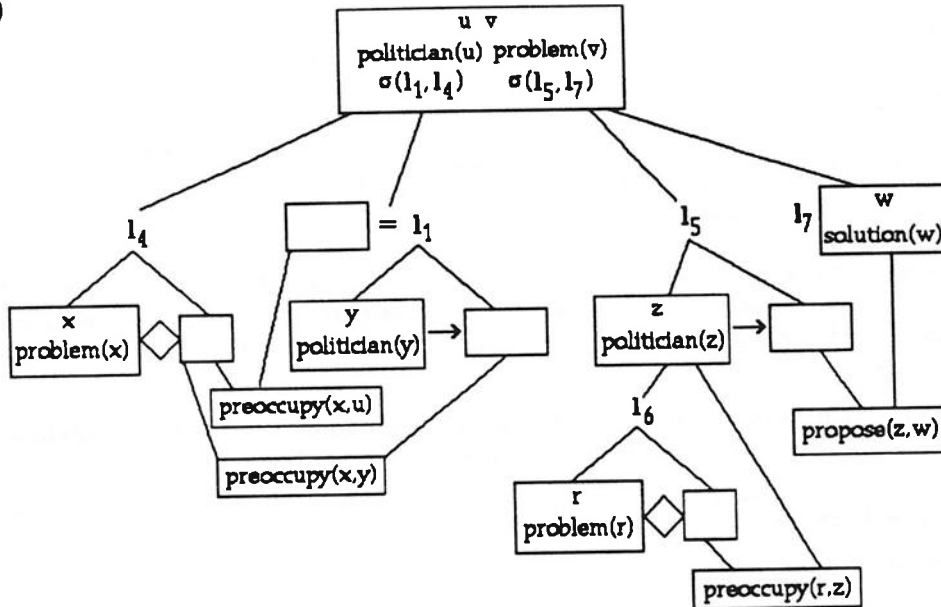
but is still underspecified with respect to the relative scope of **many a problem** and **every politician**. The equality sign between l_3 and l'_{32} means that alphabetical variants of the l_{32} -conditions are associated to the level denoted by l_3 .⁷ In (15), the set of l_{32} -conditions is empty. But in general this need not be so, and then it would be incorrect to associate the newly introduced l'_{32} -conditions with the top-node l_T . Because there they would get existential force which they don't have if (15) was true with respect to the reading where **many a** has wide scope.

For the last point in our justification of DET1 consider the following argument.

- (17) There is a politician. Many a problem preoccupies every politician.
Every politician who many a problem preoccupies proposes a solution.
 A politician proposes a solution.

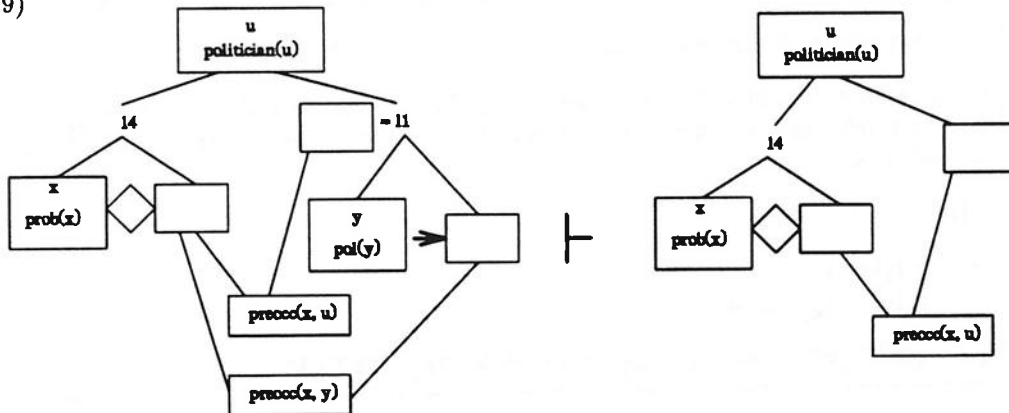
Suppose we have already applied DET1 once, with the result given in (18).

(18)

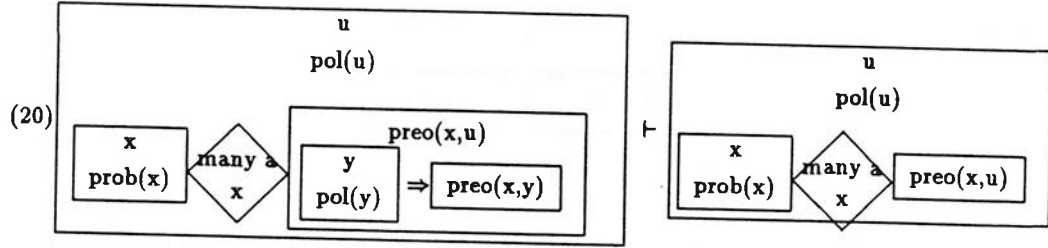


We now try to use the new information for another application of DET1. There is an embedding of the representation of **every politician who many a problem preoccupies** into the shaded area in (18), so it seems to be no problem to do this. To justify this application, however, we must make sure that the following proof is legal.

(19)



To see this, we consider its translation into the two DRS proofs which show that the right hand side of (19) is entailed by each of the two readings of its left side. The interesting proof we have to check is the one in which l_4 has wide scope over l_1 .



Inferences such as the one in (20) characterize quantifiers which are monotonically increasing in their second argument – which **many a** is. Thus (19) is legal only because l_4 dominates a monotone increasing quantifier. (Note that (17) does not hold if we replace **many a** by, e.g., **no**.)

We therefore have to restrict DET1 to *admissible* embeddings, i.e. to embeddings that respect the monotonicity properties of the quantifiers in their range.

DET(achment)1

Suppose

- (a) $l_\tau : \Rightarrow (l_1, l_2) \in D_{l_\tau}^\mathcal{K}$, or
- (b) $l_\tau : \sigma(k_1, \dots, k_n) \in D_{l_\tau}$ and $\{k_i : \Rightarrow (l_1, l_2)\} = D_{k_i}^\mathcal{K}$ for some i , $1 \leq i \leq n$;

and suppose that there is an admissible embedding f of $\mathcal{K}'_{l_1}$ in $\mathcal{K}$, such that $f(l_1) = l_\tau$, where $\mathcal{K}'_{l_1}$ results from $\mathcal{K}_{l_1}$ by

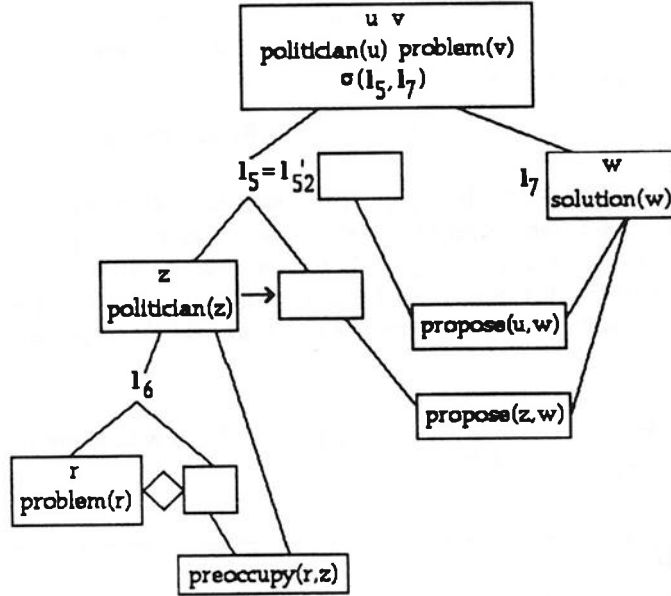
- (i) adding $k' \leq l_1$ for all k' for which there is some $k \leq l_1$ such that $k \leq k'$, and
- (ii) extending the argument list of $l_1 : \sigma$ with all these k' .

Let g be an extension of f to $\mathcal{K}_{l_2}$ such that $g \setminus f$ is 1-1 and such that g maps $\underline{U}_{l_2}$ to a set of discourse referents that are new to $\mathcal{K}$ and $\underline{L}_{l_2}$ to a set of labels that are new to $\mathcal{K}$. Then we add to $\mathcal{K}$

- (i) $g(\mathcal{K}_{l_2})$,
- (ii) $g(l_2) = l_\tau$, in case (a)
 $g(l_2) = l_{k_i}$, in case (b)
- (iii) $g(k''') \leq k''$ for each $k'' \notin \mathcal{L}_{l_2}$ and each $k''' \in \mathcal{L}_{l_2}$ if $k''' \leq k''$.

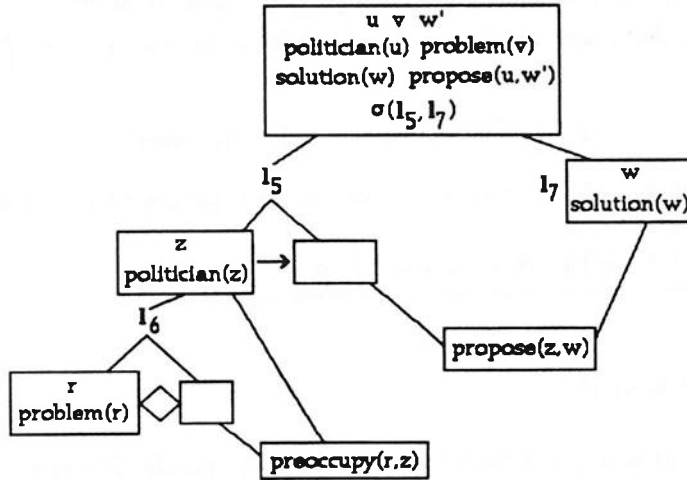
The second application of DET1 transforms the representation of the third sentence of (17) into (21).

(21)



Now, it seems to be straightforward to conclude the proof because $\langle \{w\}, \{sol(w)\} \rangle$ is monotone increasing. But there is a slight complication. As it stands, (21) does not distinguish between the solution proposed by u and the solutions proposed by any politician that is preoccupied by many a problem. To discriminate between the (possibly) different solutions involved, we need another inference rule, which we call DET2. It transforms (21) into (22).

(22)



Informally, an application of DET2 to some UDRS $\mathcal{K}$ consists of the following actions.

Suppose l_7 contains a condition $\sigma(l_1, \dots, l_n)$, and each argument l_i of σ is either a non-scope bearing node or it is scope bearing and set equal to some l'_i (by means of $l_i = l'_i$ as a result of DET1). Let us call the set of non-scope bearing nodes N and the set of nodes l'_i S' . ((21) satisfies these conditions and we have $N = \{l_7\}$ and $S' = \{l'_5\}$.)

1. Consider an alphabetic variant $\mathcal{K}^{NUS'}$ of the structure that results from the union of the substructures determined by NUS' by identifying all nodes in NUS' .

2. Eliminate all nodes of $\mathcal{K}$ that are either subordinate to some $l'_i \in S'$ or have an infimum with such a node. (The latter condition is necessary because there might be further nodes between $\text{propose}(u, w)$ and l_7 in (21).)
3. Add $\mathcal{K}^{NUS'}$ to this reduced structure by identifying their top nodes.

This does not yet transform (21) into (22). The result is a structure that differs from (22) in that there is no condition $l_\top: \text{propose}(u, w')$, but a condition $l': \text{propose}(u, w')$ which is immediately subordinate to $l_\top$, i.e. $l' \leq l_\top$. It is clear enough what to do in this case: we simply identify l' with $l_\top$. But in general the alphabetic variant constructed in the first step might be a genuinely ambiguous structure. We therefore add the appropriate σ -condition to $l_\top$.

4. Add $\sigma(k_1, \dots, k_n)$ to $l_\top$, where $\{k_1, \dots, k_n\}$ the set of labels that are immediately subordinate to the top node of $\mathcal{K}^{NUS'}$.

DET(achment)2

Suppose $\sigma(l_1, \dots, l_n) \in D_1^K$. Let N be the set of non-scope-bearing labels in σ , S the set of scope bearing ones and S' the set of labels l'_i such that $\mathcal{K}$ contains $l_i = l'_i$. Suppose further that for each $l_i \in S$ there is an $l'_i \in S'$. If $n \geq 2$, we define

$$\mathcal{K}_1^{NUS'} = \bigcup_{l_i \in NUS'} \mathcal{K}_{l_i}$$

Let g be a 1-1 function such that g maps $\bigcup_{l_i \in NUS'} \bigcup_{l_i} l_i$ to a set of discourse referents that are new to $\mathcal{K}$ and that maps all labels in NUS' to l . Then we may replace $\mathcal{K}_1^{NUS'} \cup \bigcup_{l_i \in N} D_1^K$ by

$$g(\mathcal{K}_1^{NUS'}) \cup \{\{\}, \{\sigma(k_1, \dots, k_m)\}\}, \text{ where}$$

$\{k_1, \dots, k_m\}$ is the set of labels that are immediately subordinate to the top node of $g(\mathcal{K}_1^{NUS'})$.

In case $n=1$, i.e. $\sigma(l_1) \in D_1^K$, we simply add $l_1 = l$.

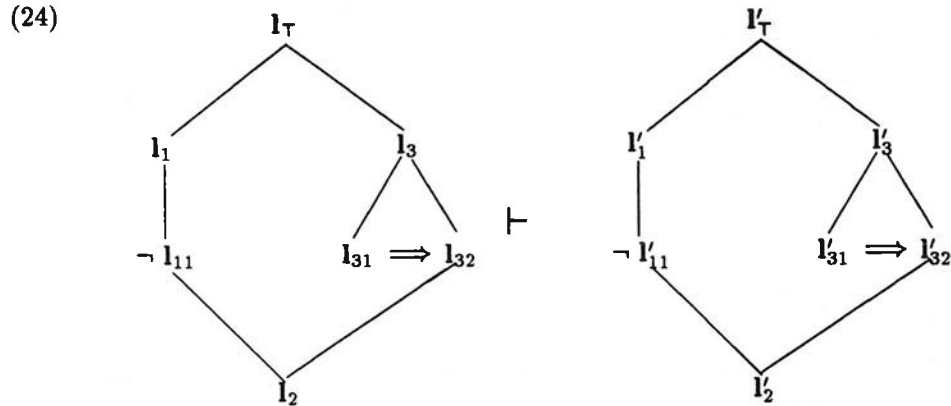
4.2 Rules of Proof

Rules of proof are used to prove goals by establishing sub-proofs. The system of Kamp and Reyle ([Kamp/Reyle 91]) contains two rules of proof. One is the rule of conditional proof (COND) which allows us to prove a DRS-condition of the form $K_1 \Rightarrow K_2$ by adding K_1 to the premises and deriving K_2 . The other rule is Reductio ad Absurdum (RAA) according to which a DRS K is shown if a contradiction can be derived from adding $\neg K$ to the premise set. We will see that – although we might adapt these two rules without any substantial changes to UDRS proofs – we need a further principle that allows us to detect inconsistencies of UDRS's. But before we come to this we will discuss formulations of COND and RAA, which seem to be more appropriate in the case of underspecified representations. We will call these variants *weak* COND and *weak* RAA. The difference between the *strong* (or *standard*) and weak version of COND, for instance, is that the strong version can only be applied if the

to be applied to implicative conditions that occur in subordinate positions. Consider the following argument.

$$(23) \quad \frac{\text{Every boy ate no doughnut.}}{\text{Every boy touched no doughnut.}}$$

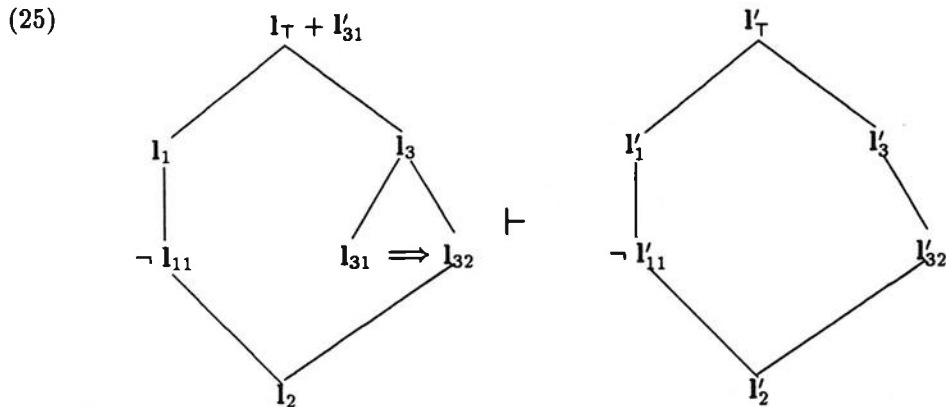
We assume that both sentences are ambiguous in the way (24) exhibits.⁸



The conditions associated with (24) are: l_{31} -conditions x and $\text{boy}(x)$, l'_{31} -conditions x' and $\text{boy}(x')$, l_{12} -conditions y and $\text{doughnut}(y)$, l'_{12} -conditions y' and $\text{doughnut}(y')$, l_2 -condition $\text{ate}(x,y)$, and l'_2 -condition $\text{touched}(x',y')$.

Let us also assume that eating implies touching, which amounts to adding to (24) the l_T -condition $l_T \Rightarrow (l_{T1}, l_{T1})$, with l_{T1} -conditions u , v and $\text{ate}(u,v)$ and l_{T2} -condition $\text{touched}(u,v)$. Then we should be able to prove (24).

The right side of (24) allows for an application of weak COND. (Standard COND cannot be applied because $l_3 \Rightarrow (l_{31}, l_{32})$ is subordinate to l'_T .) The results of its application is shown in (25), where $l_T + l'_{31}$ means that the l'_{31} -conditions of the conclusion are added to the set of l_T -conditions of the premises.



But then the antecedent l'_{31} is deprived of the possibility to have narrow scope with respect to $\neg l'_{11}$. This means that the decision to reduce the implicative condition before dealing with $\neg$ amounts to choosing to prove that reading of the right hand side of (24) which assigns

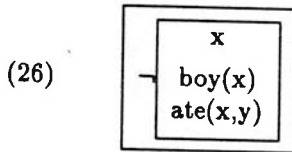
$\neg$ amounts to choosing to prove that reading of the right hand side of (24) which assigns $\neg$ narrow scope with respect to $\Rightarrow$. Therefore the application of weak COND to (24) can be paraphrased as follows: first, give the implicative condition wide scope and then apply standard COND.

weak COND:

Suppose $\mathcal{K}'$ is a goal containing an implicative label k , i.e. $D_k = \{k:\Rightarrow(l_1, l_2)\}$, such that either $k = l_\top$, or $D_{l_\top} = \{\sigma(k_1, \dots, k_n)\}$ and $k = k_i$ for some i , $1 \leq i \leq n$.

The CP-proof triggered by this goal is $\mathcal{K} \cup \mathcal{K}'' \vdash \mathcal{K}'''$, where $\mathcal{K}''$ results from $\mathcal{K}'_{l_1}$ by adding $k' \leq l_1$ for all k' for which there is some $k \leq l_1$ such that $k \leq k'$, and $\mathcal{K}'''$ results from $\mathcal{K}'$ by replacing $k_i:\Rightarrow(l_1, l_2)$ with $l_2=l_\top$.

But the decision to give the conditional wide scope over the negation turns out to be wrong: The conclusion of (25) corresponds to the DRS (26),



which is not entailed by that reading of the antecedent in which $\neg$ has wide scope over $\Rightarrow$.

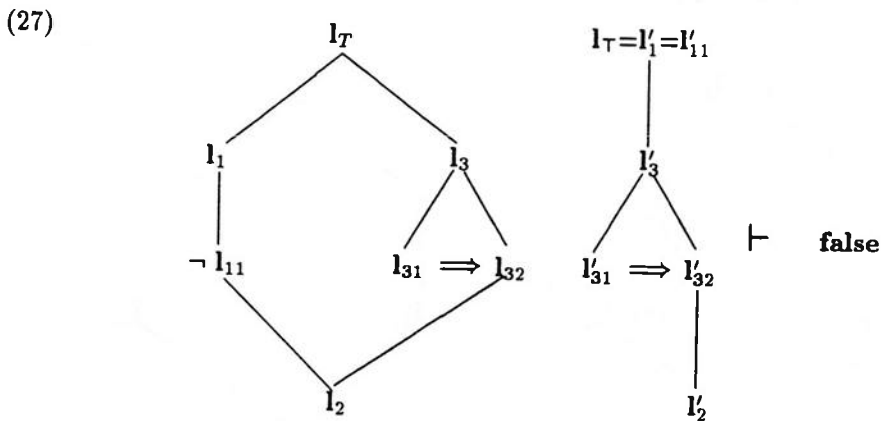
We formulate weak RAA in analogy to weak COND, i.e. apply standard RAA after having given wide scope to some subordinate label.

weak RAA:

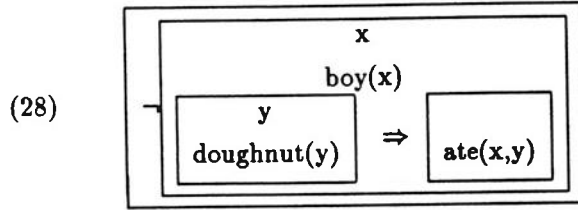
Suppose $\mathcal{K}'$ is a goal containing a negative label k , i.e. $D_k = \{k:\neg(l_1)\}$, such that either $k = l_\top$, or $D_{l_\top} = \{\sigma(k_1, \dots, k_n)\}$ and $k = k_i$ for some i , $1 \leq i \leq n$.

The RAA-proof triggered by this goal is $\mathcal{K} \cup \mathcal{K}'' \vdash \text{false}$, where $\mathcal{K}''$ results from $\mathcal{K}'$ by replacing $k_i:\neg(l_1)$ with $l_1=l_\top$.

This transforms (24) into (27).



But again the premise set of (27) is not inconsistent because we may choose the reading (28) for its first condition.

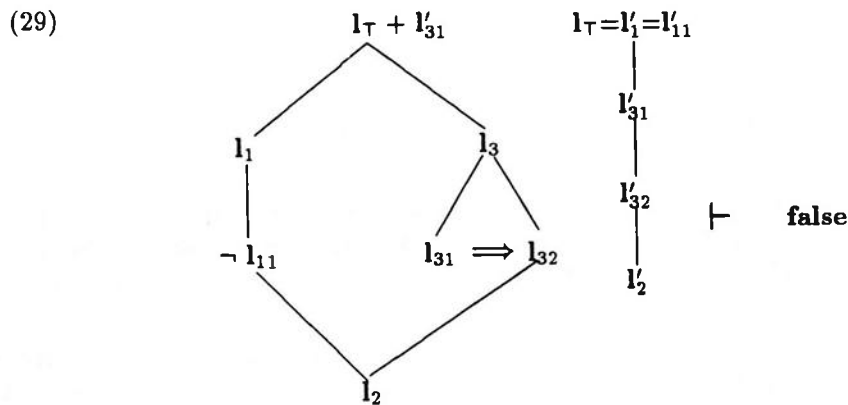


To compensate for the failure in both cases we need a further rule. It is called **RESTART** rule (introduced in [Gabbay 85]). **RESTART** allows to continue any (sub)computation by replacing the current goal with the original one.

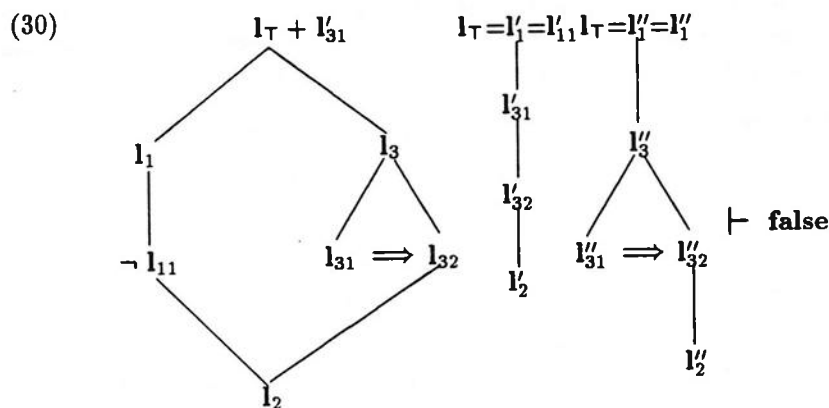
RESTART:

Suppose we have $\mathcal{K} \vdash \mathcal{K}'$. Then $\mathcal{K}'$ may be replaced by $\mathcal{K}''$, where $\mathcal{K}''$ is the original (or some previously introduced) goal.

To illustrate the rule consider once more (25). Suppose we continue this subproof by RAA, yielding (29),



and then apply **RESTART**, replacing **false** by the original goal, i.e. the one in (24), followed by another application of RAA. Then we get (30),



the premises of which are clearly inconsistent.

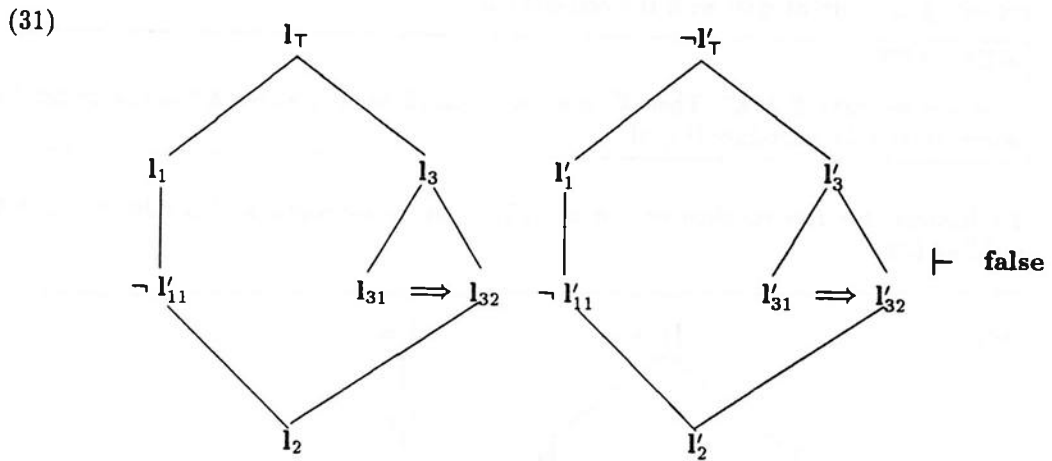
You might think that we can do it without RESTART if we combine the weak version of COND with the standard version of RAA.

RAA:

Suppose $\mathcal{K}'$ is a goal.

The RAA-proof triggered by this goal is $\mathcal{K} \cup \neg\mathcal{K}' \vdash \text{false}$, where $\neg\mathcal{K}'$ is the structure that results either from $\mathcal{K}'$ by adding l_τ to $L\mathcal{K}'$ with $D_{l_\tau} = \{ l_\tau : \neg l'_\tau \}$.

This would transform (24) to (31),



which can be shown to succeed. In [Reyle 92] we show that the standard version of RAA is in fact derivable from weak COND, weak RAA and RESTART; completeness is shown with respect to weak COND and standard RAA. (Again there is no space to compare the two rule sets here.)

Reconsider the structures (14) and (14') of the previous section. Recall that (14) represents the set of readings $\langle l_4, l_2, l_1 \rangle$, $\langle l_2, l_4, l_1 \rangle$, and $\langle l_2, l_1, l_4 \rangle$, whereas (14') stands for the disambiguations $\langle l_2, l_4, l_1 \rangle$, and $\langle l_2, l_1, l_4 \rangle$. Suppose now that some data contains (14) and the negation (of some alphabetic variant) of (14'), which means that this set of data is equivalent to the reading given by $\langle l_4, l_2, l_1 \rangle$. Let us sketch the proof that this reading follows from the data. It is by RAA: We first add the negation of the goal to the data. Then we have to show a contradiction. In this case the contradiction is manifest in the fact that there is no possible order of the quantifiers left. To establish the contradiction we have to do some calculations on the structures. This will be done as follows.

1. Take the set theoretical difference between the theories of the two structures, and
2. Add the negation of each formula in this set theoretical difference to the information about the structure.

A consistency check will then show that we have got an inconsistent set. There is no structure $\mathcal{L}$ satisfying all the formulas in the so extended structural description of the premises.

For the following definition, we introduce the convention that $\mathcal{K}_{l:\sigma(l_1, \dots, l_n)}$ denotes the sub-structure determined by the arguments of σ if there are any, and in case where $n=0$ it denotes $\mathcal{K}_l$.

Definition Suppose $D_{\Gamma}^{\mathcal{K}}$ contains conditions $l_{\Gamma}:\sigma(l_1, \dots, l_n)$ and $l_{\Gamma}:\neg l_i$, where $l \neq l_i$ for all $i=1, \dots, n$. Let f be a 1-1 embedding from $\mathcal{K}_1$ onto $\mathcal{K}_{l_{\Gamma}:\sigma(l_1, \dots, l_n)}$. Then the *structural difference* between $\mathcal{K}_{l_{\Gamma}:\sigma(l_1, \dots, l_n)}$ and $\mathcal{K}_1$ is $\underline{\mathcal{L}\mathcal{K}_1} \setminus \underline{\mathcal{L}\mathcal{K}_{l_{\Gamma}:\sigma(l_1, \dots, l_n)}}$.

- We abbreviate this by $DIFF(\mathcal{K}_1, \mathcal{K}_{l_{\Gamma}:\sigma(l_1, \dots, l_n)})$.
- $\neg DIFF(\mathcal{K}_1, \mathcal{K}'_1)$ is defined as the set $\{\neg \gamma \mid \gamma \in DIFF(\mathcal{K}_1, \mathcal{K}_{l_{\Gamma}:\sigma(l_1, \dots, l_n)})\}$, where $\neg \gamma$ is false if $DIFF(\mathcal{K}_1, \mathcal{K}_{l_{\Gamma}:\sigma(l_1, \dots, l_n)}) = \{\}$.

The notion of structural difference generalizes the traditional notion of inconsistency, according to which a UDRS would be *inconsistent* if it contained conditions of the form $l_{\Gamma}:\gamma$, $l_{\Gamma}:\neg l$, and $l:\gamma'$, where γ' is some alphabetic variant of γ . If we build the structural difference between $\mathcal{K}_1$ and $\mathcal{K}_1$, it will turn out to be empty, and the constant **false** will be added to the data, making the inconsistency overt.

DIFF

Suppose $D_{\Gamma}^{\mathcal{K}} l_{\Gamma}:\neg l$. Then we may add $\neg DIFF(\mathcal{K}_1, \mathcal{K}_{\sigma(l_1, \dots, l_n)})$ to L , if $l \neq l_i$ for all $i=1, \dots, n$.

In cases where the structural difference is given by a non-empty set, we need a consistency check in order to find out if the resulting structure is an inconsistent UDRS. To do this we need deduction rules which reason about the partial order of $\mathcal{L}$ and the conditions of the form $\sigma(l_1, \dots, l_n)$. We will call them *structural rules*. We will not state them explicitly. But it is easy to see how they might be implemented. For example we might interpret conditions of the form $\sigma(l_2, l_4)$ as abbreviations of the clause $l_4 \leq l_2 \vee l_2 \leq l_4$. We have to show that this clause together with $\neg l_2 \leq l_4$ and $\neg l_4 \leq l_2$ is inconsistent, which is most effectively performed by a resolution based method.

5 Conclusion

We presented a proof theory for structures that are underspecified with respect to scope ambiguities. For reasons of space we could only give an informal overview of how these structures are constructed and what truth conditions they have. In [Reyle 92] this gap is filled. There the construction of underspecified structures is discussed in detail and the formal apparatus necessary for the definition of truth is provided. On this basis correctness and completeness of the deduction system is shown.

6 Notes

¹In a similar way the interpretation of the premises could be weakened. It seems also plausible to make the choice of the consequence relation depend on some definition of coherency of discourse or dialogue, which may be triggered by syntactic parallelism or other means. This option would depend on the way in which the representation of the premises is transparent to the one who asks the question. We see that a theory of communication is needed to discuss all the possibilities properly.

²In [Fenstad et. al.], the problem is approached from a somewhat different perspective: in order to get the non-specific reading of a book they analyze the NP **Every professor who recommends a book** as a binary quantifier.

³For the purpose of this paper, I' can be identified with the position of the verb l_{verb} of which the indefinite is an argument. There are factors which force I' to be located at some place superordinate to l_{verb} . For more details, see [Reyle 92].

⁴For the treatment of general quantifiers in the framework of DRT, see [Kamp/Reyle 91].

⁵When we deal with plural a node of the form $D_1 = \{I:\phi(X)\}$, where $\phi(X)$ is some condition with a plural discourse referent X occurring free in it, will also be considered as a scope bearing node.

⁶Whenever there is no danger of confusion, we will omit the superscript K .

⁷This guarantees that each time information is lifted by some inference rule, it is identifiable as the result of this rule application.

⁸This assumption is at least problematic. But again there is no space to discuss the issue here.

References

- [Fenstad et. al.] Fenstad, J.E. / Halvorsen, P.-K. / van Benthem, J. (1987) *Situations, Language and Logic*, Reidel Publishing Company, 1987
- [Gabbay 85] Gabbay, D. (1985): "N-Prolog: An Extension of Prolog with Hypothetical Implications. Part II" in: *Journal of Logic Programming*, Vol. 4, 1985
- [Hobbs/Shieber] Hobbs, J.R. Shieber, S.M. (1987): "An Algorithm for Generating Quantifier Scopings", In: *Computational Linguistics* 13
- [Kamp/Reyle 91] Kamp, H. / Reyle, U. (1991): *From Discourse to Logic, Vol I*, to appear: Kluwer, Dordrecht (1990)
- [Kamp 85] Kamp, H. (1985): "Context, Thought and Communication", *Proceedings of the Aristotelian Society*, 85, pp 239-261
- [Kamp/Reyle 91] Kamp, H. / Reyle, U. (1991): "A Calculus for First Order Discourse Representation Structures", SFB-Report, Stuttgart, 1991
- [Nerbonne] Nerbonne, J. (1992), "A Feature Base Syntax/Semantics Interface", ms. Saarbrücken
- [Reyle 92] Reyle, U. (1992 b): "Dealing with Ambiguities by Underspecification: Construction, Representation and Deduction", appears as Dyana-Report
- [Schubert/Pelletier] Schubert, L.K. / Pelletier, F.J. (1982), "From English to Logic: Context-Free Computation of Conventional Logic Translations", in: *Journal of the Association for Computational Linguistics*

Belief revision and ordered theory presentations

Mark Ryan, Imperial College, London

Abstract

The standard theory of belief revision—known as the AGM theory after its authors, C. Alchourrón, P. Gärdenfors and D. Makinson [3]—suffers from at least two disadvantages. One is that it represents belief states as *infinite* objects, namely deductively closed sets of sentences. Another is that existing belief revision models make *too strong assumptions* about what information is available to guide revisions. A consequence of this is that repeated revisions are impossible.

In this paper, we show how to use *ordered theory presentations* to represent belief states¹. Ordered theory presentations are theory presentations equipped with a partial order. In this paper we are only concerned with a special case, in which the sets are finite and the order is total, or linear. We define a revision operator, which is shown to satisfy some, but not all, of the AGM postulates. No information other than that encoded in the OTP is needed to effect the revision; this makes repeated revision easy.

1 Introduction

The central question in belief revision is the following: given a *belief state* and a *sentence*, how should one obtain a new belief state in which the sentence is *true*, but which preserves as much of the old belief state as possible? In other words, one wants a function

$$* : \text{belief states} \times \text{sentences} \rightarrow \text{belief states}$$

such that

1. ϕ is true in $\Gamma * \phi$; that is, the revision has been *effective*; and
2. given this constraint, $\Gamma * \phi$ contains 'as much' of Γ as is consistent; that is, old beliefs persist through revisions if they can.

The case of interest, of course, is that in which $\neg\phi$ is true in Γ , so that the revision is more than just *refinement*, or the addition of compatible information. We also hope that

3. $\Gamma * \phi$ does not contain any extraneous information which was present in neither Γ nor ϕ .

In the above requirements, some things are easy to formulate and some are not. We assume that any satisfactory representation of belief states comes with a function $|\cdot|$ which takes a belief state and returns the set of sentences true in it. $|\Gamma|$ is called the *extension* of Γ . But formalising the 'as much' requirement and the requirement of no extraneous information (numbers 2 and 3) is not so easy, and is the subject of this paper.

Belief revision has obvious applications in artificial intelligence (eg. robotics), computer science (eg. deductive databases), the philosophy of science, social theory and so on. It also has applications beyond the idea of revising 'beliefs'. For example, in specification theory and in AI, there is

¹Ordered theory presentations have already been described in a previous paper [12]. In that paper, they were called 'structured theories', but since they are in fact a way of using an ordering on a set of sentences to present a theory, their name has been changed to *ordered theory presentations*, or *ordered presentations*, or even OTPs, for short. Familiarity with [12] is not required in order to understand this paper.

the well-known frame problem to do with the semantics of actions. Given (the representation of) a state of a system and the post-condition of an action performed in the state, what is the state which results from performing the action? The same requirements on the revision function apply here too: the post-condition should be true in the resulting state, which (given this constraint) should preserve as much of the original state as possible.

We begin the next section by describing the standard theory of belief revision, known as the AGM theory. The AGM theory suffers from several disadvantages. One is that it represents belief states as *infinite* objects, namely deductively closed sets of sentences. Another is that existing belief revision models make *too strong assumptions* about what information is available to guide revisions. A consequence of this is that repeated revisions are impossible.

We propose representing belief states by linear ordered theory presentations instead of by theories. Definitions and examples will be given. The result is a finite representation of belief states. The revision operator is shown to satisfy some, but not all, of the AGM postulates. The counterexamples for the AGM postulates which fail are motivated. The important point is that no information other than that encoded in the OTP is needed to effect the revision; this makes repeated revision easy.

The paper is organised as follows. The AGM theory is described in the next section. Section 3 describes the properties which a theory of belief revision should have. The notion of *ordered theory presentations* (OTPs) is introduced in section 4, which is the main section. Sections 5 and 6 discuss the properties of the revision operator on OTPs; and further examples are given in section 7.

2 The AGM theory

The standard theory of belief revision—known as the AGM theory after its authors, C. Alchourrón, P. Gärdenfors and D. Makinson [3]—models belief states as *deductively-closed sets of sentences*. More recent developments of the AGM theory are described in S. O. Hansson's thesis [4]. It describes a small set of postulates which any belief revision operator should satisfy (see below). If K is a belief state and ϕ a formula, then $K * \phi$ is a belief state, the result of revising K with ϕ . As already noted, the case of interest is when $\neg\phi \in K$, that is, when the revising sentence conflicts with the current belief state. We suppose we are utilising classical logic with the usual connectives, and the usual entailment relation $\models$.

Notation 1

1. Let Φ be a set of sentences. $Cn(\Phi) = \{\phi \mid \Phi \models \phi\}$.
2. L is the set of all sentences in the language.
3. $K + \phi = Cn(K \cup \{\phi\})$.

The AGM postulates are the following:

- K1 $K * \phi$ is a deductively-closed theory;
- K2 $\phi \in K * \phi$;
- K3 $K * \phi \subseteq K + \phi$;
- K4 If $\neg\phi \notin K$ then $K + \phi \subseteq K * \phi$;
- K5 $K * \phi = L$ implies $\phi = \perp$;
- K6 If $\models \phi \leftrightarrow \psi$ then $K * \phi = K * \psi$;
- K7 $K * (\phi \wedge \psi) \subseteq K * \phi + \psi$;
- K8 If $\neg\psi \notin K * \phi$ then $K * \phi + \psi \subseteq K * (\phi \wedge \psi)$.

K1 says that $K * \phi$ should be a belief state. K2 says that the revision should be successful, i.e. the resulting theory should at least contain ϕ . The third axiom says that $K * \phi$ should have no more than what we would get by just adding ϕ set-theoretically and closing under entailment. Of course, if ϕ is inconsistent with K then adding it in that way would yield the whole of L (the theory with every sentence in it). K4 asserts that if ϕ is consistent with K then we get precisely the result of adding it set-theoretically. We should point out that this is one of the (two) axioms with which we take issue in §6. K5 says that the revision yields the contradictory theory L only if ϕ is inconsistent. This is not just that ϕ is inconsistent *with* K , but is inconsistent on its own. The converse, that revising with an inconsistent sentence yields the inconsistent theory, is guaranteed by K2. K6 says simply that revising with logical equivalents yields the same theory.

K7 and K8 are more complicated, approximating what happens with repeated revisions. They are analogues of K3 and K4.

Note that K7 and K8 do not contain expressions like $K * \phi * \psi$, and therefore do not constrain repeated revision in any explicit way. The only constraints on repeated revision are those inherited from the more general case of revision which K1–8 describe.

We submit that the AGM axioms to be neither *sound* nor *complete* with respect to intuitively rational belief revision. Of course such a statement is necessarily imprecise, because ‘intuitively rational’ belief revision is not amenable to mathematical description. The argument to show lack of soundness consists of ‘counterexamples’ to K4 and K8 later in the paper. (Again the scare quotes show that these are not counterexamples to any fully spelled-out conjecture.) The argument against completeness is the following proposition, which shows that K1–8 admit revision functions which have no element of the ‘persistence’ requirement (number 2 above).

Proposition 2 The revision function

$$K * \phi = \begin{cases} K + \phi & \text{if } \neg\phi \notin K \\ \text{Cn}\{\phi\} & \text{otherwise} \end{cases}$$

satisfies axioms K1–8.

Proof K1–4 and K6 are immediate.

K5 Suppose $K * \phi = L$. By K3, $K + \phi = L$, so $\neg\phi \in K$. Therefore, $K * \phi = \text{Cn}\{\phi\}$, so $\phi = \perp$.

K7 Suppose $\neg(\phi \wedge \psi) \in K$. Then $K * (\phi \wedge \psi) = \text{Cn}(\phi \wedge \psi)$. If $\neg\phi \in K$ then $(K * \phi) + \psi = \text{Cn}(\phi \wedge \psi)$; otherwise it is $(K + \phi) + \psi$, which contains $\text{Cn}(\phi \wedge \psi)$.

Otherwise $\neg(\phi \wedge \psi) \notin K$, and so $\neg\phi \notin K$. $K * (\phi \wedge \psi) = K + (\phi \wedge \psi) = (K + \phi) + \psi = (K * \phi) + \psi$.

K8 Suppose $\neg\psi \notin K * \phi$. Suppose also that $\neg\phi \notin K$. Then $K * \phi = K + \phi$. Therefore, $K * \phi + \psi = K + \phi + \psi$. Also, since $\neg\psi \notin K * \phi$ and $K * \phi = K + \phi$, we have that $\neg\psi \notin K + \phi$ and therefore $\neg(\phi \wedge \psi) \notin K$. Therefore, $K * (\phi \wedge \psi) = K + (\phi \wedge \psi) = K + \phi + \psi$, as required.

Now suppose $\neg\phi \in K$. Then $\neg(\phi \wedge \psi) \in K$, so $K * \phi + \psi = K + \phi + \psi = K + (\phi \wedge \psi) = K * (\phi \wedge \psi)$. $\diamond$

Of course there are more interesting functions satisfying the axioms. The following two are the most important in the AGM literature: *partial meet* revision; and revision by *epistemic entrenchment*.

2.1 Selection functions

Suppose K is a belief state and ϕ is a sentence other than $\perp$. Let

$$K|_{\phi} = \text{the } \subseteq\text{-maximal elements of } \{K' \subseteq K \mid \neg\phi \notin K'\},$$

that is, the set of maximal subsets of K which are consistent with ϕ . $K|_\phi$ may be pronounced 'K without ϕ '. The operation of partial meet revision assumes a selection function S_K which selects some of these subsets. Then revision is defined by

$$K * \phi = \begin{cases} \text{Cn}(\bigcap S_K(K|_\phi, \phi) \cup \phi) & \text{if } \phi \neq \perp; \\ L & \text{otherwise.} \end{cases}$$

That is to say, if $\phi \neq \perp$ it is the intersection of those ϕ -consistent maximal subsets chosen by S_K with ϕ added set-theoretically. If $\phi = \perp$ it is simply L , the inconsistent theory (the set of all sentences).

It should be clear that this is unsatisfactory, since the whole problem of how to make a revision has just been packaged up in the existence of a selection function, and has not been solved at all. Obviously, the selection function must depend on K . Therefore, we need not bother with the information K alone provides us, since everything we need might just as well be given by this magical S ! The drawback of coding everything in S is that repeated revisions are then impossible.

There is a limiting case of partial meet revision, in which $S_K(K|_\phi, \phi)$ is always a singleton. This case is known as maxichoice contraction. There is another limiting case in which $S_K(K|_\phi, \phi) = \bigcap K|_\phi$, the intersection of all the candidate theories, which is known as full meet revision. The first of these is unsatisfactory for the same reason as the general case, namely that the selection function remains to be defined. (It has other, worse, problems too, detailed in Gärdenfors' book.) The second limiting case does not have this problem, and is worth spelling out in full, since it fully specifies how to carry out a revision without the need for extra information. According to it,

$$K * \phi = \begin{cases} \text{Cn}(\bigcap K|_\phi \cup \phi) & \text{if } \phi \neq \perp; \\ L & \text{otherwise.} \end{cases}$$

It is straightforward to check that this definition satisfies the postulates K1-K8. But there are problems. Consider, for example, how to revise $\text{Cn}(\{p, q\})$ with $\neg p \vee \neg q$. Intuitively, there are at least three plausible answers: $\text{Cn}(\{p\})$, $\text{Cn}(\{q\})$ and $\text{Cn}(\{p \leftrightarrow \neg q\})$. Full meet contraction gives us the last of these, because no information is available to choose whether to give up p or to give up q . But, in practice there may be criteria for choosing to give up one rather than the other. This is what leads to consideration of selection functions, since they could encode the extra information required. But then, as already remarked, repeated revision is impossible. The moral we draw from this situation is different. It is that *deductively closed theories are inadequate as representations of belief states*. We return to this point later, after considering the other main way of providing the information necessary to guide revisions, namely epistemic entrenchment orderings.

2.2 Epistemic entrenchment

Revision by epistemic entrenchment is effected as follows. First we require an epistemic entrenchment ordering on the current belief state. This is a linear pre-order on the sentences in the state, which represents the degree to which they are believed. Those less entrenched according to the ordering are dispensed with more readily in the case of a revision which conflicts with the current state. An epistemic entrenchment ordering for a belief state K must satisfy the following axioms:

- E1 If $\phi \leq_K \psi$ and $\psi \leq_K \chi$ then $\phi \leq_K \chi$ (*transitivity*);
- E2 If $\phi \models \psi$ then $\phi \leq_K \psi$ (*dominance*);
- E3 Either $\phi \leq_K \phi \wedge \psi$ or $\psi \leq_K \phi \wedge \psi$ (*conjunctiveness*);
- E4 If K is consistent then $\phi \leq_K \psi$ for all ψ iff $\phi \notin K$ (*minimality*);
- E5 If $\phi \leq_K \psi$ for all ϕ , then $\models \psi$ (*maximality*).

As in the case of the K postulates, these axioms are intended to encode rationality constraints on what an epistemic entrenchment ordering might be. For example, E2 says that it is always better to give up logically weaker sentences during the course of a revision; therefore, these should be

less entrenched. E3 says that giving up a conjunction is at least as hard as giving up either of the conjuncts. Taken together, axioms E1–E3 imply that $\leq_K$ is a linear order, that is, either $\phi \leq_K \psi$ or $\psi \leq_K \phi$ (or both). E4 says that a sentence is minimally entrenched in K iff it is not in K . E5 says that just the tautologies are maximally entrenched.

Given a belief state K , an epistemic entrenchment ordering $\leq_K$ on K , and a sentence ϕ , the revision of K by ϕ is given by

$$K * \phi = \begin{cases} \text{Cn}(\{\psi \in K \mid \neg\phi <_K \neg\phi \vee \psi\} \cup \{\phi\}) & \text{if } \phi \neq \perp; \\ L & \text{otherwise.} \end{cases}$$

($<$ is the usual strict counterpart of $\leq$, defined by: $\phi < \psi$ if $\phi \leq \psi$ and $\psi \not\leq \phi$.)

Full motivation for the K and EE axioms, as well as for the definition of $*$ in terms of $\leq_K$, can be found in Gärdenfors' book [3].

We now summarise the main weaknesses we have described of the AGM theory. Belief states are represented as deductively closed theories. This means that they are (in general) impossible to write down fully, or to store on a computer. Moreover, as noted, they are incapable of representing the necessary information required to choose between alternative revisions. Therefore, extra information in the form of a selection function or an EE ordering is required. This information is not deemed part of the belief state, and is lost during the revision, making further revision impossible. It is worth pointing out that this means that the real *intention* of axiom K1 is not satisfied by these revision functions. Its intention is that after a revision we should end up with an object of the same type as the one with which we started. Obviously, both partial meet revision and revision by epistemic entrenchment fail this requirement. In those cases we start off with a pair, respectively of type $\langle K, S_K \rangle$ and $\langle K, \leq_K \rangle$, and end up with something of type K .

There are some proposals for modifying the AGM theory to solve some of these problems. For example, some work has been done on theory base revision to address the problem of the infinite nature of deductively closed sets of sentences. In that work, belief states are represented as finite sets of sentences (theory *bases* or theory *presentations*) [1, 5, 9]. But each of these authors assume the existence of something like a selection function or an EE ordering, so are subject to objections on those grounds. There are proposals of non-deterministic revision [6], which alleviate the need for a selection function, but they rely on infinite belief state representations.

There are proposals to allow repeated revision using EE orderings, either by keeping a single EE ordering for all belief states or assuming the existence of a function which, for every belief state, gives an EE ordering [10, 13]. But as neither the single ordering nor this function is itself revised in the course of belief revisions, it is easy to find examples which are in contradiction with intuitions about iterated belief change [4].

Another modification of the AGM theory which allows EE orderings to be revised is given by H. Rott [11]. He defines revision of EE orderings as follows.

$$\psi \leq_{K*\phi} \chi \text{ if } \phi \rightarrow \psi \leq_K \phi \rightarrow \chi.$$

However, as he points out, this fails to capture much of the intuition of repeated revision because any further revision of $K * \phi$ always includes ϕ .

3 Criteria for belief revision

In this section we enumerate what we claim are the criteria by which to judge a theory of belief revision.

1. Finite representation of belief states.
2. Persistence.
3. Iteration: what you put in is what you get out.
4. The "intentions" behind the K axioms of AGM.

The criterion of finite representation means that all belief states can be explicitly written down or represented on a computer. The advantages of this should be easy to see; one in particular is that one can give examples of belief revision in action! (See section 7.)

Persistence means that as much of the former belief state should survive a revision as possible. We rule out revisions like the one of proposition 2.

The iteration criterion says that you should get out of a revision an object of the same type as you put in. As mentioned, this is violated by AGM, since you put in either an EE ordering, or a theory coupled with a selection function; but, all you get out is a theory. We call this *iteration* since, if it obtains, it guarantees that revisions may be repeated. Its absence is a serious problem in AGM.

The last criterion, concerning the K axioms of AGM, is deliberately expressed in a vague way. Obviously, if belief states are not represented as deductively closed sets of sentences then it is impossible to test them literally. Also, as we have noted, they do not specify what should happen under repeated revision, in terms of expressions of the form $K * \phi * \psi$. This is presumably because the AGM models do not support repeated revision. Moreover, for reasons which we will discuss in section 6, we dispute two of the AGM axioms. In view of these reasons, we can only say that something like the intention of the AGM axioms is desirable.

The AGM axioms K1–8 rely on a particular representation of belief states (namely, deductively closed sets of sentences). Therefore, direct comparison with theories of belief revision which use other representations of belief states is impossible. To overcome this we can rewrite the axioms in a more general way, which assumes only the following:

1. A set of belief states, together with a subset of 'contradictory' belief states.
2. A function $*$ (revision) which takes a belief state and a sentence to a belief state;
3. A function $|\cdot|$ (extension) which takes a belief state and returns the set of sentences true in it.

Here are the axioms rewritten in this way. We will write $\mathcal{K}$ for a typical 'abstract' belief state.

- $\mathcal{K}1$ $\mathcal{K} * \phi$ is a belief state;
- $\mathcal{K}2$ $\phi \in |\mathcal{K} * \phi|$;
- $\mathcal{K}3$ $|\mathcal{K} * \phi| \subseteq |\mathcal{K}| + \phi$;
- $\mathcal{K}4$ If $\neg\phi \notin |\mathcal{K}|$ then $|\mathcal{K}| + \phi \subseteq |\mathcal{K} * \phi|$;
- $\mathcal{K}5$ $\mathcal{K} * \phi$ is contradictory implies $\phi = \perp$;
- $\mathcal{K}6$ If $\models \phi \leftrightarrow \psi$ then $|\mathcal{K} * \phi| = |\mathcal{K} * \psi|$;
- $\mathcal{K}7$ $|\mathcal{K} * (\phi \wedge \psi)| \subseteq |\mathcal{K} * \phi| + \psi$;
- $\mathcal{K}8$ If $\neg\psi \notin |\mathcal{K} * \phi|$ then $|\mathcal{K} * \phi| + \psi \subseteq |\mathcal{K} * (\phi \wedge \psi)|$.

4 Ordered theory presentations

Here we present a system for belief revision which satisfies each of the criteria described above. Belief states are represented by *ordered theory presentations*, which were introduced in [12]. The reader unfamiliar with that paper is at no disadvantage, but should ignore the remainder of this subsection, skipping to section 4.1 below. The following explanations will be of use to readers familiar with [12].

In that paper, a 'structured theory' was defined to be a finite, *partially* ordered set of points each of which is labelled by a sentence in the language. We have since changed the name of these objects to 'ordered theory presentations', because what they really are is a way of using an ordering on a set of sentences to present a theory. Thus, they are theory presentations² equipped with a partial order.

²Theory presentations are sometimes also called theory bases.

Here, we are interested only in the linear case, that is to say, in the case that the order is *total*. Thus, we can simplify the definition, letting *linear* ordered theory presentations be simply finite sequences (or lists) of formulas.

4.1 Intuitions

A (linear) ordered theory presentation is a finite list of formulas; $\Gamma = [\phi_1, \phi_2, \dots, \phi_n]$. Here, n is said to be the *length* of Γ . The *extension* of Γ is the deductively-closed theory which Γ presents; that is, it is the set of sentences entailed by Γ , after taking account of the various conflicts in Γ . This is defined precisely later on, but the intuition is the following: Γ presents the theory which first of all has ϕ_n , and then has as much of ϕ_{n-1} as possible while retaining consistency, and then ... up to ϕ_1 . Put another way, we start with ϕ_1 . Then we 'force in' ϕ_2 , overriding as necessary. Then ... and so on until ϕ_n .

The following are examples of belief states.

1. $[p \wedge q]$
2. $[p, q]$
3. $[p \wedge q, \neg p]$

Their lengths are 1, 2 and 2 respectively. OTPs 1 and 2 above both have the extension $\text{Cn}(\{p \wedge q\})$. But in 2, p is less entrenched than q , and will disappear if a revision which demands that one of p and q goes. Thus, we stipulate:

Sentences later in the sequence are more entrenched than those earlier.

Belief state 3 has the extension $\text{Cn}(\{\neg p \wedge q\})$. This is because $\neg p$, which is more entrenched than $p \wedge q$, overrides the p component of $p \wedge q$. But the q component is not overridden. Thus,

Sentences later in the sequence have the effect of overriding those earlier, in the case of conflict.

It should be noted that we have not yet given a definition of extension; this comes in section 4.2. The principles above are to motivate the right intuitions.

It should now come as no surprise to find that

Revision of OTPs is effected by appending the revising sentence to the end of the sequence.

Thus, the three belief states mentioned above can be revised by $\neg p \vee \neg q$, yielding

- 1'. $[p \wedge q, \neg p \vee \neg q]$
- 2'. $[p, q, \neg p \vee \neg q]$
- 3'. $[p \wedge q, \neg p, \neg p \vee \neg q]$

The extension of presentation 1' is $\text{Cn}(\{p \leftrightarrow \neg q\})$, which was the outcome of the corresponding example for full meet revision described above. But presentation 2' has as extension $\text{Cn}(\{q\})$. Since 1 and 2 had the same extension and 1' and 2' do not, it should be clear that *there is more to an OTP than its extension*.

Ordered theory presentation 3' has the extension $\text{Cn}(\{\neg p \wedge q\})$, which is the same as it had before the revision. This is because the revising sentence was consistent with the belief state it revised.

It remains to show how to formalise the computation of extension for ordered theory presentations, which is the task of the next section. First, let us observe some important facts about OTP revision.

1. OTPs have *memory*. If Γ is an OTP, then the extension of $\Gamma * p * q * \neg(p \wedge q)$ includes $q \wedge \neg p$. This is because the theory was more recently revised with q than with p , so q is more entrenched. Older information is discarded more readily than newer.
2. But, information is never wantonly discarded.
3. The more you revise an ordered presentation, the more complicated (= longer) it gets. That is because ordered presentations are nothing more than *revision histories*.

4.2 Extension

In classical logic, a theory presentation denotes a set of models, containing those which satisfy each of the sentences³. Two presentations are equivalent if they denote the same sets. In this section we will define models of ordered presentations, extending the notion of equivalence to OTPs. Thus, the extension of an ordered presentation is the theory of the set of models it denotes. It turns out that

1. Every OTP has a non-trivial extension, and
2. In the case of classical propositional logic at least, the set of models of any OTP is denoted by a finite theory presentation.

Definition 3 An *ordered theory presentation* of length n is a finite list of formulas of length n . Notation: $\Gamma = [\phi_1, \phi_2, \dots, \phi_n]$.

As stated, our aim is to define the models of such ordered theory presentations. We will assume that we are working with a fixed propositional language L , and that $\models$ is the satisfaction relation between interpretations (assignments of truth values to propositional symbols) and sentences of the language. L is ambiguously the language and the set of sentences of the language. Everything here is easily re-worked for predicate logic, or indeed for any logic defined in terms of interpretations and satisfaction. (See [12] for details; occasionally we will add parenthesised notes to this effect.)

Let $\mathcal{M}$ be the set of interpretations of L , and let $\models$ be the satisfaction relation between interpretations and sentences of the language: $\models \subseteq \mathcal{M} \times L$.

To define the models of an ordered theory presentation Γ we need to define an ordering $\sqsubseteq^\Gamma$ on interpretations in $\mathcal{M}$ which measures how well an interpretation satisfies Γ . This relies on orderings $\sqsubseteq_\phi$, one for each sentence ϕ of the language. To define $\sqsubseteq_\phi$, it is necessary to define a notion which we call 'natural entailment', written $\models$. This definition in turn relies on the notion of the *monotonicities* of a sentence. Lest the reader be daunted by these nestings of definitions, we repeat the list here. We define, in order,

1. *Monotonicities* of a sentence ϕ , written $\langle \phi^+, \phi^- \rangle$.
2. *Natural entailment*, a relation $\models$ between sentences, being a sub-relation of ordinary entailment $\vdash$.
3. For each sentence ϕ , a reflexive and transitive order $\sqsubseteq_\phi$ on the interpretations of the language $\mathcal{M}$; as will be seen, this order grades interpretations according to how nearly they satisfy ϕ .
4. For each ordered presentation Γ , a reflexive and transitive order $\sqsubseteq^\Gamma$ on $\mathcal{M}$; this order grades interpretations according to how well they satisfy Γ .
5. The models of Γ , being a set $\llbracket \Gamma \rrbracket \subseteq \mathcal{M}$.

For the definition of the monotonicities of ϕ , we need the following notation. If M is an interpretation of L and p is a propositional symbol in L , then $M^{[p \mapsto t]}$ is an interpretation identical with M except possibly that it assigns **true** to p . (If M already assigns **true** to p then $M^{[p \mapsto t]}$ is simply M .) $M^{[p \mapsto f]}$ is defined analogously.

Definition 4 Let ϕ be a sentence *other than* $\perp$ and p any propositional symbol.

1. ϕ is *monotonic in* p if $M \models \phi$ implies that $M^{[p \mapsto t]} \models \phi$.
2. ϕ is *anti-monotonic in* p if $M \models \phi$ implies that $M^{[p \mapsto f]} \models \phi$.
3. ϕ^+ and ϕ^- are the sets of symbols in which ϕ is monotonic and anti-monotonic respectively.

³Of course, the same goes for many non-classical logics too, including intuitionistic logic, modal logics, temporal logics etc. Readers familiar with [12] will recall that ordered theory presentations are defined over any logic defined in terms of models and satisfaction.

The case that $\phi = \perp$ is handled separately; we define $\perp^+ = \perp^- = \emptyset$.

Thus, ϕ is monotonic in p if "increasing" the truth value of p in a model of ϕ preserves satisfaction of ϕ . Similarly, ϕ is anti-monotonic in p if decreasing the truth value so preserves satisfaction. (In the case that L is a predicate language, one must define the *extension* of a predicate symbol P in a model to be the set of tuples which satisfy P in the model. ϕ is (anti-)monotonic in the predicate symbol P if increasing (decreasing) the extension of P in a model of ϕ also results in a model of ϕ .)

Example 5 Suppose L has just the propositional symbols p and q .

ϕ	ϕ^+	ϕ^-
$\top$	$\{p, q\}$	$\{p, q\}$
p	$\{p, q\}$	$\{q\}$
q	$\{p, q\}$	$\{p\}$
$p \wedge q, p \vee q$	$\{p, q\}$	$\emptyset$
$p \rightarrow q$	$\{q\}$	$\{p\}$
$p \leftrightarrow q$	$\emptyset$	$\emptyset$
$\perp$	$\emptyset$	$\emptyset$

Having defined monotonicities, we turn to point 2 of the five-point plan mentioned above, i.e. the definition of natural entailment. Let ϕ and ψ be sentences of L .

Definition 6 ϕ *naturally entails* ψ , written $\phi \models \psi$, if

1. $\phi \models \psi$, and
2. $\phi^+ \subseteq \psi^+$ and $\phi^- \subseteq \psi^-$

Natural entailment is a sub-relation of ordinary entailment; in addition to ordinary entailment we require that the monotonicities of the premise be preserved by the conclusion.

Proposition 7 $\models$ is reflexive and transitive⁴. ◇

Example 8 The relations $\models$ and $\models$ on the set of sentences formed from the language containing the propositions $\{p, q\}$ are shown in figure 1 for comparison. Thus: $p \wedge q \models p$ and $p \wedge q \models p \vee q$, but $p \wedge q \not\models p \leftrightarrow q$ and $p \not\models p \vee q$. Moreover, $\perp \models \phi$ for all ϕ .

The definition of natural entailment is perhaps not very satisfying, because (one might ask), what is so special about preserving monotonicities? One way to answer this is purely pragmatic: as we will see, it is essential for the next definition, which does have a satisfying feel. But first, we justify the term *natural entailment* by showing examples of how much more natural this entailment really is.

Natural entailment is something like relevant entailment; it stops us adding irrelevant disjuncts in our conclusions. (This is not the same notion of relevance as Anderson/Belnap, for there one is interested in stopping irrelevant conjuncts in the premisses.) The following entailments, which are ordinarily valid, are not naturally valid:

$$\begin{array}{lll} p \models p \vee q & p \models q \rightarrow p & p \wedge q \models p \leftrightarrow q \\ p \models p \vee \neg q & \neg p \models p \rightarrow q & \end{array}$$

Regarding the first pair, the premise p tells us nothing about q , and therefore it is suspect to introduce q or $\neg q$ as a disjunct. The second pair are the standard inelegancies of material implication, which everyone will be glad to see the back of. Finally, we dislike $p \wedge q \models p \leftrightarrow q$ because

⁴the proofs of this proposition and others in this section which are given without proofs may be found in [12].

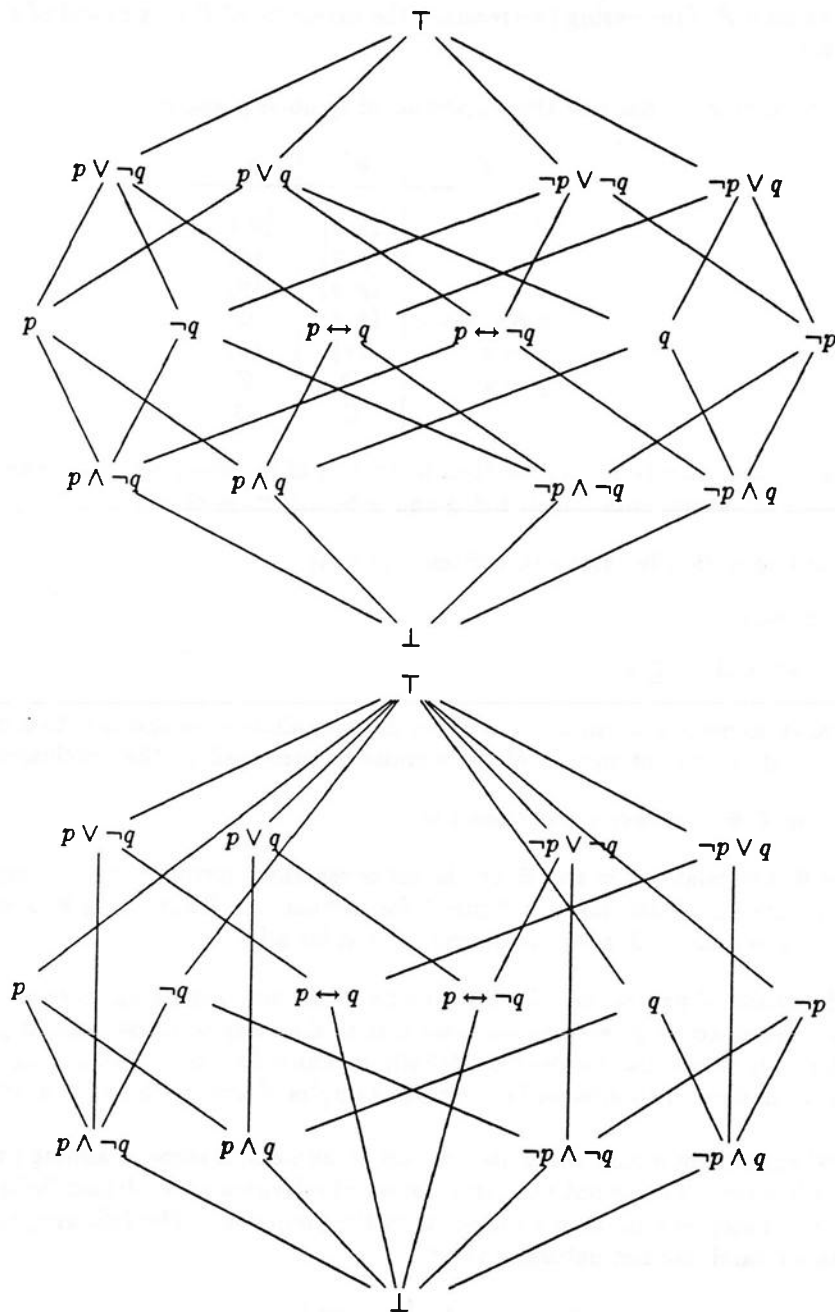


Figure 1: The *ordinary* and *natural* entailment relations over $\{p, q\}$

the right-hand side suggests that p and q are in some way bound together, whereas the left-hand side only says that they are both true.

On the other hand, the simplicity of the definition and the fact that it is based on satisfaction by models ensures that there is nothing untoward going on. In particular, if ϕ and ψ are classically equivalent then they are naturally equivalent; indeed

$$(\phi \models \psi \text{ and } \psi \models \phi) \text{ iff } (\phi \vDash \psi \text{ and } \psi \vDash \phi)$$

I really do think there is something wonderful about natural entailment, even though I only stumbled upon it as a means of achieving the following definition of $\sqsubseteq_\phi$. More in another paper!

How, then, to define $\sqsubseteq_\phi$? As stated, $M \sqsubseteq_\phi N$ means that N is as good at satisfying ϕ as M is. It is not just that N satisfies ϕ and M does not; perhaps neither satisfy ϕ , but N more nearly does. For example, let M be a propositional interpretation which assigns false to both p and q ; and let N assign true and false to p and q respectively. Then $M \sqsubseteq_{p \wedge q} N$, while $N \not\sqsubseteq_{p \wedge q} M$. Neither satisfy $p \wedge q$, but at least N satisfies p ; M doesn't satisfy either of p and q .

This example shows that one has to look at which *consequences* of ϕ are satisfied by M and N . However, defining $M \sqsubseteq_\phi N$ to mean that N satisfies all the consequences of ϕ which M does gives us precisely the bipartite ordering rejected in the preceding paragraph. This is because ϕ has too many irrelevant consequences; we should just look at the *natural* ones.

Definition 9 $M \sqsubseteq_\phi N$, if for each ψ ,

$$\phi \vDash \psi \Rightarrow (M \Vdash \psi \Rightarrow N \Vdash \psi)$$

We can show that $\sqsubseteq_\phi$ has precisely the mathematical behaviour we want.

Proposition 10 1. $\sqsubseteq_\phi$ is a pre-order, that is to say, it is reflexive and transitive.

2. If $\phi \neq \perp$, the maximal elements of $\sqsubseteq_\phi$ (which are in fact *maximum*) are just the models of ϕ .

3. The orderings $\sqsubseteq_\perp$ and $\sqsubseteq_\top$ are the indiscrete ordering; that is, $\sqsubseteq_\perp = \sqsubseteq_\top = \mathcal{M} \times \mathcal{M}$. $\diamond$

The proofs of these assertions, together with many examples of $\sqsubseteq_\phi$ for various sentences ϕ , can be found in [12].

We have defined, for each sentence ϕ , an ordering on interpretations $\sqsubseteq_\phi$ which measures the extent to which interpretations satisfy ϕ . If M satisfies ϕ to the fullest extent (that is, if it simply satisfies it) then M is $\sqsubseteq_\phi$ -maximum. If M does not fully satisfy ϕ then it may satisfy it to a greater, lesser, equal or incomparable extent than some N which perhaps also fails fully to satisfy ϕ . As stated, examples of this ordering on models for various sentences ϕ can be found in [12].

We continue with the five-point plan at the beginning of this section. We define $\sqsubseteq^\Gamma$ by induction on Γ . If Γ is the empty list $[]$, then $M \sqsubseteq^\Gamma N$ for all M, N . Otherwise, if Γ is $[\phi_1, \dots, \phi_n]$ then $M \sqsubseteq^\Gamma N$ if either $M \sqsubseteq_{\phi_1} N$, or $M \sqsubseteq_{\phi_1} N$ and also $M \sqsubseteq^{\Gamma'} N$, where Γ' is $[\phi_2, \dots, \phi_n]$. In other words, to determine whether N is as good at satisfying Γ as M is, one looks at the most important sentence. If N is strictly better than M as far as that sentence is as concerned, then N is certainly as good overall. Otherwise, it is at least necessary for N to be as good on that sentence, and the question of whether it is as good overall defers to the remaining sentences which are treated in a similar way. We summarise:

Definition 11

1. $M \sqsubseteq^{[]} N$ always; and
2. $M \sqsubseteq^{\Gamma * \phi} N$ if $M \sqsubseteq_\phi N$ or $(M \sqsubseteq_\phi N \text{ and } M \sqsubseteq^\Gamma N)$.

Remark 12 $M \sqsubseteq^{\Gamma * \phi} N$ if $M \sqsubseteq_\phi N$ or $(M \sqsubseteq_\phi N \text{ and } M \sqsubseteq^\Gamma N)$.

$\Gamma * \phi$ is Γ with ϕ appended. Finally, it is easy to define the models of Γ . They are simply the interpretations which are rated maximally by $\sqsubseteq^\Gamma$.

Definition 13 $M \Vdash \Gamma$ if M is $\sqsubseteq^\Gamma$ -maximal.

Naturally we expect that the highest priority sentence is satisfied by models of the theory:

Proposition 14 Let $\Gamma = [\phi_1, \phi_2, \dots, \phi_n]$ be an OTP and let $M \Vdash \Gamma$. If $\phi_n \neq \perp$ then $M \Vdash \phi_n$. $\diamond$

Definition 15 Let Γ be an ordered theory presentation.

1. The *extension* of Γ , written $|\Gamma|$, is the theory of the set of models of Γ :

$$|\Gamma| = \{\phi \mid M \Vdash \Gamma \text{ implies } M \Vdash \phi\}$$

2. The *consequences* of Γ are the sentences in its extension:

$$\Gamma \models \phi \text{ if } \phi \in |\Gamma|.$$

4.3 Summary of definitions

In this section we summarise the position so far. We assume we are working in classical logic. Ordered theory presentations consist of finite lists of sentences in the language (def. 3). To define the models of an ordered theory presentation, we first define the monotonicities $\langle \phi^+, \phi^- \rangle$ of each sentence ϕ (def. 4). This is a pair of sets of atomic sentences. Then we define the relation of natural entailment between sentences (def. 6). We claim, in passing, that this has intuitive properties which ordinary entailment fails, but our main purpose is to use it to define the degree of satisfaction between an interpretation and a sentence. We do this by ordering interpretations according to how well they satisfy a particular sentence ϕ , in definition 9. These ordering are used to define an ordering for the whole OTP which measures how well interpretations do at satisfying it. Finally, its models are the interpretations maximal in this ordering. The extension of an OTP and its consequences are then straightforward to define (def. 15).

5 Properties of OTPs

Proposition 16 Every ordered presentation has a model.

Proof This is a complicated proof, using Zorn's lemma to find maximal elements in the order $\sqsubseteq^\Gamma$. For details, see [12]. $\diamond$

A consequence of this result is that contradictions can never be derived from an ordered presentation, not even one with the contradictory sentence in it!

Proposition 17 If $\Gamma \models \phi$ then $\phi \neq \perp$.

Proof Let $M \Vdash \Gamma$. Since $M \Vdash \phi$, $\phi \neq \perp$. $\diamond$

Definition 18 Let Γ and Δ be OTPs.

1. Γ and Δ are *statically equivalent*, written $\Gamma \equiv \Delta$, if they have the same extension:

$$\Gamma \equiv \Delta \text{ if } |\Gamma| = |\Delta|.$$

2. Γ and Δ are *dynamically equivalent*, if, for all ϕ , $\Gamma * \phi \equiv \Delta * \phi$.

Dynamic equivalence implies static equivalence, but the converse is not so as the following example shows.

Example 19 $[p, q] \equiv [p \wedge q]$, since both have the models $\{11\}$ in the obvious notation. But $[p, q, \neg p \vee \neg q] \not\equiv [p \wedge q, \neg p \vee \neg q]$, since the model sets are respectively $\{01\}$ and $\{01, 10\}$.

However, if $\Gamma \models \phi$ we would not expect that revising Γ by ϕ should change the set of models.

Proposition 20 If $\Gamma \models \phi$ then $\Gamma \equiv \Gamma * \phi$. ◇

We also obtain weak analogues of the usual structural properties:

Proposition 21 1. Weak inclusion: if $\phi \neq \perp$ then $\Gamma * \phi \models \phi$

2. Weak monotonicity:
$$\frac{\Gamma \models \phi \quad \Gamma \models \psi}{\Gamma * \phi \models \psi}$$

3. Weak cut:
$$\frac{\Gamma * \phi \models \psi \quad \Gamma \models \phi}{\Gamma \models \psi}$$

These principles are accepted as being requirements which a system for belief revision or for defaults should have (see for example [2, 7, 8]).

6 Belief revision: the AGM postulates

As stated, we intend to use these ordered theory presentations as representations of belief states in order to model belief revision. The obvious way to do this is to let

belief states = ordered theory presentations

and define $\Gamma * \phi$ to be Γ with ϕ appended; of course we have been implicitly assuming this definition so far in the paper. Note that under this arrangement there are no contradictory theories (proposition 17).

In this setting, we can investigate the truth or falsity of the abstract K axioms given in section 3. We obtain the following.

K1 $\Gamma * \phi$ is a belief state.

This is true. If Γ is an OTP then so is $\Gamma * \phi$.

K2 $\phi \in |\Gamma * \phi|$.

This is false. For example, $\perp \notin [] * \perp$; for, as one can check, $[] * \perp = \text{Cn}(\emptyset)$. However, K2 is true if $\phi \neq \perp$, by proposition 14.

K3 $|\Gamma * \phi| \subseteq |\Gamma| + \phi$.

True. We need to show that $M \Vdash \Gamma$ and $M \Vdash \phi$ imply $M \Vdash \Gamma * \phi$. Suppose not, i.e. suppose $M \sqsubset^{\Gamma * \phi} N$ for some N . By lemma 12, either $M \sqsubset_{\phi} N$, which contradicts $M \Vdash \phi$ (proposition 10) or $M \sqsubset^{\Gamma} N$, which contradicts $M \Vdash \Gamma$ (definition 13).

K4 If $\neg\phi \notin |\Gamma|$ then $|\Gamma| + \phi \subseteq |\Gamma * \phi|$

This is false. Let $\phi_1 = p \wedge q \wedge r$, $\phi_2 = \neg p \vee \neg q \vee \neg r$ and $\phi_3 = (p \leftrightarrow q) \vee \neg r$. The counterexample is obtained by setting: $\Gamma = [\phi_1, \phi_2]$ and $\phi = \phi_3$. To see this, we should first examine the orderings for each of ϕ_1 , ϕ_2 and ϕ_3 . They are shown in the top half of figure 2. Applying definition 11, the orderings $\sqsubseteq^{\Gamma}$ and $\sqsubseteq^{\Gamma * \phi}$ (i.e. $\sqsubseteq^{[\phi_1, \phi_2]}$ and $\sqsubseteq^{[\phi_1, \phi_2, \phi_3]}$ respectively) are as shown in the bottom half of the figure. We can check the following:

- $\neg\phi_3 \notin [[\phi_1, \phi_2]]$, that is to say, there is a model M such that M is $\sqsubseteq^{[\phi_1, \phi_2]}$ -maximal and $M \not\models \neg\phi_3$. Such an M is 110. Thus, the antecedent of K4 holds.

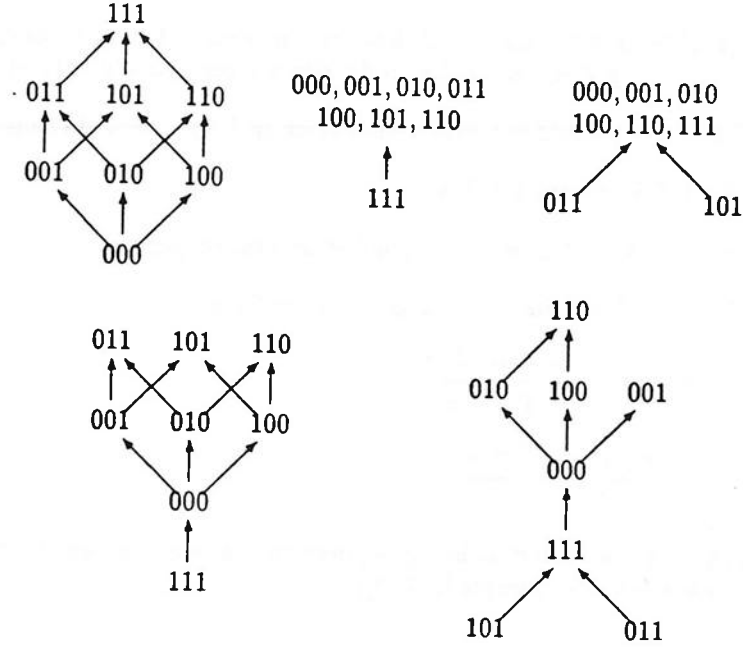


Figure 2: The counterexample to $\mathcal{K}4$ (see text)

- But the consequent is false. For if $|\Gamma| + \phi \subseteq |\Gamma * \phi|$ then $M \Vdash \Gamma * \phi$ implies $M \Vdash |\Gamma| + \phi$, i.e. $M \Vdash \Gamma$ and $M \Vdash \phi$. But by inspecting the diagrams we can find M such that $M \Vdash \Gamma * \phi$ but $M \not\Vdash \Gamma$, namely $M = 001$.

$\mathcal{K}5$ $\Gamma * \phi$ is contradictory implies $\phi = \perp$.

This is vacuously true since there are no contradictory belief states.

$\mathcal{K}6$ If $\models \phi \leftrightarrow \psi$ then $|\Gamma * \phi| = |\Gamma * \psi|$.

True. Suppose $\models \phi \leftrightarrow \psi$. It is sufficient to prove $\sqsubseteq_\phi = \sqsubseteq_\psi$. Suppose $M \sqsubseteq_\phi N$, and $\psi \models \chi$ and $M \Vdash \chi$. By reflexivity, $\phi \models \psi$, so by transitivity $\phi \models \chi$. Therefore, $N \Vdash \chi$, so $M \sqsubseteq_\psi N$. The converse is proved similarly.

$\mathcal{K}7$ $|\Gamma * (\phi \wedge \psi)| \subseteq |\Gamma * \phi| + \psi$.

True. We need to show that if $M \Vdash \Gamma * \phi$ and $M \Vdash \psi$ then $M \Vdash \Gamma * (\phi \wedge \psi)$. If $\phi = \perp$ then $\Gamma * \phi \equiv \Gamma * (\phi \wedge \psi)$, and we are done. So suppose $\phi \neq \perp$, and $M \Vdash \Gamma * \phi$ and $M \Vdash \psi$, but $M \not\sqsubseteq^{\Gamma * (\phi \wedge \psi)} N$ for some N . Since $M \Vdash \Gamma * \phi$ and $\phi \neq \perp$, we have $M \Vdash \phi$ by proposition 14. Therefore, $M \Vdash \phi \wedge \psi$. By lemma 12, either $M \sqsubseteq_{\phi \wedge \psi} N$, which contradicts $M \Vdash \phi \wedge \psi$, or $M \sqsubseteq^\Gamma N$. But this also leads to a contradiction, for then, since $M \sqsubseteq_\phi N$, we obtain $M \sqsubseteq^{\Gamma * \phi} N$ by lemma 12, contradicting $M \Vdash \Gamma * \phi$.

$\mathcal{K}8$ If $\neg\psi \notin |\Gamma * \phi|$ then $|\Gamma * \phi| + \psi \subseteq |\Gamma * (\phi \wedge \psi)|$

False. The counterexample given for $\mathcal{K}4$ holds here too. Set $\Gamma = [p \wedge q \wedge r]$, $\phi = \neg p \vee \neg q \vee \neg r$ and $\psi = (p \leftrightarrow q) \vee \neg r$.

On this way of using OTPs as belief states, we have shown that $\mathcal{K}1$, $\mathcal{K}3$, $\mathcal{K}5$, $\mathcal{K}6$ and $\mathcal{K}7$ are valid; that $\mathcal{K}2$ is valid under the proviso that $\phi \neq \perp$; and that $\mathcal{K}4$ and $\mathcal{K}8$ are not valid.

It is worth pointing out that the lack of contradictory belief states and the partial failure of $\mathcal{K}2$ are easily solved, by adding a new belief state to represent the contradictory belief state and modifying the definition of revision. Thus,

belief states = ordered theory presentations $\cup \{\perp\}$.

Revision on these belief states is defined as follows:

$$\Gamma \star \phi = \begin{cases} \perp & \text{if } \phi = \perp \\ [\phi] & \text{if } \phi \neq \perp \text{ and } \Gamma = \perp \\ \Gamma * \phi & \text{otherwise} \end{cases}$$

This emulates what the AGM axioms intend for $\perp$, in that

1. There is a unique contradictory belief state.
2. Revising any state with the contradictory sentence results in the contradictory state (K2).
3. The contradictory state can only be obtained in this way (K5), so in particular
4. Revising the contradictory state with a non-contradictory sentence will *not* result in the contradictory state.

For the psychological plausibility of these stipulations, or otherwise, see [3]. Especially the first one is debatable! Our point is simply that if we take this definition of $\Gamma \star \phi$ on board, we obtain that $\mathcal{K}1$, $\mathcal{K}2$, $\mathcal{K}3$, $\mathcal{K}5$, $\mathcal{K}6$, and $\mathcal{K}7$ are satisfied, and $\mathcal{K}5$ is satisfied in a more satisfying manner. $\mathcal{K}4$ and $\mathcal{K}8$ are still false for the same reasons.

6.1 The AGM axioms $\mathcal{K}4$ and $\mathcal{K}8$

$\mathcal{K}4$ and $\mathcal{K}8$ are serious violations of the AGM axioms, and there is no easy way of making them satisfied in the framework of OTPs. One must face the question: are they desirable axioms for belief revision? We believe the answer is no.

Consider the diagrams given in figure 2. As far as our counterexample is concerned, the question of the validity of $\mathcal{K}4$ hinges on whether $001 \sqsubseteq_{\phi_1} 110$ or not. If this was so, then we would also have $001 \sqsubseteq_{[\phi_1, \phi_2, \phi_3]} 110$ and $[\phi_1, \phi_2, \phi_3]$ would have only the model 110. Therefore, $\mathcal{K}4$ (and $\mathcal{K}8$) would hold.

Should $001 \sqsubseteq_{p \wedge q \wedge r} 110$ be the case? At first sight it seems clear that 110 is better at satisfying $p \wedge q \wedge r$ than 001 is, for 110 satisfies two of the atomic propositions while 001 satisfies only one. But this kind of cardinality argument is flawed. Why is it better to satisfy $p \wedge q$ rather than r ? Perhaps r itself expresses a conjunction of facts. Are two oranges better than one apple?

The AGM book does not provide any argument in favour of $\mathcal{K}4$ and $\mathcal{K}8$. Consider the following story. I am expecting a friend called John to arrive. He can come by car, bike, or train. I am doubtful about whether he will arrive or not, however, because I believe that his car and bike are both at the repairers; and also, the trains are not working today (for a change). Let:

- p mean that his car is unavailable for use
- q his bike is unavailable
- r the trains are unavailable

Initially I believe

$$p \wedge q \wedge r.$$

Now suppose John actually arrives. I have no reason to doubt that he came by one of the usual means of transport (for example, he didn't ask me for money for a taxi). Therefore I revise my beliefs by

$$\neg p \vee \neg q \vee \neg r.$$

In the course of conversation it turns out that the repairer phoned him this morning to say that both his car and his bike were available for collection. I reason as follows. If the trains are still not working, he may have asked Richard for a lift to the repairer. His bike fits in the back of Richard's car, so then they could have collected both items. But, Richard may have been unavailable or unwilling. Either way, he will have collected both items or neither, so I revise with:

$$r \rightarrow (p \leftrightarrow q)$$

If the trains are working ($\neg r$) I cannot draw the conclusion $p \leftrightarrow q$, since he may have gone by train to pick up either the car or the bike, or neither, or he may still have asked Richard and got both.

The question now is: have I got enough information to conclude which means of transport were available for John to use?

We believe the answer is no. Suppose r , that is, the trains are still not working. I have already reasoned that this implies $p \leftrightarrow q$, and since John is actually here (so $\neg p \vee \neg q \vee \neg r$), it must be that $\neg p \wedge \neg q$. Therefore, $\neg p \wedge \neg q \wedge r$. On the other hand, suppose $\neg r$, i.e. that the trains are working. This tells me nothing about p and q . But since I started with the belief that $p \wedge q$ and John's arrival (by train, presumably) is consistent with these, I retain them. Therefore, $p \wedge q \wedge \neg r$. So I conclude $(\neg p \wedge \neg q \wedge r) \vee (p \wedge q \wedge \neg r)$, or, equivalently, $(p \leftrightarrow q) \wedge (p \leftrightarrow \neg r)$.

We have argued that it is not rational to conclude $\neg r$ in this case. We have also noted that the theory of belief revision outlined in this paper does not conclude $\neg r$. Indeed, we have argued that it concludes precisely what it is rational to conclude. It should be pointed out in fairness to the AGM theory that it does not insist on $\neg r$ either. To see this, consider what happens if the revision function specified in proposition 2 is applied to the revision history in question. We get

$$\begin{aligned} \text{Cn}\{p, q, r\} * (\neg p \vee \neg q \vee \neg r) * (r \rightarrow (p \leftrightarrow q)) \\ = \text{Cn}\{\neg p \vee \neg q \vee \neg r\} * (r \rightarrow (p \leftrightarrow q)) \\ = \text{Cn}\{(p \wedge q) \vee \neg r\} \end{aligned}$$

$\neg r$ is not derivable from this theory.

What we have shown is that if we augment the system of OTPs for belief revision so as to obtain $\mathcal{K}4$ and $\mathcal{K}8$, then we would have a system which concluded $\neg r$ in this case, which is undesirable.

7 Examples

Here we list some facts about linear OTPs, together with some references to examples in the literature to which the facts seem relevant.

$$\begin{aligned} |[p]| &= \text{Cn}(\{p\}) \\ |[p, q]| &= \text{Cn}(\{p, q\}) \\ |[p, q, \neg q]| &= \text{Cn}(\{p, \neg q\}) \\ |[p, q, \neg p]| &= \text{Cn}(\{\neg p, q\}) \\ |[p \wedge q, \neg p]| &= \text{Cn}(\{\neg p, q\}) \\ |[p \wedge q, \neg p \vee \neg q]| &= \text{Cn}(\{p \leftrightarrow \neg q\}) \\ |[p, q, \neg p \vee \neg q]| &= \text{Cn}(\{\neg p, q\}) \\ |[p \vee q, \neg q]| &= \text{Cn}(\{p, \neg q\}) \end{aligned}$$

We also have that

$$s \rightarrow p \in |[s, s \rightarrow p, s \rightarrow q, \neg q, \neg p]|$$

(cf. Hansson [4, page 7:12]), and, for example,

$$\begin{aligned} p \leftrightarrow q \in |[p, q]|, \quad \text{but} \quad p \leftrightarrow q \notin |[p, q, \neg p]| \\ p \leftrightarrow q \in |[p, p \leftrightarrow q]| \quad \text{and} \quad p \leftrightarrow q \in |[p, p \leftrightarrow q, \neg p]| \end{aligned}$$

(cf. [4, page 4:3]).

References

- [1] A. Fuhrmann. Theory contraction through base contraction. *Journal of Philosophical Logic*, 20:175-203, 1991.
- [2] D. M. Gabbay and M. R. B. Clarke. An intuitionistic basis for non-monotonic logic. In P. Smets, A. Mamdani, D. Dubois, and H. Prade, editors, *Non-standard Logics for Automated Reasoning*. Academic Press, 1988.

- [3] P. Gärdenfors. *Knowledge in Flux: Modelling the Dynamics of Epistemic States*. MIT Press, 1988.
- [4] S. O. Hansson. *Belief Base Dynamics*. PhD thesis, Department of Philosophy, Uppsala University, 1991.
- [5] S. O. Hansson. From logical atoms to basic beliefs. Submitted for publication, 1992.
- [6] S. Lindström and W. Rabinowicz. Epistemic entrenchment with incomparabilities and relational belief revision. In A. Fuhrmann and M. Morreau, editors, *The Logic of Theory Change*, volume 465 of *LNAI*. Springer Verlag, 1991.
- [7] D. Makinson. General patterns in non-monotonic reasoning. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence*. Oxford University Press, 1992. Forthcoming.
- [8] D. Makinson and P. Gärdenfors. Relations between the logic of theory change and non-monotonic logic. To appear.
- [9] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems*. Springer Verlag LNAI 422, 1990.
- [10] H. Rott. Preferential belief change using generalised epistemic entrenchment. *Konstanzer Berichte zur Logik und Wissenschaftstheorie* 15.
- [11] H. Rott. Two methods of constructing contractions and revisions of knowledge systems. *Journal of Philosophical Logic*, 20:149–173, 1991.
- [12] M. D. Ryan. Defaults and revision in structured theories. In *IEEE Symposium on Logic in Computer Science (LICS)*, pages 362–373, 1991.
- [13] K. Schlechta. Some results on theory revision. In A. Fuhrmann and M. Morreau, editors, *The Logic of Theory Change*, volume 465 of *LNAI*. Springer Verlag, 1991.

The Generation of Noun Phrases: Adequacy and Anaphora

Kjell Johan Sæbø
University of Oslo

Abstract

This paper investigates the semantics of anaphora in the generative perspective, centering on the so-called Principle of Adequacy for the generation of referential noun phrases, which states that the NP should be sufficient to identify the intended referent. I present an algorithm for generating discourses with definite and indefinite NPs, including definite descriptions containing pronouns, from discourse representations. In this framework, I develop a formal definition of PA and show how it must be modified for the discourse to preserve the truth conditions of the representation. The requirement that the NP identify the familiar referent is too strong and must be replaced by the requirement that it distinguish it from every other distinguishable referent. And, adequacy cannot be stated in terms of an NP alone but only in terms of the tuple of NPs at top level in a sentence.

Introduction

When generating NPs in a dynamic environment, where what is generated depends on what has been generated, it is often necessary - for the discourse to preserve the truth conditions of the discourse representation, so that it is possible to revert to it - to use definite NPs. In fact, whenever a referent occurs twice in the representation, the second NP ought to express coreference with the first. Robert Dale (1988: 161) formulates three principles for the problem of generating referential noun phrases, the three «Principles of Reference»:

1. The *principle of sensitivity* states that the referring expression chosen should take account of the hearer's knowledge.
2. The *principle of adequacy* states that the referring expression chosen should be sufficient to identify the intended referent.
3. The *principle of efficiency* states that the referring expression chosen should provide no more information than is necessary for the identification of the intended referent.

One could say that Adequacy concerns truth conditions while Sensitivity concerns presuppositions. I concentrate on the Principle of Adequacy (PA). Appelt (1985), Novak (1988), and Dale (1988) represent attempts at satisfying this requirement. As yet, however, there is no treatment of definite descriptions containing pronouns, and there is no satisfactory formal definition of the principle.

A formal definition must be set in a formal framework. To this end, I have written a program that generates discourses with definite or indefinite noun phrases from discourse representations. The greater framework is Discourse Representation Theory. I presuppose an analysis of definite and indefinite noun phrases as in Kamp (1981) or Heim (1982). While there are several implementations of anaphora interpretation in this framework - eg. Johnson / Klein (1986), Johnson / Kay (1990) - treatments of anaphora generation are scarce.

My aim is twofold: To construct an algorithm which demonstrates the possibility of treating noun phrases as parts of other noun phrases, ie. in relative clauses, and to use such an algorithm in an effort to provide a satisfactory formal definition of the Principle of Adequacy. I suggest a succession of definitions of this principle on the basis of the generation algorithm, and I stress the importance of what I call 'semi-definite' descriptions in noun phrase generation.

A Survey of the System

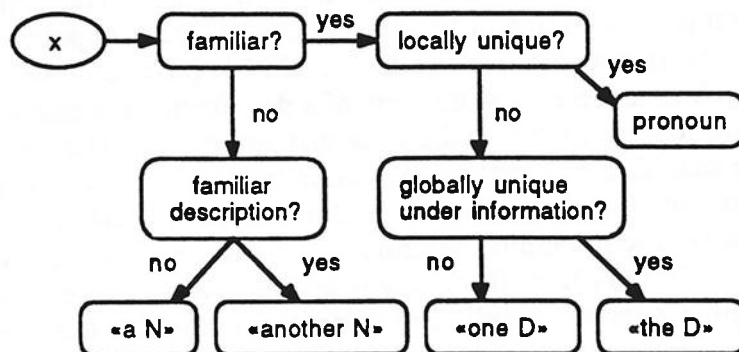
0 Overview

The algorithm is based on a simplistic semantic representation, corresponding to discourse representation structures without quantifiers. For example, the following results in the discourse «A farmer owns a donkey. He beats it.»:

[farmer(x), donkey(y), owns(x,y), beats(x,y)].

If the referent to be «covered» by the noun phrase under construction is familiar, not from the local but from the global history of the discourse, an attempt is made at generating a definite description. In comparison with a pronoun, the definite description adds information to guide the identification. The test is whether the referent is unique under the description: There is no other familiar referent with the same properties. Information may have to be added in many stages, and the preliminary description is step by step tested for uniqueness, until all accessible information on the referent is consumed. Should uniqueness fail ultimately, the definite determiner «the» yields to the semi-definite determiner «one», signalling familiarity in spite of non-uniqueness.

The overall noun phrase generation algorithm looks as follows.



1 The Representation

The algorithm is based on a conceivably simple semantic representation language. It contains discourse referents, unary and binary relation symbols, and commas for conjunctions. Unary relations are always nouns, and one referent is never ascribed two different unary relations. The representation has two limiting characteristics: It is scopeless, and, it is omniscient. Scopeless means that there are no quantifiers with scope. The only «quantifiers» are implicit existential quantifiers which bind globally in the sense that variables with the same name are understood to corefer. This corresponds to a discourse representation structure (DRS) without quantifiers. But unlike a DRS, a structure of my representation does not present a «cast» of discourse referents at the top, and it is in linear list format. For example, the following corresponds to the discourse «A farmer owns a donkey. He beats it.».

[farmer(x),donkey(y),owns(x,y),beats(x,y)].

The representation is omniscient in the following sense. Two variables which do not have the same name are understood to not corefer. This implies that a second mention of a noun corresponds to the determiner «another» instead of the indefinite article «a». The reason for this choice is that the indefinite article is not so neutral after all in the generative perspective. When unfamiliar referents are to be generated in the form of noun phrases, an indefinite description is frequently problematic if the property is familiar. I return to this point in section 4.

Lexical relations like hyponymy are ignored, so if we have man(x) and farmer(y), «the man» is to refer unambiguously to x. The generation algorithm is deterministic in the sense that a representation structure gives rise to just one discourse.

2 The Scheme

The aim of the generation of surface discourse from representation structure is to retain the truth conditions of the structure in the discourse. This is no trivial affair. It depends on the recognition of familiar entities and their proper characterisations. When a referent is recognised as familiar, ie. already processed, we must attempt to identify it with its antecedent in the form of a definite noun phrase. A definite noun phrase is at its least a pronoun in the sense that pronouns carry the least information to guide the identification. Next comes a definite description, which may have to be more or less complex. At the bottom of this definiteness hierarchy comes what I propose to term a semi-definite description. This is the description we use when a referent is familiar but identification fails because the information on it applies to another familiar referent as well. The common locution for this is the determiner «one of the» combined with a plural N.

Let me present the superordinate loop rule of the Prolog generation program:

```
generate(F,R,G,H) :-    member(P,F), P=..(V,X,Y),  
                        n1(F,G,H,K,X,P1,N1), n2(F,G,H,K,Y,P2,N2),  
                        append(N1,VP,S), append((V),N2,VP),  
                        append(S,(new||R1),R), append((P2,P1,P),G,G1),  
                        minus(F,(P,P1,P2),F1),  
                        generate(F1,R1,G1,(P2,P1,P)).
```

The variable F stands for the logical form from which to generate the discourse which the variable R stands for. G represents the global and H the local history of the current stage of the discourse. It is necessary to keep a record of several levels of context in order to characterise different context-dependencies: Basically, a pronoun is locally «bound» while the global context is what governs the distribution of definite, semi-definite, and indefinite descriptions. The variable K in the n1 and n2 predicates represents a local context for pronoun «binding» inside the sentence - the sentence-internal history.

The first task of the generator is to locate a binary relation in the representation list. The relation P consists of a relation symbol V, the referent X - the subject -, and the referent Y - the object. The two predicates n1 and n2 build the corresponding NPs with the aid of the unary relation symbols P1 and P2, describing the two referents. Then the three relations P, P1, and P2 are subtracted from the representation list, the sentence constructed is appended to the discourse to be generated, the global history grows by the three relations, and the local history is replaced by them.

Then the generator starts over again. The chief interest lies in the construction of the noun phrases.

3 *Familiar or Unfamiliar?*

When the subject NP predicate n1 and the object NP predicate n2 are to construct NPs in their N1 and N2 variable places to describe the parameters X and Y in the relation P, they run through four options: The NP is to be

- a pronoun
- a «the» or a «one» (semi-) definite description
- an indefinite «a» description
- an «another» description.

The first two options have in common that the referent is familiar: The history (a list of relations like the representation itself) contains a unary relation with the referent. This means that the same entity has been mentioned at least once before.

The pronoun requires the referent to be in the local history. The next option is only open if the first fails and requires the referent to be in the global history. The last two options only apply if the second option fails too - the referent must be novel. The «a» option requires the description of it to be novel as well, else the «another» option succeeds.

4 *Indefinite Descriptions: «A» and «another»*

I start the description of the particular options by describing the indefinites, then treat the definite or semi-definite descriptions and only at last turn to the pronouns. The distinction between the «a» and the «another» description deserves comment. It involves a strict condition on the indefinite article: The representation

[farmer(x),donkey(y),farmer(z),donkey(u),owns(x,y),owns(z,u)]

yields the text «A farmer owns a donkey. Another farmer owns another donkey.» In general, whenever a referent is unfamiliar and has a familiar noun description, only the «another» option succeeds. This implies an «omniscient» interpretation of the logical form: The case where a referent may or may not corefer with another never occurs (assuming that two referents with different descriptions do not corefer either). To be sure, this is an oversimplification. In reality, we can often interpret two occurrences of an indefinite in the same discourse in the sense that the two referents may but must not corefer. Thus the following is no worse than the above:

A farmer owns a donkey. Another farmer owns a donkey too.

These cases, where we are only interested in the type and not in the token, so to speak, are the contexts where «one» anaphora are used.

A farmer owns a donkey. Another farmer owns one too.

However, the alternative to the method I use is to only use the determiner «another» if a condition like $\neq(x,z)$ is stated in the representation, and this leads to texts like

? *A farmer owns a donkey. A farmer owns a horse (too).*

Note that a «one» anaphor is much more natural in object than in subject position.

? *A German sub has sunk a British ship. One has sunk a Norwegian ship too.*
A German sub has sunk a British ship. A Norwegian ship has been sunk by one too.

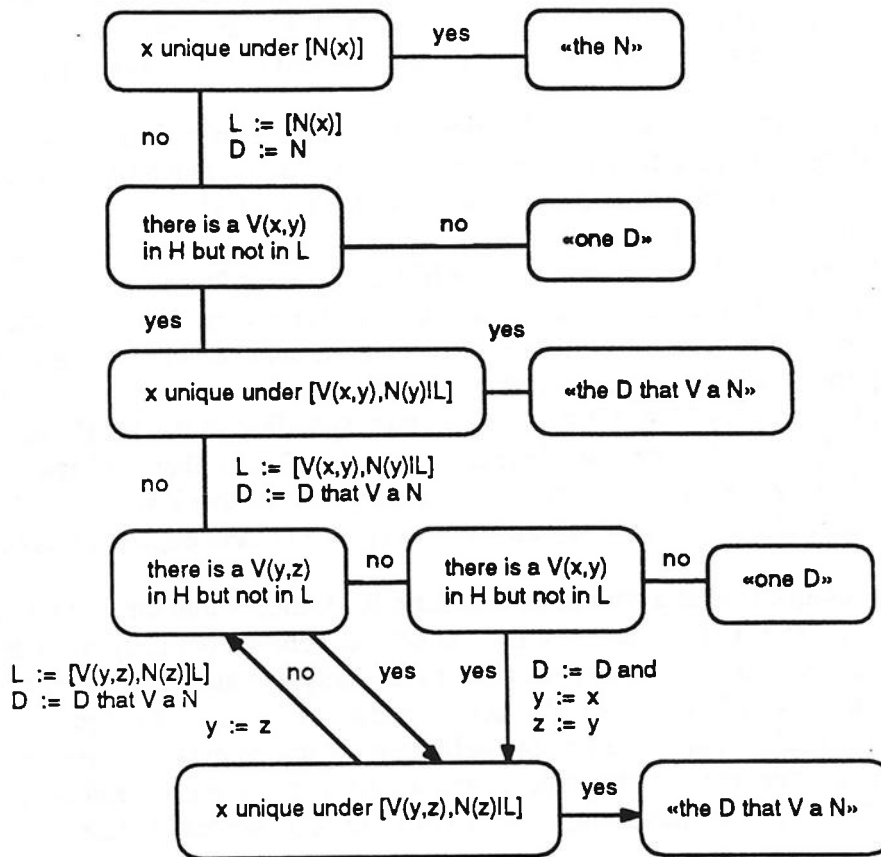
Surely, the felicity of one indefinite description in the neighbourhood of another depends on topic-focus phenomena. I shall not pursue the matter any further, only note that this question only arises when indefinite descriptions are generated and not just interpreted.

5 *Semi-Definite and Definite Descriptions*

If the parameter to be «covered» by the NP under construction is familiar, not from the local but from the global history of the discourse, the system tries to generate a definite description. A «the» description serves the purpose of identifying the parameter in cases where a pronoun would not succeed in identifying it. In comparison with a pronoun, the definite description adds information to guide the identification. The addition of a noun description may not suffice. The test is whether the familiar parameter is unique under the description: There is no other referent with the same description in the history. Information may have to be added in many stages, and the preliminary description is step by step tested for uniqueness, until all accessible information on the parameter is consumed. Should uniqueness fail ultimately, the definite determiner «the» yields to the semi-definite determiner «one».

The definiteness algorithm is the core of the system, composed from complicated components. While on the whole, sentences are constructed as terse as possible, and in particular, indefinite descriptions contain no relative clauses, noun phrases that signal familiarity and serve to identify parameters as closely as possible may have to contain relative clauses both coordinated and subordinated. In addition, they may have to contain pronouns, pronouns «bound» both outside and inside themselves, and they must provide suitable contexts for pronouns in the next NP.

The overall algorithm can be depicted in rough outline in the following way. Consider the parameter x found to be familiar not from the local but from the global history.



The overall (semi-)definite description algorithm.

Uniqueness under a list of formulae is a complicated notion that takes into account sloppy and strict identity. The subalgorithm is described in the next section.

Note that definite descriptions cannot contain other definite descriptions. However, they can contain pronouns, a fact which complicates the picture. First, pronouns can replace «a N» in the description, and second, if they are bound externally, they act as constants in the uniqueness algorithm described in the next section.

6 Pronouns in Descriptions I

When a description contains a pronoun, there are two distinct cases:

- (1) The pronoun refers inside the description, ie. it acts as a bound variable;
- (2) the pronoun refers outside the description, ie. it acts as a free variable.

(1) and (2) have different effects on the definiteness of the description. Specifically, in the uniqueness algorithm, free-variable pronouns cause the initial pair $\langle y_1, y_2 \rangle$ to be not anonymous but instantiated to one and the same referent in both places (see below). This section treats case (1), and the next section treats case (2).

In natural language, indefinite descriptions contained in definite descriptions are read «sloppily» in the following sense. For an x_1 to be unique under a description «N that V a N*», corresponding to the list $[N(x_1), V(x_1, y), N^*(y)]$, the condition is that there is no x_2 such that the history contains $N(x_2)$, $V(x_2, z)$, and $N^*(z)$ for a z ; it is not that there is no x_2 such that the history contains $N(x_2)$, $V(x_2, y)$ and $N^*(y)$. An alternative formulation of this is that the indefinite cannot outscope the definite. Moreover, if a definite description contains a pronoun «bound by» a referent of that description, that pronoun is interpreted sloppily too: For an x_1 to be unique under a description «N that V an N* that V* it», corresponding to the list $[N(x_1), V(x_1, y), N^*(y), V^*(y, x_1)]$, the condition is that there is no x_2 such that the history contains $N(x_2)$, $V(x_2, z)$, $N^*(z)$, and $V^*(z, x_2)$; not that there is no x_2 such that the history contains $N(x_2)$, $V(x_2, z)$, $N^*(z)$, and $V^*(z, x_1)$. The system captures this conduct.

In sum, uniqueness of a parameter x under a list L means that there is no parameter y different from x such that there is a list L^* that comes from L by replacing x by y and every occurrence of every parameter uniformly and such that every member of L^* is a member of the history. To accommodate this notion of uniqueness in terms of uniform substitution in a list, the definition of uniqueness involves four entities - the parameter, the list, the history, and a pair of parameters - and four recursive rules. The main rule makes use of an auxiliary relation we may notate $\approx$.

The parameter x is unique under the list L with respect to the history H iff there is no $y \neq x$ such that $x \approx y$ with respect to L , H , and the empty parameter pair $\langle _, _ \rangle$. Two parameters x_1 and x_2 stand in the approximate equality relation $\approx$ wrt. L , H , and an arbitrary pair $\langle y_1, y_2 \rangle$ iff for every formula P a member of L such that x_1 occurs in P there is a formula Q a member of H such that either

- (i) $x_1 = x_2$
- (ii) $P = R(x_1)$ and $Q = R(x_2)$
- (iii) $P = R(x_1, y_1)$, I , and $Q = R(x_2, y_2)$ and $y_1 = y_2$ wrt. $L \setminus P$, H , and $\langle x_1, x_2 \rangle$
- (iv) $P = R(x_1, z_1)$ and $Q = R(x_2, z_2)$ and $z_1 = z_2$ wrt. $L \setminus P$, H , and $\langle y_1, y_2 \rangle$
- (v) $P = R(y_1, x_1)$, I , and $Q = R(y_2, x_2)$ and $y_1 = y_2$ wrt. $L \setminus P$, H , and $\langle x_1, x_2 \rangle$
- (vi) $P = R(z_1, x_1)$ and $Q = R(z_2, x_2)$ and $z_1 = z_2$ wrt. $L \setminus P$, H , and $\langle y_1, y_2 \rangle$.

(The «cut» - ! - in option (iii) in this quasi-Prolog notation has the effect that if the option succeeds thus far and subsequently fails, the other options are not eligible.)

Consider a history

[farmer(x),farmer(y),donkey(z),donkey(u),owns(x,z),owns(y,u),kicks(z,x),
kicks(u,x)]

and a list

[farmer(x),owns(x,z),donkey(z),kicks(z,x)],

corresponding to the description «... farmer that owns a donkey that kicks him».

«The» is warranted, in spite of there being another farmer owning a donkey kicking «him» in the strict sense. The algorithm works as follows. x is compared to y , and $x \approx y$ only if for the formula $\text{owns}(x,z)$ there is a formula $\text{owns}(y,u)$ in the history and $z \approx u$ wrt. [farmer(x),donkey(z),kicks(z,x)], the history, and $\langle x,y \rangle$, which is the case only if for the formula $\text{kicks}(z,x)$ there is a formula $\text{kicks}(u,y)$ in the history (option (iii) again), which is not the case. Option (iii) fails, and option (iv) is not eligible, because of the cut.

It is important that this (non-)uniqueness algorithm in terms of the parameter, the alternative parameter, the property list, the history, and the anonymous parameter pair works in arbitrary depth - a pronoun and its antecedent may be arbitrarily deeply embedded in the description. Consider a more complex example, the history

[farmer(x),farmer(y),horse(z),horse(u),donkey(v),donkey(w),
owns(x,z),owns(y,u),loves(z,v),loves(u,w),kicks(v,x),kicks(w,x)]

and a list [farmer(x),owns(x,z),horse(z),loves(z,v),donkey(v),kicks(v,x)], corresponding to the description «farmer that owns a horse that loves a donkey that kicks him». Again, «the» is warranted, because for x to be $\approx y$, z must be $\approx u$ wrt. $\langle x,y \rangle$, and for this to be the case, for $\text{loves}(z,v)$ there must be a formula $\text{loves}(u,w)$ in the history (option (iv)) and v must be $\approx w$ wrt. $\langle x,y \rangle$, which cannot be the case for the same reason as above (option (iii) again).

The next section describes pronouns that refer strictly, ie. to antecedents outside the description.

7 Pronouns in Descriptions II

According to the last section, a list of formulae [farmer(x),owns(x,z),donkey(z),kicks(z,x)] wrt. x corresponds to a description «farmer that owns a donkey that kicks him» (and «the» is chosen as the determiner in case uniqueness holds, or else «one» is selected). The pronoun «him» is chosen for the third mention of x because the noun for the first mention provides an antecedent - in terms of logical form, the unary formulae in the list underlying the description add up to a local history K for the three noun-phrase-in-description rules:

$n3(_,H,K,F,X,P,NP,(X,X)) \quad :- \quad obpron(X,P,H,K,N), obpron(X,P,H,F,N), NP=(N).$
 $n3(_,H,K,_,X,P,NP(_,_)) \quad :- \quad obpron(X,P,H,K,N), NP=(N).$
 $n3(G,_,_,_,X,P,NP(_,_)) \quad :- \quad m(P,G), P=..(N,X), NP=(a,N).$

Superficially, there are two options: A pronoun or an indefinite description. However, there are two distinct pronoun cases - the case where the pronoun is «bound» inside the description, as in «the farmer that owns a donkey that kicks him», or in «the farmer that owns a donkey that kicks a farmer that beats it», etc., and the case where it is «bound» outside the description, as in «the farmer that feeds it», be it in the last sentence or in the subject NP description. The second rule takes care of the former case and the first rule takes care of the latter case.

Let me explain the various variables.

H is the recent history consisting of the formulae used for the last sentence. K is the particular recent history consisting of the formulae used for the description so far, - together with the formulae used for the subject. F is in the subject case the same as H but in the object case the particular recent history consisting of the formulae used for the subject. Totally three obpron goals are required because pronouns have two sides to them: First, their distribution itself, and second, their effect on uniqueness of the description containing them: If they are bound description-internally, they act as bound variables, subject to uniform substitution; if they are bound description-externally, they act as constants, subject to no substitution.

The distribution of pronouns in object position is governed by the following algorithm. A pronoun may be used to «cover» the parameter x wrt. the two histories H and K if x is the only parameter to occur in K or if x is the only parameter in H+K (under a noun of the specific gender). Consider now the NP «the farmer that owns it». The pronoun «it» is «bound» outside the NP, either in the last sentence or, in the event of an object NP, in the subject NP. This means that the first n3 rule above succeeds via the second obpron goal, and it has consequences for the definiteness of the large NP: The last argument position of n3 is then a pair of one and the same variable, and this is the one and only case where the initial pair of parameters for the uniqueness algorithm is not empty (anonymous). It effects a strict reading of pronouns referring out of the NP. Thus if we consider a global history

[farmer(x),farmer(y),donkey(z),donkey(u),owns(x,z),owns(y,u),...],

a list [farmer(x),owns(x,z),donkey(z)], possibly corresponding to the description «farmer that owns it», and the parameter pair (z,z), we see that «the» is again warranted: For x to be $\approx y$ with reference to the uniqueness algorithm in the last section, for the formula owns(x,z) for R(x1,y1) in option (iii) there must be a formula owns(y,z) for R(x2,y2) in the history, $\langle y1,y2 \rangle$ being set to $\langle z,z \rangle$. So x is unique under the list wrt. the excerpt of the history and the parameter pair.

8 Donkey Anaphora

Donkey sentences have been studied in depth, but mostly, the determiner of the NP containing the antecedent has been «every», «most», etc., not «(one of) the». The semantics of the sentence «the farmer that owns a donkey beats it» is unproblematic as long as the farmer that owns at least one donkey owns just one. As with current analyses of donkey sentences with other determiners, summed up by Heim (1990), however, difficult questions arise when a farmer owns two or more donkeys. It is clear that if the farmer that owns a donkey does in fact own at least two, the donkey pronoun cannot be interpreted as «every donkey he owns». The interpretation in the system is that he beats at least one of his donkeys - the «weak reading» advocated by Chierchia (1991). The program produces eg.:

The farmer that owns a donkey beats it.

The man that loves a woman that owns a dog kicks it.

The man that loves a woman beats the donkey that kicks her.

The system functions as follows. The context for the pronoun is the list of formulae used for generating the subject description. If a farmer owns two donkeys *x* and *y*, this list will contain either *donkey(x)* or *donkey(y)*, depending on their order in the history. Thus the system will produce a pronoun *or* a (semi-)definite description. It is unsatisfactory to have one and the same state of affairs in essentially the same context described in two different ways depending on the generational history. But if we accept the «weak reading» of the pronoun, there is no harm in it. The problem arises if we require «it» to mean essentially the same, not as «one donkey he owns» but as «the donkey he owns». If the subject noun phrase is held responsible, a solution based on the Kamp (and early Heim) analysis of «every» cases suggests itself: The determiner affects every referent in the description, so that uniqueness requires there to be a unique tuple of referents. For an *x* to be described as «the farmer that owns a donkey», there must be a unique pair (*x*,*y*) such that [*farmer(x)*, *donkey(y)*, *owns(x,y)*]. However, the alternative to describing the farmer owning two donkeys as «the farmer that owns a donkey» - the semi-definite description «one farmer that owns a donkey» - is clearly misleading: It presupposes that there is another farmer that owns a donkey. In other words, the «the» determiner may like «every» be symmetric in the sense that pairs or tuples are what count, but the «one» determiner is like «most» asymmetric.

Definite descriptions can occur in «donkey pronoun» position too (Cooper 1991):

The farmer that owns a donkey and that owns a horse beats the donkey.

This is a case where the definite description is sensitive to the local history built up by the subject noun phrase, and it is accommodated in the system.

Towards a Definition of Adequacy

An adequate definition of adequacy must not require complete identification but only that the referent be distinguished from all from which it is distinguishable. Dale's original formulation (1988: 161) is

«the NP is sufficient to identify the intended referent».

Which is too strong; a referent may share its history with others, and still the noun phrase must count as adequate if only it makes it possible to restrict it to that class. This requirement is the motivation behind what I term semi-definite descriptions, indicating familiarity by means of the determiner «one (of the)». To account for cases where it is impossible to identify the referent because it is indistinguishable from some other referent, we reformulate the informal definition as

«the NP is sufficient to distinguish the intended referent from every other referent from which it is distinguishable».

Formally, the Principle of Adequacy must be defined in terms of the NP itself, the «intended referent», and the history. On the basis of the generation algorithm, it is possible to define a partial function mapping a noun phrase, a discourse referent, and a pair of histories onto a list of formulae. The function will be undefined if a pronoun has no proper antecedent in the local history or if a description cannot be matched with corresponding relations in the global history.

An NP is adequate wrt. the referent x and the two histories H and G iff it corresponds to a list L and L is such that

there is no referent $y \neq x$ such that

there is a list L^* that comes from L by replacing x by y
and every occurrence of every other referent uniformly
such that every member of L^* is a member of H

and such that

there is no list H^* that comes from H by replacing x by y
and every occurrence of every other referent uniformly
such that every member of H^* is a member of H .

In ordinary words: There is no other referent from which the description does not distinguish the referent and from which it is distinguishable; for any other referent: If x is indistinguishable from it on the description, then x is indistinguishable from it on the complete history too.

Note that if we cut the definition off after the first clause, it expresses the notion of identification (uniqueness, definiteness) implicit in Dale's original formulation.

Pronouns that act as free variables are rendered as underscored referents, referents marked as irreplaceable in L. This is akin to a method used to a similar purpose by Dalrymple et al. (1991). Thus top level pronouns are trivially adequate, since they actually identify the intended referent even if it is indistinguishable from another. Conversely, an indefinite description will correspond to an L headed by a fresh referent and thus - on a refined definition - be trivially inadequate.

Asymmetric and Symmetric Adequacy

There is an interesting distinction between symmetrical and asymmetrical adequacy, where symmetrical adequacy relies on negative information. There are cases where the information on the referent in question is properly included in the information on another referent. Suppose the history has two farmers *x* and *y* and two donkeys *z* and *u* such that *x* owns *z* and *y* owns *u* and in addition, *y* beats *u*, and suppose the formula to be generated is *feeds(x,z)* - if *x* or *z* cannot be covered by a pronoun and if nothing more is known about *x* or *z* than mentioned, *x* must be covered by «one farmer that owns a donkey», which is intuitively inadequate since it describes *x* or *y*. The general situation is this. There are two referents *x* and *y* such that what is known about *x* is a proper subset of what is known about *y*, so that *x* is not distinguishable from *x* but *y* is distinguishable from *x*: There is a list *L* a sublist of the history such that not *y* = *x* wrt. *L* but there is no list *L** a sublist of the history such that not *x* = *y* wrt. *L**, with reference to the uniqueness algorithm in section 6. In a situation like this, some speakers say (1) while others say (2).

- (1) *One farmer that owns a donkey - but not the one that beats it - feeds it.*
- (2) *The farmer that owns a donkey but that does not beat it feeds it.*

(2) corresponds to a closed world assumption, while (1) reflects an open world assumption. To necessitate such an option, we must make the definition symmetric, requiring not only that *x* be indistinguishable from *y* on the complete history but that *y* be indistinguishable from *x* as well. This means strengthening the definition.

Adequacy does not Apply to NPs but to NP Tuples

The availability of pronouns is likely to enhance overall adequacy. There are two ways for a pronoun to support adequacy: As a noun phrase in its own right, ie. at top level, replacing a semi-definite description, or at a lower level, as a free-variable or a bound-variable pronoun, replacing an indefinite description in a semi-definite description and rendering the description adequate, maybe definite. However, it is interesting to note that certain cases where intuitively, a pronoun is necessary for adequacy, come out as adequate without the pronoun on the definition as it stands.

Consider these two structures:

[farmer(x),farmer(y),donkey(z),donkey(u),owns(x,z),owns(y,u),kicks(u,y)].
[farmer(x),farmer(y),donkey(z),donkey(u),owns(x,z),owns(y,u),kicks(z,x)].

With pronouns, the upper structure corresponds to the discourse «A farmer owns a donkey. Another farmer owns another donkey. It kicks him.», and the lower structure corresponds to the discourse «A farmer owns a donkey. Another farmer owns another donkey. One donkey kicks the farmer that owns it.». Without pronouns, the last sentence would in both cases have to be «One donkey (that a farmer owns) kicks one farmer (that owns a donkey)».

But on the definition of adequacy as it stands, this last sentence is not inadequate in either noun phrase. As long as the adequacy criterion is limited to one NP at a time, those pronouns cannot be shown to be adequacy-necessary. Which means that the definition of adequacy is inadequate; it is not strong enough. It misses cases where pronouns create or reflect a dependency between the two referents in a formula.

- (1) *one donkey kicks one farmer that owns a donkey*
- (2) *one donkey kicks the farmer that owns it*
- (3) *it kicks him*

Intuitively, only (2) or (3) is adequate, but according to the definition as it stands, (1) is adequate too. The point is, of course, that it does not matter which donkey or which farmer you select, but it does matter which farmer you select once you have selected a donkey. By itself, either referent (the subject or the object referent) is indeed indistinguishable, but the pair of referents is distinguishable. This shows that it is impossible to define adequacy with reference to the NP in isolation. Adequacy can only be satisfactorily defined in terms of pairs (or triples etc) of NPs and pairs of parameters, corresponding to pairs of lists. The specifics of such a modification remain to be worked out.

Definites versus Indefinites in Descriptions

Interestingly, there seems to be no adequacy difference between a definite, a semi-definite, and an indefinite description embedded in a (semi-)definite description. As long as definite descriptions are interpreted with respect to the global history, there seems to be no adequacy benefit in embedding definite descriptions. It is just in case it refers locally that an embedded definite description may be necessary for adequacy.

The farmer that owns a donkey and a horse and that beats the donkey...

Even more interestingly, indefinite descriptions inside (semi-)definite descriptions do not, as usual, signal novelty; it is equally felicitous to use «a» as «the» or «one» in such contexts, where on account of the uppermost determiner every piece of information is presupposed as familiar anyhow. The novelty condition for indefinite descriptions (Heim 1982: 369) is invalid under (semi-)definite descriptions. (It is another matter that «the» is strongly preferred to signal functionality with relations like «owns», cf «the donkey that kicks ?a / the farmer that owns it».)

Furthermore, definite descriptions in definite descriptions are not governed by the normal rule either. The uniqueness condition for definite descriptions is weakened under definite descriptions. A definite description inside another definite description does not require uniqueness on itself but only on the uppermost description.

This point is made by Dale and Haddock (1991: 166): Suppose we have a history

[farmer(x),farmer(y),donkey(z),donkey(u),beats(x,z),...].

Dale and Haddock point out that it is felicitous to refer to x in a context like this as «the farmer that beats the donkey», although there is another donkey in the history. This poses a problem for compositional interpretation, on which the embedded NP would refer uniquely in isolation. The uniqueness of the referent of an embedded definite description must be relativised to the list underlying the head description, in this case, [donkey(z),beats(x,z),farmer(x)].

Conclusions

Let me sum up the main points of the second part of this paper:

- (1) A definition of adequacy must not require complete identification but only that the referent be distinguished from all from which it is distinguishable.
- (2) The distinction between symmetrical and asymmetrical adequacy deserves attention.
- (3) A definition of adequacy must require that the tuple of referents at top level in a sentence be distinguished from the tuples from which it is distinguishable; adequacy cannot be defined in terms of one NP alone.
- (4) The distinction between a definite and an indefinite description is irrelevant to adequacy and partly also to felicity when the description is embedded in a definite description.

References*

- Appelt, D.: 1985, Planning English Referring Expressions, *Artificial Intelligence* 26, 1-33
- Chierchia, G.: 1991, Anaphora and Dynamic Logic, in Stokhof et al. (eds.), *Quantification and Anaphora I* (= DYANA deliverable R2.2A).
- Cooper, R.: 1991, Donkeys and Situations, in Stokhof et al. (eds.), *Quantification and Anaphora I* (= DYANA deliverable R2.2A).
- Dale, R.: 1988, *Generating Referring Expressions in a Domain of Objects and Processes*, University of Edinburgh Dissertation.
- Dale, R. and N. Haddock: 1991, Generating Referring Expressions Involving Relations, in *Proceedings of the 5th Conference of the European Chapter of the ACL*, Berlin, 161-166.
- Dalrymple, M., F. Pereira, and S. Shieber: 1991, Ellipsis and Higher-Order Unification, *Linguistics and Philosophy* 14, 399-452.
- Heim, I.: 1982, *The Semantics of Definite and Indefinite Noun Phrases*, University of Massachusetts at Amherst Dissertation.
- Heim, I.: 1990, E-Type Pronouns and Donkey Anaphora, *Linguistics and Philosophy* 13, 137-177.
- Johnson, M. and M. Kay: 1990, Semantic Abstraction and Anaphora, in *COLING 1990*, Helsinki, 17-26.
- Johnson, M. and E. Klein: 1986, Discourse, Anaphora, and Parsing, in *COLING 1986*, Bonn, 669-675.
- Kamp, H.: 1981, A Theory of Truth and Semantic Representation, in Groenendijk et al. (eds.), *Formal Methods in the Study of Language*, Amsterdam, 277-321.
- Novak, H.: 1988, Generating Referring Phrases in a Dynamic Environment, in Zock and Sabah (eds.), *Advances in Natural Language Generation*, Volume 2, London, 76-85.
- Reiter, E.: 1990, *Generating Appropriate Natural Language Object Descriptions*, Harvard University Dissertation.

* I am indebted to Helle Frisak Sem and Kjetil Strand for insightful comments and valuable suggestions.

0. INTRODUCTION

In the semantics of plurals a distinction is recognized between distributive and non-distributive understandings or readings. For example, (1b), is true distributively if John lifted three cartons and Mary did too. On the other hand, it is true non-distributively, if for example John lifted two cartons and Mary lifted one carton. If a speaker wishes to clarify that he is interested in the distributive understanding of (1b), he may use the distributive marker *each* as in (1c), while if it is the non-distributive understanding he is interested in, he may use the non-distributive marker *together* as in (1a).

- (1) a. [John and Mary together] lifted 3 cartons.
- b. John and Mary lifted 3 cartons.
- c. John and Mary each lifted 3 cartons.

In this paper I will be concerned proposals for handling the non-distributive marker *together*. After a few remarks about the syntax of *together*, I will turn to the semantics attempting a formal characterization of the notion of non-distributivity.

A few terminological remarks are in order before turning to the body of the paper. Above, I used the term "non-distributive". In fact, non-distributive understandings are often characterized as either "collective" or "cumulative". The situation I gave in which (1b) would be true non-distributively, namely one in which John lifted some of the three cartons and Mary lifted the others, would involve us in a cumulative understanding of (1b). (1b) would be true collectively if each of the cartons was lifted by the group consisting of John and Mary. This further subdivision will not be important here so I will stick with "non-distributive". Next, I have used this funny term "understanding" where you might have expected "reading". This is because the term "reading" is generally associated with ambiguity and I want, at least at the outset, to leave open the question of whether the distributive/non-distributive distinction is one of ambiguity or one of vagueness. Also for this reason, I talk of sentences being 'true (non-)distributively' rather than 'true on a (non-)distributive reading'. Finally, let me clarify that I use the term "individual" in the ordinary language sense, applying to single people or things and never to pluralities such as the books on my shelf or the people at the next table. Unless I specifically say otherwise, I assume throughout a semantics in which pluralities are modeled as objects corresponding to sets of individuals and called "groups", not to be confused with the groups of Link(1984) or Landman(1989). I will represent the group

composed of a and b as: $a*b$.

1. SYNTAX

The bracketing in (1a), represents a syntactic analysis of *together* in which it functions as a noun phrase post-modifier. Evidence that *together* indeed combines with a noun phrase and doesn't just happen to be sitting next to one, is the fact that *together* noun phrases occur almost anywhere noun phrases occur. In (2a) the object is a *together* noun phrase and in (2b) the *together* noun phrase is in the complement of *price*.

- (2) a. He outweighs John and Mary together.
- b. What surprised him was the price of the two books together.

A purely adverbial analysis of *together* would not account for this last example. This is not to say that *together* does not occur adverbially. In fact, I would claim that a pair such as (3a-b) differ on just this question. In (a) *together* is an ad-NP and in (b) it is an adverb or ad-VP.

- (3) a. The U.S. and Saudi Arabia together control the world's oil.
- b. The U.S. and Saudi Arabia control the world's oil together.

Notice furthermore that the two are not synonymous. (3b) entails some collaboration between the two oil producers whereas (3a) does not.

Having given some justification for treating ad-NP and adverbial *together* separately, I now turn to the semantics of the former. Throughout the remainder of this paper, references to *together* should be taken as referring exclusively to the ad-NP variant.

3. SEMANTICS for ad-NP *together*: PART I

We turn now to the semantics of *together* which I have described as a non-distributive marker. This is because a noun phrase ending in *together* is such that any verb phrase that combines with it is interpreted non-distributively. This means that in working out the meaning of *together* we need to come up with a way to characterize non-distributivity formally and include this characterization in the meaning of *together*. To get an idea of what is needed here, it is useful to look at what has been done with the distributive marker *each*, even though the syntax of these two markers differs. I have sketched, by way of example (4) below, an analysis of *each* based roughly on the modification in Link(1987) and Roberts(1987) of Dowty & Brodie(1984) and couched in a semantics which models pluralities as sets.

- (4) a. John and Mary each lifted 3 cartons.

- b. $[\lambda x[\lambda y((y \in x \ \& \ IN(y)) \rightarrow L(y))]](John'+Mary')$
 c. $\lambda y((y \in John'+Mary' \ \& \ IN(y)) \rightarrow L(y))$

' ϵ ' in the translation in b. is interpreted as set-membership, while the predicate 'IN' in b. is true only of individuals and false of pluralities. After lambda conversion, we get (4c) as the translation for (4a). (4c) says that the predicate *lift 3 cartons* must hold of every individual member of the group consisting of John and Mary. In other words, both John and Mary have to have lifted three cartons for (4a) to be true and indeed this is the distributive understanding of the simple *each*-less sentence. Looking at (4b) again, we can say that *each* relates a group to a predicate, requiring that all the individual members of that group be in the extension of the predicate. Having characterized *each* this way, we now can ask what requirement should be imposed on a predicate to capture the contribution of *together*?

The first possibility we consider is based on a suggestion of Hoeksema(1983). Since the distributive marker requires individuals in the extension of the predicate, why not have the non-distributive marker require the group itself to be in the extension of the predicate. In other words, whereas *each* in (1c) requires that the individual members of the group of John and Mary be in the extension of the main predicate, the effect of adding *together* in (1a), is to require that the group of John and Mary be in the extension of the predicate *lift 3 cartons*. The difficulty facing this proposal, as pointed out in Lasnik(1990), is that given certain standard assumptions concerning the meaning of verb phrase conjunction, it turns out that if a predicate is true of a group, that group will be in the extension of that predicate regardless of whether the predication is understood distributively or not. To see this consider the following example:

- (5) These cars were put together in Malaysia and sent to different countries in Europe.

Here I am interested in a construal of the sentence in which the first verb phrase conjunct is understood distributively while the second is understood non-distributively. In other words, the cars were not attached to one another, but rather each car was assembled in Malaysia and *go to different countries* is true of the group; each car went to just one country. Now, if we assume that the extension of a conjoined verb phrase is just the intersection of the extensions of the conjuncts, then if the group of cars is in the extension of the second verb phrase, because it is understood non-distributively, it must also be in the extension of the first verb phrase. But this means that having a group in its extension does not mark a predicate as non-distributive.'

If we can't distinguish a non-distributive predicate by its having a group in its extension, perhaps we can distinguish it from a distributive predicate by not having individuals in its extension.

The idea here would be to say that in fact predicates are truly ambiguous, and the non-distributive variant has no individuals in its extension. Then the role of *together* would be to impose the requirement that the incoming predicate, the one the *together*-NP combines with, have no individuals in its extension and this would disambiguate towards the non-distributive reading. Unfortunately, both Hoeksema(1983) and after him Lasersohn(1988) have given arguments that can be used to show that non-distributive predicates in fact do have individuals in their extension. Their arguments are based on examples of the type in (6a,b) below.

- (6) a. Only John and Mary lifted a carton.
b. Nobody lifted a carton.

Consider a situation in which there were exactly two carton liftings: one by Bill and the other one by John and Mary. (6) is intuitively false in this situation, however consider what happens if we give L, the translation for *lifted a carton*, the proposed groups-only, non-distributive, interpretation in the translation of (6a) given in (7):

- (7) $\neg \lambda x [L(x) \rightarrow x = \text{John} + \text{Mary}]$

Since Bill is not a group, $L(\text{Bill})$ is false, and since nobody else lifted any cartons, (7) comes out true. Similarly, (6b) is intuitively false in situation in which Bill and only Bill lifted a carton. However, again, if there was a groups-only reading of the predicate in (6b), then it would yield a true reading of the sentence in that situation. Hoeksema and Lasersohn's examples involve monotonically decreasing quantifiers, however problems crop up elsewhere as the following example should show:

- (8) Many customers withdrew money on black Friday and so the bank collapsed.

Imagine a situation in which many in fact almost all of the customers withdrew money from their accounts causing the bank to collapse. In such a situation, (8) appears true, plain and simple. But now consider that the majority of the withdrawals were from individual accounts. While this doesn't appear to affect the truth of (8), if we allow for a groups-only non-distributive interpretation for *withdrew money* then (8) will come out false, since there weren't many groups that withdrew money nor were there many individuals that are members of groups that withdrew money, as most of the customers withdrew money as individuals.

In addition to the problems associated with examples (6) and (8), there is the unpleasant fact that if we posit an ambiguity in terms of having or not having individuals in a predicate's extension, we end up with 2ⁿ meanings for an n-place predicate of English. Thus (9) has four readings depending on whether there is distributivity with the subject or object or both or neither:

(9) John and Mary lifted the two boxes together.

As Bennett(1974:233) points out, the ambiguity explosion is even greater once we start to consider categories that take predicates as arguments.

Reviewing now, it turns out that while the extension of a distributively understood predicate is identifiable in terms of whether certain individuals are in there or not, the extension of a non-distributively understood predicate is not so easy to identify. On the one hand, it is not enough to say that it has groups in it, since distributive predicates do as well. On the other hand, we cannot say that it fails to have individuals in it, since that idea leads to incorrect readings of quantificational statements and to massive inflation of syntactic categories.

There is a certain inequity here which we might be able to use to our advantage. While intuitively, the notion of a distributive construal certainly doesn't seem to be simpler than the notion of a non-distributive construal, the former appears much easier to capture than the latter. The imbalance is even more unfair when you consider that the two notions are so obviously related. Rather than lament this relationship, we should exploit it. For if we can relate non-distributivity to distributivity and we can capture the latter in our formal system, then all we need to do to formalize non-distributivity is capture the relationship itself. Consider the triplet in (1) again:

- (1) a. John and Mary together lifted 3 cartons.
- b. John and Mary lifted 3 cartons.
- c. John and Mary each lifted 3 cartons.

Notice that if c. is false and b. is true, then a. is true. This leads then to (10a) as a translation for (1a):

- (10) a. $L(\text{John}' + \text{Mary}') \ \& \ \text{not}([EACH(L)](\text{John}' + \text{Mary}'))$
- b. $EACH = \lambda P \lambda x [\lambda y [(IN(y) \ \& \ y \in x) \rightarrow P(y)]]$

Here I use "EACH" as an abbreviation for the meaning given to *each* in (4) above. In effect, the contribution of *together* is to say that the sentence itself (without *together*) is true but not distributively.

This proposal encounters problems simply because while it is true that if a sentence with a plural subject is true but not distributively, then it is true non-distributively, the reverse is not the case. A sentence may be true on both construals. Consider a situation in which John lifted three cartons and then Mary lifted three cartons and then the two of them got together and jointly lifted three cartons. In this case, while (1a) is true, the translation in (10) is not.

Lasnik(1990) presents a more sophisticated version of this proposal which does not at first appear susceptible to the problem

I've raised. The crucial ingredient in Lasersohn's semantics is events. For while (1b) could be true distributively and non-distributively, as in the situation just described, we can envision a subdivision of that situation into minimal situations or events in which (1b) is true only distributively or only non-distributively. If the meaning for *together* in (1a) requires that there be a minimal event in which (10) above holds, the problem mentioned here will not arise.

While the introduction of events into the meaning for *together* does solve the problem for the example in (1) and for the examples that Lasersohn considers, there are a class of examples for which it fails, given certain standard assumptions concerning the structure of events. Examples of this type, such as those in (11) below, involve verb phrases that are in a certain sense downward entailing: if they apply to a group they necessarily to apply members of that group.

- (11) a. John and Mary lifted less than 3 cartons.
- b. The ax and the box are light enough to carry.

If a sentence whose main predicate is downward entailing is true non-distributively then it must also be true distributively. Assuming that if a proposition is true in an event then so are its entailments, there can not be an event in which (11a) or (11b) is true non-distributively without also being true distributively. In this case, the analysis implemented in (10) fails, even if it is relativized to minimal events.

Reviewing then, we began this last foray with the idea that if the relationship between the distributive and the non-distributive was elucidated correctly, one could get an analysis for the non-distributive by piggybacking on the success of analyses of the distributive. Our proposal included the mistaken assumption that the non-distributive is true only if the distributive is false. This solution dissolved in the face of sentences for which the non-distributive entails the distributive.

The initial idea of connecting the distributive and the non-distributive cannot be wrong; it is just too intuitive. Something must have gone wrong in the implementation then. I believe the problem lies in equating the non-distributive with the unconditional negation of the distributive. To overcome this problem, I will make a proposal in which predicate negation plays a significant role. We begin then with the observation that predicates and their negations both have distributive and non-distributive construals.⁴ Compare (12a) and (12b):

- (12) a. John and Mary are heavy.
- b. John and Mary are not light.

(12a) could be true distributively if John is heavy and if Mary is heavy. On the other hand, it may be that John is not heavy, though

he and Mary together are in which case (12a) is true non-distributively. Exactly the same set of circumstances and construals are possible for (12b) which involves the negated predicate *be not light*. (13) presents another example containing a negated predicate, *did not earn over \$5,000*.

(13) John and Mary did not earn over \$5,000 last year, so they will not have to pay taxes.

(13) may be true non-distributively, if the group consisting of John and Mary earned less than \$5,000 and this may indeed be relevant if they are filing a joint tax return. On the other hand, the group may have earned over \$5,000 though neither John nor Mary alone made over \$3,000 in which case (13) is true distributively. Given that both predicates and their negations have both construals, we can distinguish four logically possible situations, for a random predicate F , a group g , and predicate negation ' $\sim$ '. These four situations are presented in (14). Note, these are logically possible situations for a random predicate, though some of the situations presented maybe ruled out for a given predicate:

(14) <u>$F(g)$ is true:</u>	<u>$\sim F(g)$ is true:</u>
a. non-distributive	distributive
b. non-distrib. & distrib.	---
c. distributive	non-distributive
d. ---	non-distrib. & distrib.

Staring at lines a. and b. you might have noticed that a situation in which $F(g)$ is true non-distributively can be characterized in a roundabout way as one in which $F(g)$ is true and if $\sim F(g)$ is true it is true distributively. This observation leads to the following recipe for translating a sentence in which the subject is post-modified by *together*:

(15) if NP translates as g and VP translates as F then:

[NP together] VP translates as $F(g) \ \& \ [\sim F(g) \rightarrow \text{EACH}(\sim F)(g)]$

where $\text{EACH}(\sim F)$ abbreviates: $\forall y[(\text{IN}(y) \ \& \ y \in x) \rightarrow \sim F(y)]$

To see how this will work, we focus on the example in (16a) and its translation in (16b), arrived at by following the method of (15), letting L be the translation for *weigh less than 3 kilo*:

- (16) a. Alex and Gila together weigh less than 3 kilo.
 b. $L(\text{Alex}' + \text{Gila}') \ \& \ [\sim L(\text{Alex}' + \text{Gila}') \rightarrow \text{EACH}(\sim L)(\text{Alex}' + \text{Gila}')]]$
 c. $L(\text{Alex}' + \text{Gila}') \ \& \ [\sim L(\text{Alex}' + \text{Gila}') \rightarrow [\sim L(\text{Alex}') \ \& \ \sim L(\text{Gila}')]]$

If Alex in fact weighs one kilo and Gila also weighs a kilo then (16a)

is true. What of the translation? (16b) is equivalent to (16c) and we will focus on that formula. I will assume at this point that the distributive/non-distributive distinction is one of vagueness. Later on, I will discuss how to lift this assumption leaving the analysis intact. Proceeding then, $L(\text{Alex}'+\text{Gila})$ is true, since Alex and Gila weigh less than 3 kilo, on either construal. $\sim L(\text{Alex}'+\text{Gila}')$ is false since again Alex and Gila do not weigh less than 3 kilo is false, again on either construal. It follows then that (16c) is true, in line with (16a).

If Alex weighs two kilo and Gila weighs two kilo then (16a) is false. In this case, $\sim L(\text{Alex}'+\text{Gila}')$ is true, non-distributively, however $\sim L(\text{Alex}')$ is false and so is $\sim L(\text{Gila}')$, so (16c) is false, again in line with (16a).

Up to now, we have considered the two possible ways in which Alex and Gila could be said to weigh less than three kilo. Of course in a situation in which one of them weighs three kilo or more, the simple *together*-less sentence will be false, and so will (16a) and (16c).

While this method for analyzing *together* sentences seems to capture our intuitions, we need to be careful about how to interpret the translations. In particular, predicate negation cannot be interpreted in the standard way such that the following holds:

$$(17) \quad \neg F(g) \iff \neg [F(g)]$$

To see this just consider this formula from (15) again:

$$(18) \quad F(g) \ \& \ [\sim F(g) \rightarrow \text{EACH}(\sim F)](g)$$

It follows from (17) that the formula in (18), our translation for sentences with *together*, is equivalent to $F(g)$. Whenever $F(g)$ is true, $\sim F(g)$ must be false hence the conditional must be true hence the whole formula is true. If $F(g)$ is false, then the formula is false. In other words, *together* adds nothing to the meaning of the sentence. Looking at the chart in (14), you see the same thing. The only possible way of interpreting lines a. and c. of the chart in a system that included (17) would be to assume that F corresponds to different readings of an ambiguous predicate of English in different columns of a. and c. However, if we take that approach then we cannot use (18) to capture lines a. and b. as we want to, for if we did, the first instance of F in (18) would have to be a different reading of the VP than the second instance and it is not clear what kind of translation procedure would give you that. Summarizing then, we want a semantics that allows $F(g)$ and $\sim F(g)$ to be simultaneously true (on different construals) hence the system should not entail (17) and distributivity should be a matter of vagueness and not ambiguity. In fact, the latter requirement may not be necessary and later on I will consider a way of reintroducing ambiguity into the system for those who want it.

Meeting the above demands, I assume a semantics in which

functional expressions (of types $\langle a, b \rangle$ for any type a, b) are assigned both positive and negative extensions.³ At this point, I will outline the non-standard aspects of the semantics. Later on, I will add some of the remaining rules as the need arises.

First, meanings are assigned to basic expressions according to the constraints in (19). I have left off relativization to worlds or other indices as they are not relevant here. And I have left off relativization to the model and the assignment function for typographical reasons.

(19) Constraints on the assignment of meanings to basic expressions α of type $\langle e, t \rangle$,

Positive denotations:

- a. for any individual u in the domain of $\|\alpha\|$, $\|\alpha\|.(u) = 1$ if u has the relevant property; otherwise $\|\alpha\|.(u) = 0$.
- b. for any group u in the domain of $\|\alpha\|$, $\|\alpha\|.(u) = 1$ if the group itself has the relevant property otherwise $\|\alpha\|.(u) = 0$.

Negative denotations:

For all u , $\|\alpha\|.(u) = 1$ iff $\|\alpha\|.(u) = 0$
 For all u , $\|\alpha\|.(u) = 0$ iff $\|\alpha\|.(u) = 1$

The intent of (19a) is that, for example, *weigh 10 lbs* will be a function which applies truthfully to any individual who weighs 10 lbs and falsely to any other individual having mass. The intent of (19b) is that if a group, qua group, weighs 10lbs then the denotation of *weigh 10 lbs* will apply truthfully to that group, otherwise, the function will yield the value 0 for that group (if the group has mass). The phenomenon of distributivity is not taken into account at this stage.

Next we introduce predicate negation:

(20) Predicate negation:

If α is of type $\langle e, t \rangle$, then $\sim\alpha$ is a well-formed expression of type $\langle e, t \rangle$ where $\|\sim\alpha\| = \|\alpha\|$.

It follows from (20) and the way negative and positive denotations were connected in (19) that for any u in the universe and any predicate P :

$\|\sim P\|.(u) = 1$ iff $\|P\|.(u) = 0$ and $\|\sim P\|.(u) = 0$ iff $\|P\|.(u) = 1$

Finally, in (21) a rule for interpreting predicate application is introduced. I have used a subscripted $\#$ to range over $+$ and $-$, where $\#$ gets the same value throughout the rule. Since e type expressions are not assigned a pair of extensions, they are not

subscripted.

(21) Predicate application:

If α is of type $\langle e, t \rangle$ and β is of type $\langle e \rangle$, $\alpha(\beta)$ is of type $\langle t \rangle$
where:

$$\|\alpha(\beta)\|_i = 1 \text{ iff } \|\beta\| \in \text{Grp-C1}(\{u: \|\alpha\|_i(u) = 1\}).$$

$\text{Grp-C1}(K)$ is the closure under group formation of the set K , that is the set containing all the individuals and groups in K along with all groups formed from members of K .

The phenomenon of distributivity enters in at this point. This rule is in a class with proposals such as Roberts(to appear) which view distributivity as a 'mode of predication'. Though I am not proposing a special rule for distributive readings, rather there is one rule for interpreting (one-place) predicate application for both distributive and non-distributive construals. To get a feel for how these rules work, we begin with an example without *together*:

- (22) a. John and Mary weigh less than 100 kilo.
b. $W(\text{John}'+\text{Mary}')$

Consider first a situation in which John weighs 65 kilo and so does Mary. The function (positively) denoted by W , the translation for *weigh less than 100 kilo*, will apply to John to give 1, to Mary to give 1 and to the group of John and Mary to give 0. The closure under group formation of the things for which $\|W\|$ yields the value 1, includes then John, Mary and the group of John and Mary and so the translation in (22b) will come out true and correctly so since (22a) is true. In a situation in which John weighed 45 and so did Mary then $\|W\|$ would apply truthfully to the group, hence the group would be in the closure under group formation of the things for which $\|W\|$ yields the value 1, and again the translation in (22b) will come out true and correctly so, since (22a) is true in this case.

Before continuing on to a demonstration of the workings of this system in the analysis of *together*, I would like to say some words in defense of rule (21). This rule does violence to the traditional Fregean view that predicate application is interpreted as function application. It seems to me that that tradition relies on the intuition that for instance, to say "John is happy" is to say that John is among the set of things each of which is happy, what we call the extension of *happy*. Similarly to say that "John and Mary are a happy couple" is to say that John and Mary, the group, is among the set of things each of which is a happy couple. However, this way of thinking simply does not carryover straightforwardly to an example such as "John and Mary weigh 10 lbs". If John weighs 10 lbs and Mary weighs 10 lbs, then that group is not in the set of things each of which weighs 10 lbs. I see three ways to handle this

problem: a) Claim that the noun phrase *John and Mary* isn't really a referential expression but rather has a quantificational meaning something like 'everything that is either John or Mary'. b) Deny, as most current accounts do, that the extension of *weigh 10 lbs* is the set of things each of which weighs 10lbs. Rather its extension includes sets of such things as well. c) Admit that the relationship between the denotation of a term and the extension of the predicate applied to it, is not simply set inclusion, even though in the singular case, it reduces to that. It seems to me that, a priori, any of these are reasonable departures from the standard view, although arguments have been given against the first approach. What remains is to show that the move in c) is not only possible but also desirable.

Returning to the example in (16) repeated below, we reconsider the two situations in which sentence a. without *together* could be true.

(16) a. Alex and Gila together weigh less than 3 kilo.

b. $L(\text{Alex}' + \text{Gila}') \ \& \ [\sim L(\text{Alex}' + \text{Gila}') \rightarrow [\sim L(\text{Alex}') \ \& \ \sim L(\text{Gila}')]]$

First, in situation 1, Alex weighs one kilo and so does Gila. In this case, (16) is true. The results of invoking the semantic rules and constraints are outlined on the left side of (23) below, where 'a' is Alex, 'g' is Gila and 'a*g' is the group whose members are Alex and Gila. I have left off the '+' subscripts on the denotation symbol. Since the group itself weighs less than 3 kilo, $\|L\|(a*g)=1$ (line ii). Since the first conjunct in (16b) is true while the antecedent in the conditional is false (line viii), (16b) comes out true. Turning to second situation, where Alex and Gila both weigh 2 kilo, (16) is now intuitively false. The results of invoking the semantic constraints and rules in this case are outlined on the right side of (23). In this case, the group itself weighs more than three kilo, hence $\|L\|(a*g)=0$ (line ii), however, $\|L(\text{Alex}' + \text{Gila}')\|=1$ because when the extension of the predicate is closed under group formation, a*g comes in (lines iii,iv). In effect, this is the distributive construal and in fact, this is why the consequent of the conditional comes out false (line ix) and so the entire (16b) comes out false.

(23)

S1. $\text{wt}(\text{Alex})=\text{wt}(\text{Gila})=1 \text{ kg}$ S2. $\text{wt}(\text{Alex})=\text{wt}(\text{Gila})=2 \text{ kg}$

- | | |
|--|---|
| i. $\ L\ (a)=\ L\ (g)=1$ | $\ L\ (a)=\ L\ (g)=1$ |
| ii. $\ L\ (a*g)=1$ | $\ L\ (a*g) = 0.$ |
| iii. $\text{Grp-Cl}(\{u:\ L\ (u)=1\})=$
$\{a,g,a*g\}$ | $\text{Grp-Cl}(\{u:\ L\ (u)=1\})=$
$\{a,g,a*g\}$ |
| iv. $\ L(\text{Alex}' + \text{Gila}')\ = 1$ | $\ L(\text{Alex}' + \text{Gila}')\ = 1$ |

- v. $\| \sim L \|(a) = \| \sim L \|(g) = 0$ $\| \sim L \|(a) = \| \sim L \|(g) = 0$
- vi. $\| \sim L \|(a * g) = 0$ $\| \sim L \|(a * g) = 1.$
- vii. $\text{Grp-Cl}(\{u: \| \sim L \|(u) = 1\}) = \emptyset$ $\text{Grp-Cl}(\{u: \| L \|(u) = 1\}) = \{a * g\}$
- viii. $\| \sim L(\text{Alex}' + \text{Gila}') \| = 0$ $\| \sim L(\text{Alex}' + \text{Gila}') \| = 1$
- ix. $\| \sim L(\text{Alex}') \| = \| \sim L(\text{Gila}') \| = 0$

Because of the way predicate application is interpreted (21), there are two kinds of circumstances in which a predicate can applying truthfully to a term: one distributive, the other non-distributive. Since predicate negation is defined independently of sentence negation, a predicate and its negation can simultaneously apply truthfully to one and the same term. Still in such a situation, the kind of circumstance that leads to the predicate's applying truthfully must be different from the kind of circumstance leading to its negation applying truthfully. This means that the kind of circumstance guaranteeing a predicate's applying truthfully can be determined in terms of the kind of circumstance guaranteeing its negation truthfully apply. And this means that since there is a viable analysis for distributivity, there is a viable analysis for non-distributivity. To complete this analysis a meaning for *together* is needed:

(24) *together* translates as: $\lambda x \lambda P [P(x) \ \& \ [\sim P(x) \rightarrow [\text{EACH}(\sim P)](x)]]$

This meaning is fine for the examples examined here in detail, examples whose subjects were e-type. However, *together* combines with quantificational noun phrases as well, and so a higher type meaning is required, with π , a variable of type $\langle\langle e, t \rangle, t \rangle$:

(25) *together* translates as: $\lambda \pi \lambda P [\pi(P) \ \& \ [\pi(\sim P) \rightarrow \pi(\text{EACH-}\sim P)]]$

Moreover, we have looked only at sentences whose subject contains *together* and have not looked inside verb phrases at all. This gap will be filled to some extent in the next section.

For now, I want to mention a few side benefits that follow from the way the semantics has been set up. First, consider the sentences given in (26) in a world like our own in which Bush and Reagan are American but Thatcher is not:

- (26) a. Bush and Reagan are American.
b. Bush and Thatcher are American.

The first sentence is true while the second is not, since in general if a group is American then all of its members must be. *be American*

is an inherently distributive predicate. It is inherently distributive for the simple reason that the property of being American is essentially a property of individuals. This can be captured formally by assigning to *be American* a function whose domain is the set of individuals. That function applies truthfully to Bush and to Reagan hence the group consisting of those two will be in the closure under group formation of the (positive) extension of *be American* and so (26a) is true. That function does not apply truthfully to Thatcher hence the group consisting of Bush and Thatcher will not be in the closure under group formation of the extension of *be American* and (26b) is false. (26b) is false, however, notice what happens when the predicate of (26b) is negated:

(27) Bush and Thatcher are not American.

This sentence is not true either, except possibly as a correction of (26b), in which case, I would argue, the *not* doesn't express predicate negation. It turns out that the system outlined here predicts that (27) is false, as follows. Recall that it follows from (20) and the way negative and positive denotations were connected in (19) that for any u in the universe and any predicate P :

$$\| \sim P \|_+(u) = 1 \text{ iff } \| P \|_+(u) = 0 \text{ and } \| \sim P \|_-(u) = 0 \text{ iff } \| P \|_-(u) = 1$$

This means that if *be American* denotes a function whose domain is the set of individuals, then *be not American* also denotes a function whose domain is the set of individuals. That function applies truthfully to Thatcher but not to Bush hence the group consisting of Bush and Thatcher will not be in the closure under group formation of the extension of *be not American* and (27) is false. Compare this to a more standard account in which *be American* somehow denotes the set of groups and individuals all of whom are American and in which predicate negation is defined in terms of sentence negation (i.e. (17) above holds). In such a system, *be not American* would be true of any group that did not consist solely of Americans, in particular it would be true of the Thatcher-Bush team.

The assumption that a predicate such as *be American* contains only individuals in its extension, may at first appear to contradict our earlier claim, based on verb phrase conjunction, that distributively understood predicates must contain groups in their extension. To allay fears in this direction, let us briefly consider how verb phrase conjunction is interpreted, on the assumption that a verb phrase conjunction is translated in the standard way as follows, where α' translates the first verb phrase and β' the second:

(28) $\lambda x [\alpha'(x) \ \& \ \beta'(x)]$

Assume a situation in which the extension of β' contains only the group $\{a, b\}$ while the extension of α' contains only a and b . When the variable x is assigned $\{a, b\}$ as a value, then $\alpha'(x)$ comes out

true, for $\{a,b\}$ is in the closure under group formation of the extension of α' and similarly $\beta'(x)$ comes out true. Hence, the lambda expression of (28) denotes a predicate which is true of the group $\{a,b\}$. In other words, it is possible given the way predicate application is now interpreted, for a distributively understood predicate to be conjoined with a non-distributively understood predicate without having to include groups in the extension of distributive predicates.

Finally, let me note that a discussion of predicates whose extension is undefined for certain arguments necessitates a slight change in the rule for predicate application, as follows:

(21') Predicate application:

If α is of type $\langle e,t \rangle$ and β is of type $\langle e \rangle$, $\alpha(\beta)$ is of type $\langle t \rangle$ where:

$\|\alpha(\beta)\|_i = 1$ iff $\|\beta\| \in \text{Grp-Cl}(\{u: \|\alpha\|_i(u) = 1\})$

$\|\alpha(\beta)\| = 0$ iff $\|\alpha(\beta)\| \neq 1$ and $\|\beta\| \in \text{Grp-Cl}(\text{domain of } \|\alpha\|_i)$

We turn now from the inherently distributive predicates to inherently non-distributive predicates, customarily referred to as collective predicates. *meet* is one such predicate as can be seen in the following example:

- (29) a. The boys met (on the lawn).
b. # Alex met (on the lawn).

(29a) has no distributive construal. Following what was said for the inherently distributive, this fact is captured by assigning predicates such as *meet* functions whose domain includes only groups. It follows then that (29b) is infelicitous, since the *meet*-function is not defined for Alex. And again, the domain of a predicate is inherited by its negation, so we have that:

- (30) # Alex didn't meet (on the lawn).

is infelicitous as well. Furthermore, this treatment of inherently non-distributive predicates explains why they cannot be modified by *each*, as in (31a) translated in (31b):

- (31) a. #The boys each met (on the lawn).
b. $\wedge y (y \in \text{the-boys}' \ \& \ \text{IN}(y)) \rightarrow \text{meet}'(y)$

'y' in (31b) is restricted to individuals and $\|\text{meet}'\|$ is not defined for individuals hence (31a) is infelicitous. But what is somewhat surprising about these inherently distributive predicates is that they don't seem to sit well with ad-NP *together* either:

- (32) a. Gila and Alex met in the park.
 b. # Gila and Alex together met in the park.

If *together* is a non-distributivity marker, than where should it be happier than sitting next to a non-distributive predicate? A possible solution to this mystery may be tucked away in the meaning we have assigned to *together*:

$\lambda x \lambda P [P(x) \ \& \ [\sim P(x) \rightarrow [EACH(\sim P)](x)]]$

Using this meaning for *together*, we get the following for the meaning of *Gila and Alex together*:

(33) $\lambda P [P(g+a) \ \& \ [\sim P(g+x) \rightarrow [EACH(\sim P)](g+a)]]$

This meaning is looking for a predicate such that EACH applied to the negation of that predicate is defined. However, since *met* and its negation denote functions from groups only, EACH cannot be applied to them. So, perhaps that is why *together*-NPs cannot combine with non-distributive predicates. Of course, many issues are involved here, in particular, the analysis of semantic anomaly and the question of how anomaly percolates or projects. An adequate treatment of these questions is beyond the scope of this paper. Nevertheless, I would like briefly to pursue this idea that *together* creates noun phrases that are looking for predicates that are defined for individuals. Specifically, notice that *each* itself creates predicates that behave as inherent non-distributives:

- (34) a. #John each weighs 10 lbs.
 b. # The boys each each weigh 10 lbs.

They cannot combine with individual denoting phrases and they cannot be modified by *each*. Given what has been said about *each* and about *together*, the impossibility of combining them is expected:

(35) #John and Mary together each weigh 10 lbs.

(35) is in fact out for the same reason that (34b) is: it involves a double application of *each* and that is impossible. Finally, a *together* analogue to (34a) is the following:

- (36) a. #John together weighs 10 lbs.
 b. $W(j) \ \& \ [\sim W(j) \rightarrow \lambda y [[y \in j \ \& \ IN(y)] \rightarrow \sim W(j)]]$

In other words, we do not need a special stipulation that *together* combines only with plural noun phrases.

The purpose of the above discussion was in part to blaze a path back to an ambiguity analysis of distributivity. At this point we can, following Link(1987) and Roberts(1987), allow for a covert D-operator having the semantics of *each*. In general, this will give us

a distributive reading. We needn't worry about what happens when such a predicate is combined with a *together* noun phrase since such a combination is ruled out. As in the systems mentioned, this will yield a translation procedure which disambiguates towards a vague reading and a distributive reading of an English predicate. A possibility for having a non-distributive reading would be a VP operator based on the meaning given here for *together*.

4. SEMANTICS for ad-NP *together*: PART II

This section represents a step in the direction of interpreting verb phrases compositionally and not as a single unit. To begin with, we will assume that the negative and positive extensions of two-place predicates are related as with one-place predicates: the negative extension is a function that applies to a pair of entities $\langle a, b \rangle$, just in case the positive extension applies to $\langle a, b \rangle$ and it always yields the opposite value. Next, a two-place version of the predicate application rule will be needed which in turn requires that the closure operation be extended:

(37) Closure under group formation for two places:

$\text{Grp-Cl}(R)$, the closure under group formation of a set of pairs, R , is the smallest set such that:

$$\langle a, b \rangle \in R \text{ and } \langle c, d \rangle \in R \rightarrow \langle a * c, b * d \rangle \in \text{Grp-Cl}(R)$$

I assume that $*$ is idempotent ($a * a = a$) and this guarantees that R is a subset of $\text{Grp-Cl}(R)$, otherwise that should be stated explicitly. Now the two-place rule is introduced:

(38) Two place predicate application:

for α of type $\langle e, e, t \rangle$ and β, Γ of type $\langle e \rangle$, $\|\alpha(\beta, \Gamma)\| = 1$ iff

$$\langle \|\beta\|, \|\Gamma\| \rangle \in \text{Grp-Cl}(\{ u \in D_e \times D_e : \|\alpha\|_i(u) = 1 \}).$$

$$\|\alpha(\beta, \Gamma)\| = 0 \text{ iff } \|\alpha(\beta, \Gamma)\| \neq 1 \text{ and } \langle \|\beta, \Gamma\| \rangle \in \text{Grp-Cl}(\text{Dom}(\|\alpha\|_i))$$

It should be clear how this rule can be generalized by defining a group closure for sets of sequences of any length thus incorporating the one and two place rules as well. Furthermore, if $*$ is defined as follows:

(39) $a * b$ is the group whose members are a and b , if $a, b \in D_e$ and $a \neq b$, otherwise, $a * b = a$.

the rule for predicate application could be completely generalized for sequences of any length of entities of any type. For clarity, I will not make this generalization.

Next, we will use lambda abstraction to translate quantificational structures, so we need a rule for that:

(40) Lambda abstraction

If Ω is of type a , and x is a variable of type b , then $\lambda x[\Omega]$ is of type $\langle b, a \rangle$ such that for $\#$ equal to $+$ or $-$:

$$\| \lambda x[\Omega] \|_{\langle b, a \rangle} = h \text{ such that for any } v \text{ in } D_b, h(v) = \| \Omega \|_{a, g\{v/x\}}$$

Finally, we need to consider meanings for quantificational NPs and how they combine with predicates of $\langle e, t \rangle$. With Cooper(1983:164) as our guide, we adopt the principle that for any expression of the form $F(A)$, the negative extension of the expression is obtained by combining the negative extension of the functor with the full ordered pair extension of the argument. Likewise for the positive extension of the expression. This principle was in fact obeyed in the rules for predicate application but it was not very meaningful there since e -type expressions only have one extension.

To get a sense of how this all fits together, consider the example in (41):

(41) John and Mary together own less than three cars.

Here *together* serves to eliminate the distributive reading of the *together-less* sentence which would be true if for example John owned 2 cars and Mary owned 2. According to (41), if you add John's and Mary's cars you should come up with less than 3. (41) translates into something equivalent to the following:

(42) $[\lambda x [\text{less-than-3-cars}' \lambda y [\text{own}'(x, y)]]](\text{John}' + \text{Mary}') \ \& \ \sim [\lambda x [\text{less-than-3-cars}' \lambda y [\text{own}'(x, y)]]](\text{John}' + \text{Mary}') \rightarrow$
 $[\sim [\lambda x [\text{less-than-3-cars}' \lambda y [\text{own}'(x, y)]]](\text{John}') \ \& \ \sim [\lambda x [\text{less-than-3-cars}' \lambda y [\text{own}'(x, y)]]](\text{Mary}') \]$

In order to interpret this, an interpretation for *less-than-3-cars'* is needed. The following suggestion is again adapted from Cooper(1983:164) though his meanings are more elaborate because they take presupposition into account and less elaborate because they are not meant to cover non-distributivity:

(43) The extensions of *less than three*

$[\| \text{less-than-3} \|.(A)](X) = 1$ iff the maximal element in $A_+ \cap X_+$ has less than three members.

$[\| \text{less-than-3} \|.(A)](X) = 1$ iff the maximal element in $A_+ \cap X_+$ has more than three members.

These interpretations presuppose that the meaning of a plural common noun is the set of groups formed by closing the extension of the singular counterpart. Also, note the similarity in the relationship between the positive and negative extensions for the quantifier and the extensions of a transitive verb.

In a situation in which John owns car A and car B and Mary owns car C, we have:

$$\{\langle j,a \rangle, \langle j,b \rangle, \langle m,c \rangle\} \subseteq \llbracket \text{own}' \rrbracket.$$

Then, by the rules for lambda abstraction and two place predicate application, we have the following assuming neither John nor Mary owns anything else:

$$\begin{aligned} \{u: \llbracket \lambda y [\text{own}'(j,y)] \rrbracket.(u)=1\} &= \{a,b,a*b\} \\ \{u: \llbracket \lambda y [\text{own}'(m,y)] \rrbracket.(u)=1\} &= \{c\} \\ \{u: \llbracket \lambda y [\text{own}'(m*j,y)] \rrbracket.(u)=1\} &= \{a*c,b*c,a*b*c\} \end{aligned}$$

From the above and the rules for quantifiers and lambda abstracts we get:

$$\begin{aligned} \{u: \llbracket \lambda x [\text{less-than-3-cars}' \lambda y [\text{own}'(x,y)]] \rrbracket.(u)=1\} &= \{j, m\} \\ \{u: \llbracket \lambda x [\text{less-than-3-cars}' \lambda y [\text{own}'(x,y)]] \rrbracket.(u)=1\} &= \{j*m\} \end{aligned}$$

Finally, by our old rule of predicate application, the first conjunct of (42) comes out true. This of course, corresponds to the fact that *John and Mary own less than three cars* is true, distributively. The antecedent of the conditional of (42) also comes out true, however the consequent does not and so (42) is false as is (41) in the situation described.

NOTES

1. This work is largely a response to work by Peter Lasnik on plurals and *together* in particular. I wish to express my gratitude to him for that work. I want also to thank those who discussed this issue with me, especially Angelika Kratzer, Fred Landman, Barbara Partee and Karina Wilkinson.
2. Pairs similar to that in (3) are found in Lasnik(1990) and he in fact argues for a single interpretation of *together* in these examples. Unfortunately, space limitations prevent a full discussion of Lasnik's analysis.
3. Readers familiar with the groups (in the technical sense) of Landman(1989) may wonder if we can't get some mileage out of them here. In fact, the argument given above, transfers in the following way to Landman's theory. In order to get the desired reading for (5), Landman needs first of all to have the distributively read first conjunct be a predicate of sums and the non-distributively read second conjunct be a predicate of groups. Still, at some point the two predicates must have a common extension. To achieve this, Landman(1989:601)

introduces a general method of type shifting, one of whose instances would be the following:

$$i. \text{LIFT}({}^tP) = \lambda x_g {}^tP(x_g)$$

here x_g is a groups variable and λ is a relation that takes groups into their corresponding sums. A distributive, sums, predicate can now shift via i. to become a predicate of groups. In effect then, even for Landman distributive predicates have or can have groups in their extension.

4. Verkuyl(1987) argues that non-contrastive negation, when added to terminative sentences, can blur distributive/collective distinctions. This is accounted for in an aspectual theory.

5. I have followed Cooper(1983) in working out some of the details of a system in which positive and negative denotations are assigned. Lasnik(1988:Ch.IV) employs positive and negative denotations in the solution of other plurals problems.

6. If distributive predicates don't have groups in their extensions after all, then why not revert to the old idea that *together* annihilates distributive predicates by removing non-groups from a predicate's extension. The answer is that although a distributive predicate F doesn't contain groups in its extension, the predicate $\lambda x[F(x)]$ does.

7. In fact, such a principle was already assumed in the meaning for predicate negation. Recall, $\| \sim P \|_+ = \| P \|_-$. In other words, the positive extension of an expression depends not only on the positive extensions of its parts --- you need access to both elements of the pair extension of P .

REFERENCES

- Bennett, M. (1974) "Some Extensions of a Montague Fragment of English,"
Ph.D. Dissertation, University of California at Los Angeles.

- Cooper, R. (1983) *Quantification and Syntactic Theory*, D. Reidel, Dordrecht.
- Dowty, D. and B. Brodie, (1984) "The Semantics of Floated Quantifiers in a Transformationless Grammar," in Cobler, MacKaye & Wescoat (eds.) *Proceedings of WCCFL III*, The Stanford Linguistics Association, Stanford University, 75-90.
- Hoeksema, J. (1983) "Plurality and Conjunction," in ter Meulen (ed.) *Studies in Modeltheoretic Semantics*, GRASS 1, Foris, Dordrecht, 63-83.
- Landman, F. (1989) "Groups, I," *Linguistics and Philosophy* 12.5:559-605.
- Lasersohn, P. (1988) "A Semantics for Groups and Events," Ph.D. Dissertation, Ohio State University, Columbus.
- Lasersohn, P. (1990) "Group Action and Spatio-Temporal Proximity," *Linguistics and Philosophy* 13.2:179-206.
- Link, G., 1984, 'Hydras. On the Logic of Relative Clause Constructions with Multiple Heads' in Landman and Veltman (eds.), *Varieties of Formal Semantics*, GRASS 3, Foris, Dordrecht, 245-257.
- Link, G. (1987) "Generalized Quantifiers and Plurals," in P. Gärdenfors (ed.) *Generalized Quantifiers: Linguistic and Logical Approaches. Studies in Linguistics and Philosophy* 31, D. Reidel, Boston.
- Roberts, C. (1987) "Modal Subordination, Anaphora and Distributivity," Ph.D. Dissertation, University of Massachusetts, Amherst.
- Roberts, C., (to appear) "Distributivity & Reciprocal Distributivity," in *Proceedings of SALT I*.
- Verkuyl, H.J. (1987), "Aspect, Quantification and Negation," in J.A.G. Groenendijk, M.B.J. Stokhof and F. Veltman (eds.), *Proceedings of the Sixth Amsterdam Colloquium*, April 13-16, 1987, Amsterdam, 353-372.

Duality Distribution with Nominal and Aspectual Quantifiers in Dutch

Hans Smessaert
K. U. Leuven

0. Introduction

In this paper¹ we want to demonstrate that the adverbs of aspectual quantification in Dutch allow much richer duality configurations than their nominal counterparts. In part one, which deals with nominal quantification, we will start from the analysis of the basic duality group of Dutch *<geen* ($\neg\exists$), *enkele* ($\exists$), *niet alle* ($\neg\forall$), *alle* ($\forall$)> in section 1.1. and then proceed to the metrical group around *veel* (*many*) in 1.2. and the normative group around *te veel* (*too many*) in 1.3.. The same structure is adopted for the aspectual quantification in part two : the basic duality group of *<nog niet* (*not yet*), *al* (*already*), *nog* (*still*), *niet meer* (*no longer*)> is discussed in section 2.1., the metrical one in 2.2., and the normative one in 2.3.. In a third part we will suggest an explanation for the difference in duality distribution between the nominal and the aspectual system in terms of the difference between conservativity and the presupposition of context-change.

1. Nominal Quantification

1.1. The basic frame

As a starting point let us take a formal language L similar to the language $L(GQ)$ in Barwise & Cooper (1981:169). A model M for L can then be defined as a pair $\langle E, \llbracket \cdot \rrbracket \rangle$, namely a set of entities E and an interpretation function $\llbracket \cdot \rrbracket$. For the sake of simplicity, we take the set $\{a, b, c, d, e\}$ as a minimal universe E . For any quantified formula $Q(A, B)$ in L the quantifier frame QF can be defined as a partition of the powerset of E , $\wp(E)$, based on the cardinality of the intersection of the subsets of E and the domain set $\llbracket A \rrbracket$:

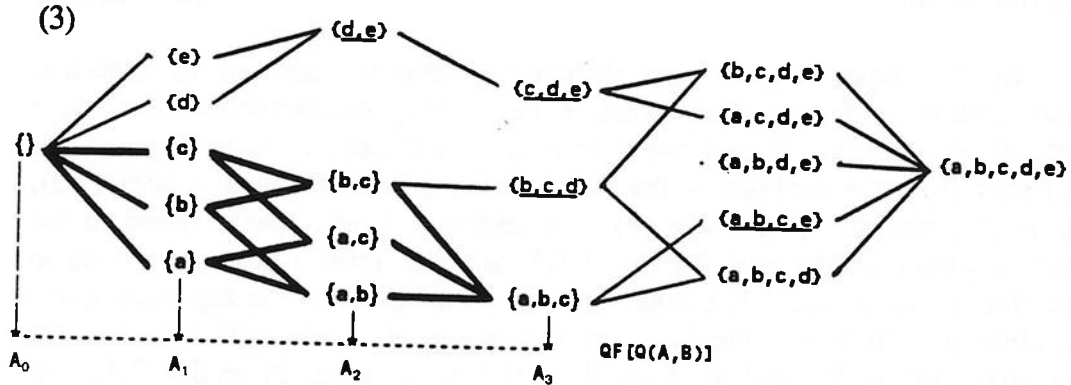
$$(1) \quad QF[Q(A, B)] = \{ A_i : 0 \leq i \leq |\llbracket A \rrbracket| \} \text{ where } A_i = \{ X \subseteq E : |\llbracket A \rrbracket \cap X| = i \}$$

Even though the B predicate does not play any role in the definition, the QF is taken to operate on complete formulae in order to achieve some uniformity with the α , β and γ operators to be introduced below. If $\llbracket A \rrbracket$ is $\{a, b, c\}$, for instance, the quantifier frame of $Q(A, B)$ has four members :

$$(2) \quad QF[Q(A, B)] = \{A_0, A_1, A_2, A_3\}$$

The set A_0 contains all the subsets of E whose intersection with $\llbracket A \rrbracket$ is empty, A_1 those whose intersection with $\llbracket A \rrbracket$ contains one element, and so on. Furthermore, the QF set as a whole is strictly linearly ordered by virtue of the

strict ordering of the indices of its members. To visualize this frame we first build the Boolean lattice² defined by $\wp(E)$. The essential parts of this lattice are represented in (3) below. The powerset of $\llbracket A \rrbracket$, $\wp(\llbracket A \rrbracket)$, which is shown in boldface in (3), then constitutes a Boolean sublattice of $\wp(E)$:



The four members of $\wp(E)$ which are underlined serve as the denotation of four B predicates in L :

$$(4) \llbracket B1 \rrbracket = \{d,e\} \quad \llbracket B2 \rrbracket = \{c,d,e\} \quad \llbracket B3 \rrbracket = \{b,c,d\} \quad \llbracket B4 \rrbracket = \{a,b,c,e\}$$

On the basis of the definition in (1) each of these denotations can be associated with the corresponding member of $QF[Q(A,B)]$:

$$(5) \begin{array}{ll} a. \llbracket B1 \rrbracket \in A_0 \Leftrightarrow & |\{a,b,c\} \cap \{d,e\}| = 0 \\ b. \llbracket B2 \rrbracket \in A_1 \Leftrightarrow & |\{a,b,c\} \cap \{c,d,e\}| = 1 \\ c. \llbracket B3 \rrbracket \in A_2 \Leftrightarrow & |\{a,b,c\} \cap \{b,c,d\}| = 2 \\ d. \llbracket B4 \rrbracket \in A_3 \Leftrightarrow & |\{a,b,c\} \cap \{a,b,c,e\}| = 3 \end{array}$$

Membership relations such as these, between an element of $\wp(E)$ and an element of a quantifier frame, will play a central role in our definition of the denotation of quantifiers. Hence we introduce an operator π which, for every $Q(A,B)$ in L, determines which member of the QF the $\llbracket B \rrbracket$ belongs to :

$$(6) \pi [Q(A,B)] = A_i \Leftrightarrow \llbracket B \rrbracket \in A_i \quad (|\llbracket A \rrbracket \cap \llbracket B \rrbracket| = i)$$

In addition, the operators α and β are introduced to pick out two elements of QF which get a special status as the "poles" of the frame :

$$(7) \alpha [Q(A,B)] = A_1 \quad \beta [Q(A,B)] = A_n \Leftrightarrow |\llbracket A \rrbracket| = n$$

Informally speaking, the α defines the first real A point on the frame (A_0 does not really say much about A), whereas β determines the last A point on the

frame. Henceforth we will drop the $Q(A,B)$ part of the definitions in (6) and (7) and simply take the α , β and π to define points on the quantifier frame.

Now the general idea is that the denotation of the nominal quantifiers can be defined in terms of the linear precedence relation between α , β and π on the strictly linearly ordered quantifier frame. Consider the following definitions of Dutch *geen* (no), *enkele* (some), *niet alle* (not all) and *alle* (all), where the boldface indicates that we are dealing with the translation in L of the corresponding quantifier in Dutch, and where $\llbracket X \rrbracket = Y$ is taken as a shorthand for "X is true if and only if Y" :

$$(8) \quad \begin{array}{ll} \text{a. } \llbracket \text{geen}(A,B) \rrbracket = \pi < \alpha & \text{c. } \llbracket \text{niet alle}(A,B) \rrbracket = \pi < \beta \\ \text{b. } \llbracket \text{enkele}(A,B) \rrbracket = \pi \geq \alpha & \text{d. } \llbracket \text{alle}(A,B) \rrbracket = \pi \geq \beta \end{array}$$

These definitions can now be applied to the B predicates of (4) by mapping the respective π 's onto the quantifier frame :

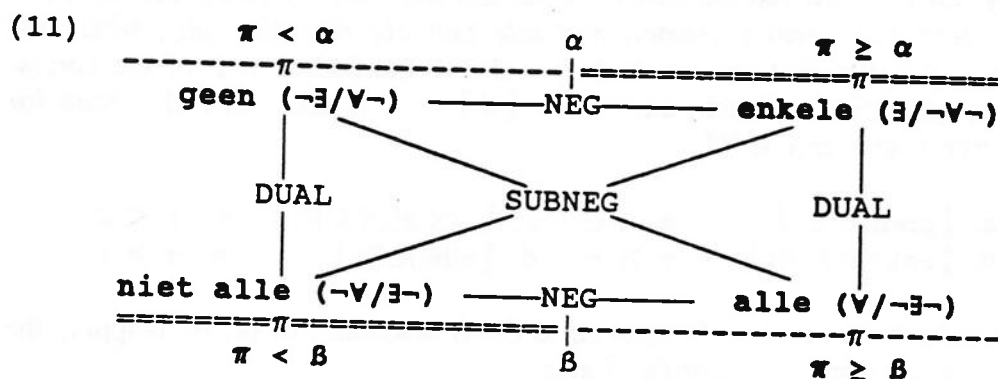
$$(9) \quad \begin{array}{ccccccc} & \pi_1 & & \pi_2 & & \pi_3 & & \pi_4 \\ & * & \text{-----} & * & \text{-----} & * & \text{-----} & * \\ & & & \alpha & & & & \beta \end{array}$$

From (9) we can infer that B1 is the only predicate satisfying the formula *geen*(A,B) in (8a), and that B4 is the only one satisfying *alle*(A,B) in (8d). The other two quantifier formulae yield the value true with three of the four B predicates : B2, B3 and B4 with *enkele*(A,B) in (8b), and B1, B2 and B3 with *niet alle*(A,B) in (8c). In addition, it can easily be shown by means of the definitions in (1), (6) and (7) that the way in which the quantifier denotations in (8) are characterized is fully compatible with that in Generalized Quantifier Theory (Barwise & Cooper, 1981:169) namely :

$$(10) \quad \begin{array}{ll} \text{a. } \pi < \alpha \Leftrightarrow \llbracket A \rrbracket \cap \llbracket B \rrbracket = \emptyset & \text{c. } \pi < \beta \Leftrightarrow \llbracket A \rrbracket \not\subseteq \llbracket B \rrbracket \\ \text{b. } \pi \geq \alpha \Leftrightarrow \llbracket A \rrbracket \cap \llbracket B \rrbracket \neq \emptyset & \text{d. } \pi \geq \beta \Leftrightarrow \llbracket A \rrbracket \subseteq \llbracket B \rrbracket \end{array}$$

Furthermore, the construction of the frame QF in (3) can be seen as a reduction of $\wp(E)$ in two steps on the basis of two general conditions on quantifier denotations. First, the Conservativity Constraint (Van Eijck, 1988:12), which assigns to any member of $\wp(E)$ a member of $\wp(\llbracket A \rrbracket)$ according to its intersection with $\llbracket A \rrbracket$, reduces $\wp(E)$ to $\wp(\llbracket A \rrbracket)$. The second step can then be seen as a "flattening" of the partially ordered $\wp(\llbracket A \rrbracket)$ sublattice to the completely linearly ordered quantifier frame. This move reflects the Cardinality or Isomorphy Constraint on quantifier denotations, according to which the identity of the particular elements in the intersection set is irrelevant. The only thing that matters is the number of elements in that set (Van Eijck, 1988:13).

Let us now turn to the Duality relations which can be shown to hold between the four quantifiers in (8), namely the NEG or external negation, the SUBNEG or internal negation, and the DUAL which is the combination of NEG and SUBNEG. These relations are brought together in a Duality Square similar to those in Löbner (1990:69ff). In addition, the denotations of (8) are mapped onto an abstract representation³ of the relevant part of the frame :



What (11) shows us is the one-to-one correspondence between the duality relations and the location of the π with respect to the poles of the frame. NEG reverses the linear precedence relation, DUAL changes the pole, and SUBNEG changes both aspects. The correlation can also be stated in terms of the two binary features $[\pm SS]$ for "switch side" and $[\pm SP]$ for "switch pole" :

(12)

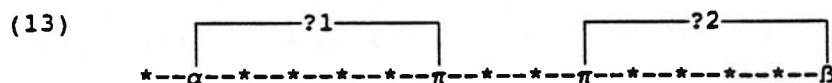
LOCATE π	$[- SS]$	$[+ SS]$
$[- SP]$	=	NEG
$[+ SP]$	DUAL	SUBNEG

We will return to this scheme, and more in particular to the $[+SS/+SP]$ status of SUBNEG in part three when we discuss the notion of complexity.

1.2. The metrical frame

The standard quantifiers defined in the previous section simply determine whether or not the predicate parameter π preceeds or follows a given pole. The function of many natural language quantifiers, however, is to be much more specific about the distance between the π and the α or β poles. In the case of cardinals, such as (*at least*) n , these distances are specified by actually counting exactly how many A's are B's, or how many A's are not B's. We will not go into these cardinal constructions in any detail here, however.

Instead we concentrate on the vague metrical quantifiers *veel* (*many*) and *de meeste* (*most*). Informally speaking, the former specifies that the distance in ?1 below between π and α is big, whereas the latter holds when the distance in ?2 below between π and β is small :



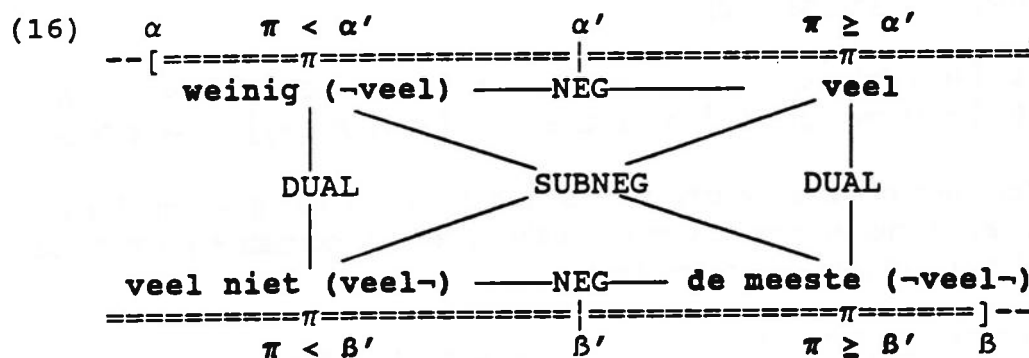
To formalize these characterizations, however, the distance between π and the α or β pole is determined indirectly by specifying the linear precedence relation between π and the secondary poles α' and β' which are located between α and β . This gives the following four definitions for *veel* (*many*), *de meeste* (*most*) and their negations :

- (14) a. $\llbracket \neg \text{veel}(A,B) \rrbracket = \pi < \alpha'$ c. $\llbracket \neg \text{de meeste}(A,B) \rrbracket = \pi < \beta'$
 b. $\llbracket \text{veel}(A,B) \rrbracket = \pi \geq \alpha'$ d. $\llbracket \text{de meeste}(A,B) \rrbracket = \pi \geq \beta'$

Here we adopt a proposal by Peterson (1979:155ff) and McCawley (1981:426ff) where *most* is defined as the dual of *many*, i.e as *not many not*. Thus we reject the traditional reading of *most* as either *more than half the* or else *almost all* (Barwise & Cooper, 1981:198; Westerstahl, 1989:12/56/71; Keenan, 1990:3). In this way the contextual dependency of *many*² and *most*² in Partee, Ter Meulen & Wall (1990:395ff) is brought together in the definition of the secondary poles α' and β' of a metrically quantified formula $Q_{\text{metr}}(A,B)$:

- (15) $\alpha' [Q_{\text{metr}}(A,B)] = A_{1+i}$ $\beta' [Q_{\text{metr}}(A,B)] = A_{n-j} \Leftrightarrow |\llbracket A \rrbracket| = n$
 with i and j fixed by context such that $i > 0$, $j > 0$, and $1+i < n-j$

On the basis of the definitions in (14) we can now build a metrical duality square similar to the one in (11) above. Notice that *niet veel* in (14a) may also surface as *weinig* (*few*), and that *niet de meeste* in (14c) tends to be realized as *veel niet*, that is the SUBNEG of *veel*. This is the result :



As was the case in (11) the duality relations in (16) are systematically related to the location of the π with respect to the α' and β' poles of the frame : NEG reverses the linear precedence relation, DUAL changes the pole, and SUBNEG changes both. In other words, the characterization of the duality relations in (12) in terms of the two binary features [$\pm$ switch side] and [$\pm$ switch pole] carries over to the metrical frame.

1.3. The normative frame

Another way of being more specific about the distances ?1 and ?2 in (13) between the π and the α or β poles involves the notion of normativity or admissibility. With adjectival opposites such as *dik/dun* (*thick/thin*) admissibility is expressed by means of the operators *genoeg* (*enough*) and *te* (*too*) and their respective negations, as in :

- (17) a. (*niet*) *dik* *genoeg* c. (*niet*) *te* *dik*
 b. (*niet*) *dun* *genoeg* d. (*niet*) *te* *dun*

Replacing *dik* and *dun* in (17) with the quantifying opposites of *veel* (*many*) and *weinig* (*few*) yields some significant gaps in the distribution, though :

- (18) a. (*niet*) (**veel*) *genoeg* c. (*niet*) *te* *veel*
 b. *(*niet*) *weinig* *genoeg* d. (*niet*) *te* *weinig*

On the basis of these restrictions, (*niet*) *te* is taken to be the primary normative operator, whereas (*niet*) *genoeg* is a secondary, derived one. Now admissibility can easily be defined in terms of the area between the secondary poles α' and β' : the former reflects the minimum level of admissibility, and the latter the maximum level. The *te* (*too*) operator then indicates that the π falls outside the admissibility area, i.e. below the minimum level, or above the maximum. This yields the following definitions for the denotations of the primary normative quantifiers in (18c) and (18d) :

- (19) a. $\llbracket \text{te weinig}(A,B) \rrbracket = \pi < \alpha'$ c. $\llbracket \neg \text{te veel}(A,B) \rrbracket = \pi \leq \beta'$
 b. $\llbracket \neg \text{te weinig}(A,B) \rrbracket = \pi \geq \alpha'$ d. $\llbracket \text{te veel}(A,B) \rrbracket = \pi > \beta'$

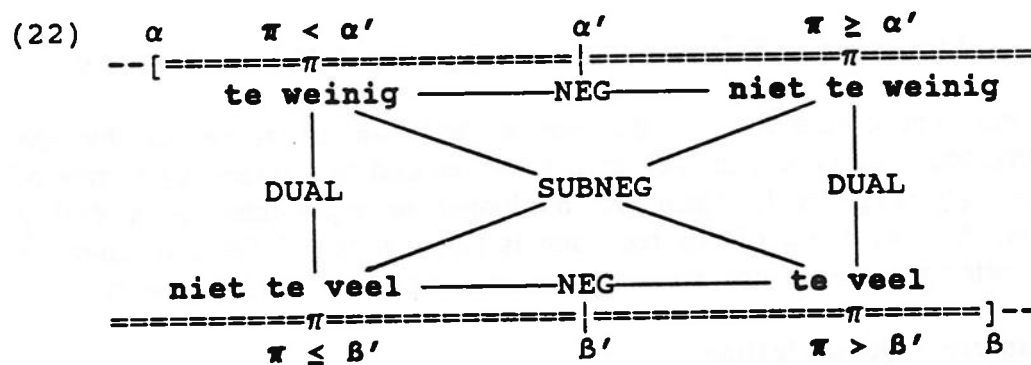
The definition of the secondary poles α' and β' , which was given in (15) for $Q_{\text{metr}}(A,B)$ formulae, now has to be modified for the normatively quantified formulae $Q_{\text{norm}}(A,B)$ in the following way :

- (20) $\alpha' [Q_{\text{norm}}(A,B)] = A_{\min}$ $\beta' [Q_{\text{norm}}(A,B)] = A_{\max}$
 with $0 \leq \min \leq \max \leq n$, min and max fixed by context as admissibility boundaries⁴.

Notice that the abstract concept of admissibility is often made much more concrete by adding an infinitival purpose-clause, as in *Hij heeft te weinig geld om het huis te kopen* (He has too little money to buy the house). As far as the duality relations between the quantifiers in (19) are concerned, the following equivalences can be posited :

- (21) a. *te veel* $\Leftrightarrow$ (*te weinig*) niet b. *te weinig* $\Leftrightarrow$ (*te veel*) niet

In other words, *te veel* and *te weinig* are one another's SUBNEG. Together with the NEG relations of (19), these SUBNEG relations allow us to build a normative duality square similar to the ones in (11) and (16) above :



Once again the duality relations in (22) are systematically related to the location of the π with respect to the α' and β' poles of the frame. However, the situation is somewhat more complex than in (11) and (16) above. Let us therefore define the following relations between linear precedence symbols : switch side [SS] for reversal in the strict sense, switch coincidence [SC] for the omission or addition of identity, and [SS/C] for the combination of the two :

- (23) a. $< \leftrightarrow \text{SS} \rightarrow \geq$ b. $< \leftrightarrow \text{SC} \rightarrow \leq$ c. $< \leftrightarrow \text{SS/C} \rightarrow >$
 $> \leftrightarrow \text{SS} \rightarrow \leq$ $> \leftrightarrow \text{SC} \rightarrow \geq$ $\leq \leftrightarrow \text{SS/C} \rightarrow \geq$

As in (11) and (16), the NEG in (22) simply switches sides [+SS]. The DUAL, however, switches both the pole and the coincidence [+SP/C], whereas the SUBNEG switches all three, namely side, pole and coincidence. They can be brought together as follows :

(24)

LOCATE π	[- SS]	[+ SS]
[- SP/C]	=	NEG
[+ SP/C]	DUAL	SUBNEG

The only difference between the scheme in (24) and that in (12) is that the simple binary feature [$\pm$ switch pole] is replaced by a more complex one [$\pm$ switch pole + coincidence]. The resulting location of the duality relations is identical, though.

Now how about the position of the so-called secondary normative quantifier in (18a), namely *genoeg*. The hypothesis is that *genoeg* has the value of the complex coordinated quantifier *niet te weinig en niet te veel*. In other words, *genoeg* corresponds to the admissibility interval from α' to β' . Hence, the definition of its denotation involves reference to both secondary poles at the same time⁵. Its negation *niet genoeg*, however, gets the same unipolar definition as *te weinig* in (19a) :

$$(25) \text{ a. } \llbracket \neg \text{genoeg}(A,B) \rrbracket = \pi < \alpha' \quad \text{b. } \llbracket \text{genoeg}(A,B) \rrbracket = \alpha' \leq \pi \leq \beta'$$

An important consequence of this bipolar definition of *genoeg* is that the original four-way system in (22) tends to be reduced to a three-way system of *te weinig/genoeg/te veel*, which can no longer be represented as a duality square. Another effect of this reduction is that *niet teveel* (*not too many*) in (19c) often gets the weaker, non-normative reading of *niet veel* (*not many*).

2. Aspectual Quantification

2.1. The basic frame

The basic group of aspectual quantifiers in Dutch consists of *nog niet* (*not yet*), *al* (*already*), *nog* (*still*), and *niet meer* (*no longer*). We will restrict our attention to the use of these elements in imperfective or durative sentences such as the following :

- (26) Hij was om drie uur *nog niet/al/nog/niet meer* aan het lezen.
He was *not yet/already/still/no longer* reading at 3 o'clock.

As is discussed in great detail in Löbner (1990:113ff), the aspectual quantifiers carry certain presuppositions about state-changes. The presupposition of *nog niet* and *al* is that there is a transition from a state at which the bare proposition *He was reading* does not hold, to one in which it does, i.e. a transition from *not reading* [-] to *reading* [+]. Exactly the opposite holds for *nog* and *niet meer* : both presuppose a transition from *reading* [+] to *not reading* [-]. The actual denotation of the quantifiers then depends on whether or not the bare proposition holds at the time *t* specified by the adverbial, such as *at 3 o'clock* in (26). In the case of *nog niet* and *niet meer*, the bare proposition

does not hold at *t*, whereas with *al* and *nog* it does hold at *t*. We can visualize the combination of presupposition and denotation of these four quantifiers as follows, where the [+] area stands for a state in which the proposition holds, and the [-] area for a state in which it does not :

$$(27) \quad \begin{array}{ccccccc} \text{nog niet} & & \text{al} & & \text{nog} & & \text{niet meer} \\ \text{-----t-----} & | & \text{++++++t++++++} & & \text{++++++t++++++} & | & \text{-----t-----} \end{array}$$

Let us now try to reformulate these basic facts about the presupposition and denotation of the aspectual quantifiers in such a way as to allow explicit comparison with the formalism introduced for nominal quantification in part one⁶. First we abstract away from the real-world continuity of the flow of time by defining a discrete axis *I* as the strictly ordered countably infinite set of points in time. The denotation $\llbracket p \rrbracket$ of a bare proposition can then be defined as a subset of *I*, that is a function which for every point on *I* determines whether [+] or not [-] it belongs to $\llbracket p \rrbracket$. If we take the indices of the points to stand for the points themselves, the following could represent a segment of *I* for a given $\llbracket p \rrbracket$:

$$(28) \quad \begin{array}{cccccccccccccc} + & + & - & - & - & + & + & + & + & - & - & - & + & + \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \end{array}$$

From the set *I*, which is added to the model *M* of our language *L*, the set *I*^o can be derived which consists of all the possible intervals in time⁷ :

$$(29) \quad I^o = \{ x_y : x \in I \ \& \ y \in I \ \& \ x \leq y \}$$

Only a very specific subset of *I*^o counts as the set of relevant subintervals *RSI* for a given aspectual quantifier. Two factors determine which intervals belong to *RSI*, namely the bare proposition *p* and the presupposition of the quantifier. As for the latter, the transition from negative to positive will be represented by $\neg Q$, whereas that from positive to negative will be indicated by $\neg Q$. The respective *RSI* sets are obtained by the following recursive definitions :

$$(30) \quad \begin{aligned} RSI[\neg Q(p,t)] = \{ i_j \in I^o : & \exists u (i < u \leq j \\ & \& \forall x [i \leq x < u \rightarrow x \notin \llbracket p \rrbracket] \ \& \ \forall y [u \leq y \leq j \rightarrow y \in \llbracket p \rrbracket]) \\ & \& \neg \exists i' [i' < i \ \& \ i'_j \in RSI[\neg Q(p,t)]] \\ & \& \neg \exists j' [j' > j \ \& \ i_j' \in RSI[\neg Q(p,t)]] \} \end{aligned}$$

$$\begin{aligned} RSI[\neg Q(p,t)] = \{ i_j \in I^o : & \exists v (i \leq v < j \\ & \& \forall x [i \leq x \leq v \rightarrow x \in \llbracket p \rrbracket] \ \& \ \forall y [v < y \leq j \rightarrow y \notin \llbracket p \rrbracket]) \\ & \& \neg \exists i' [i' < i \ \& \ i'_j \in RSI[\neg Q(p,t)]] \\ & \& \neg \exists j' [j' > j \ \& \ i_j' \in RSI[\neg Q(p,t)]] \} \end{aligned}$$

By virtue of the first two lines, the members of RSI contain one and only one polarity switch⁸, whereas the purpose of the latter two lines is to pick out the maximal relevant subintervals. For the $\llbracket p \rrbracket$ in (28), the $RSI[\nearrow Q(p,t)]$ contains the intervals 3_9 and 10_15, whereas the $RSI[\searrow Q(p,t)]$ contains 1_5 and 6_13. Each of these intervals counts as an aspectual quantifier frame QFn , where the index n keeps track of which particular occurrence of the p is focussed on in a given context. For the p in (28) p_1 is 1_2, p_2 is 6_9, and p_3 is 14_15 :

$$(31) \begin{array}{ll} QF2[\nearrow Q(p,t)] = 3_9 & QF3[\nearrow Q(p,t)] = 10_15 \\ QF1[\searrow Q(p,t)] = 1_5 & QF2[\searrow Q(p,t)] = 6_13 \end{array}$$

Remember from part one that the α and β poles were defined as operators on quantified formulae $Q(A,B)$, and that both belonged to one and the same nominal quantifier frame. The aspectual situation is different in both respects. First of all, the aspectual poles γ and δ are operators on quantifier frames QFn , and secondly each QFn contains only one pole : a γ pole for $\nearrow Q$ frames, and a δ pole for $\searrow Q$ frames :

$$(32) \begin{array}{l} \gamma (QFn[\nearrow Q(p,t)]) = u \in QFn : \forall x \in QFn (x < u \rightarrow x \notin \llbracket p \rrbracket) \\ \quad \& \forall y \in QFn (u \leq y \rightarrow y \in \llbracket p \rrbracket) \\ \delta (QFn[\searrow Q(p,t)]) = v \in QFn : \forall x \in QFn (x \leq v \rightarrow x \in \llbracket p \rrbracket) \\ \quad \& \forall y \in QFn (v < y \rightarrow y \notin \llbracket p \rrbracket) \end{array}$$

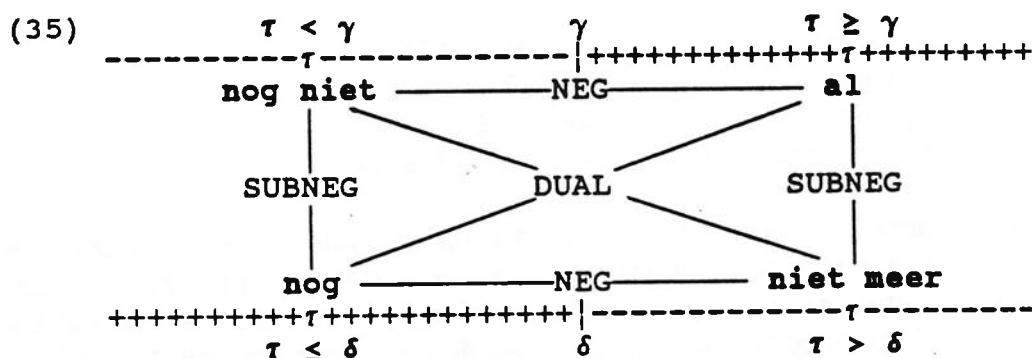
Furthermore, we will make the simplifying assumption that γ and δ are points on an aspectual quantifier frame : γ is the first positive point on a QFn with a $[-][+]$ transition, whereas δ is the last positive point on a QFn with a $[+][-]$ transition. In the same way as the denotation of a nominal quantifier was defined in terms of the linear precedence relation between a predicate parameter π and the poles α or β (or the secondary ones α' and β'), the denotation of an aspectual quantifier will be defined in terms of the precedence relation between a temporal parameter τ , corresponding to the temporal adverbial t in the language L , and the γ or δ poles on an aspectual frame. The function of this parameter τ , which should fall in one of the QFn intervals, is precisely to pick out the contextually relevant occurrence of the p , i.e. the index of QFn :

$$(33) \tau [\nearrow Q(p,t)] = w \in QFn[\nearrow Q(p,t)] \text{ where } x = \nearrow \text{ or } x = \searrow$$

The denotation of the four aspectual quantifiers is then defined as follows :

$$(34) \begin{array}{ll} \text{a. } \llbracket \text{nog niet}(p,t) \rrbracket = \tau < \gamma & \text{c. } \llbracket \text{nog}(p,t) \rrbracket = \tau \leq \delta \\ \text{b. } \llbracket \text{al}(p,t) \rrbracket = \tau \geq \gamma & \text{d. } \llbracket \text{niet meer}(p,t) \rrbracket = \tau > \delta \end{array}$$

As these quantifiers constitute a duality group (Löbner:1990:78 etc.), their four denotations can again be mapped onto a square of the general format used in part one, namely :



Once again the duality relations are systematically related to the location of the τ with respect to the γ and δ poles of the frames. Remember from (23) above that we defined a number of relations between linear precedence symbols, namely : switch side [SS], switch coincidence [SC], and the combination of the two [SS/C]. As with the nominal squares in (11), (16) and (22), the NEG in (35) simply switches sides [+SS]. The other two duality relations have a different effect, however. The SUBNEG switches both the pole and the coincidence [+SP/C], whereas the DUAL switches all three, namely side, pole and coincidence. The three relations are brought together in (36) below. The crucial difference between this scheme and the ones in (12) and (24) above lies in the location of the SUBNEG and the DUAL. We will return to this observation in part three.

(36)

LOCATE τ	[- SS]	[+ SS]
[- SP/C]	=	NEG
[+ SP/C]	SUBNEG	DUAL

2.2. The metrical frame

Remember from (13-14) above that the nominal metrical quantifiers allowed us to specify the distance between the predicate parameter τ and the α or β pole. Due to the conservative nature of the quantifier frame this distance could only stretch to the right of α or to the left of β . Because of the presuppositionality of the aspectual quantifiers, however, their frames crucially involve both a negative and a positive half. As a consequence, the metrical

quantifiers can specify distances on both sides of either γ or δ . If the distance is big, the modifier *lang* (*long*) is used. Consider the following translation rules (indicated by the $\rightarrow$ arrow) mapping the adverbials in Dutch onto the quantifier expressions in the formal language L :

- (37) a. *nog lang niet* $\rightarrow$ **lang NOG NIET**
 b. *al lang* $\rightarrow$ **lang AL**
 c. *nog lang* $\rightarrow$ **lang NOG**
 d. *(al) lang niet meer* $\rightarrow$ **lang NIET MEER**

If the distance is small, the modifier *net* (*just*) combines with the negative quantifier heads *nog niet* and *niet meer*, whereas *niet lang* (*not long*) is used with the positive *al* and *nog*. The complication with the latter two is that the negation of the modifier in L has to be derived from the negation of the quantifier head in Dutch. Hence, the translation rules are rather complex :

- (38) a. *nog net niet* $\rightarrow$ **niet lang NOG NIET**
 b. *nog niet lang* $\rightarrow$ **niet lang AL**
 c. *niet lang meer* $\rightarrow$ **niet lang NOG**
 d. *net niet meer* $\rightarrow$ **niet lang NIET MEER**

As we did with the nominal metrical quantifiers, the distance between the parameter τ and the poles γ and δ will be defined indirectly in terms of the linear precedence relation between τ and the secondary poles. As the distance can be specified on both sides of γ and δ , we introduce four secondary poles, namely γ' and γ'' for γ , and δ' and δ'' for δ , with the single prime pole in the positive half of the frame and the double prime one in the negative half :

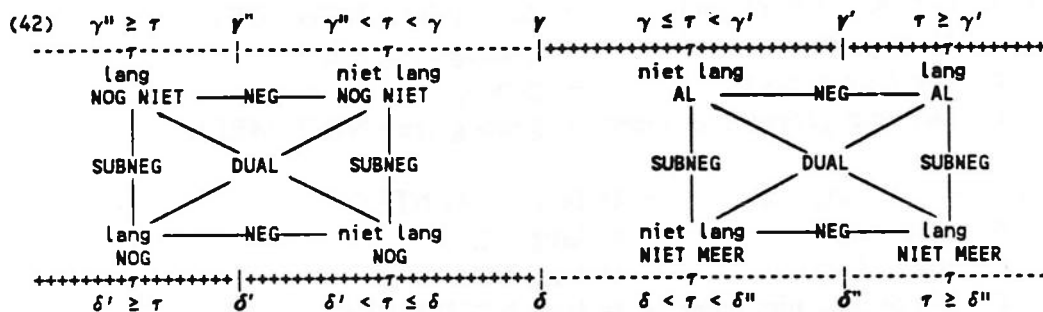
- (39) a. $\gamma''(QFn[Q_{metr}(p,t)]) = u'' \in QFn : u'' = \gamma - i$
 b. $\gamma'(QFn[Q_{metr}(p,t)]) = u' \in QFn : u' = \gamma + j$
 c. $\delta'(QFn[Q_{metr}(p,t)]) = v' \in QFn : v' = \delta - k$
 d. $\delta''(QFn[Q_{metr}(p,t)]) = v'' \in QFn : v'' = \delta + l$
 with i, j, k and l fixed by context such that $i > 0, j > 0, k > 0, l > 0$

The denotation of the "long-distance" quantifiers in (37) is given in (40). The denotations of the quantifiers in (38), which express relative proximity to the primary pole in (41), are somewhat more complex in that they require a bipolar definition to characterize the proper subinterval on the frame :

- (40) a. $\llbracket \text{lang NOG NIET}(p,t) \rrbracket = \gamma'' \geq \tau$
 b. $\llbracket \text{lang AL}(p,t) \rrbracket = \tau \geq \gamma'$
 c. $\llbracket \text{lang NOG}(p,t) \rrbracket = \delta' \geq \tau$
 d. $\llbracket \text{lang NIET MEER}(p,t) \rrbracket = \tau \geq \delta''$

- (41) a. $[\text{niet lang NOG NIET}(p,t)] = \gamma'' < \tau < \gamma$
 b. $[\text{niet lang AL}(p,t)] = \gamma \leq \tau < \gamma'$
 c. $[\text{niet lang NOG}(p,t)] = \delta' < \tau \leq \delta$
 d. $[\text{niet lang NIET MEER}(p,t)] = \delta < \tau < \delta''$

As for the duality relations among these quantifiers, the NEG simply operates on the modifier *lang*, the SUBNEG affects the quantifier head, and the DUAL combines the effects of NEG and SUBNEG. The resulting Duality Square is much more complex, however, as it duplicates the structure of (35) namely :



The NEG not only switches the side of τ with respect to the secondary pole, but also switches the restriction on the primary pole, the latter being crucial for the negative members of (42) but redundant for the positive ones. To reflect this complication, we replace the $[\pm \text{SS}]$ feature of (36) with $[\pm \text{SS/R}]$ covering the combined effect of switching side and restriction. The SUBNEG of the positive quantifiers in (42) simply switches the secondary pole. Unlike the obvious switches with $\alpha\text{-}\beta$, $\alpha'\text{-}\beta'$, and $\gamma\text{-}\delta$ above, however, we have to state explicitly which secondary poles switch in the case of (42), namely :

- (43) a. $\gamma'' \leftarrow \text{SP} \rightarrow \delta'$ b. $\gamma' \leftarrow \text{SP} \rightarrow \delta''$

The SUBNEG of the negative members of (42) switches both the secondary and the primary pole, and hence also involves the coincidence switch of (23b-24) and (36) above. We use the feature $[\pm \text{SP(C)}]$ instead of $[\pm \text{SP/C}]$ in (36), though, to indicate that the coincidence switch with the former does not affect the central precedence relation between τ and the secondary pole. As in (36), the DUAL can be defined as the combination of NEG and SUBNEG :

(44)

LOCATE τ	$[- \text{SS/R}]$	$[+ \text{SS/R}]$
$[- \text{SP(C)}]$	=	NEG
$[+ \text{SP(C)}]$	SUBNEG	DUAL

2.3. The normative frame

As was the case with the metrical quantifiers in the previous section, the aspectual notion of admissibility can specify distances on both sides of either γ or δ . If the distance is beyond the minimal or maximal admissibility boundary, the modifiers *lang genoeg* (long enough) and *te lang* (too long) are used. Consider the following rules translating the Dutch adverbials to the quantifier expressions in the formal language L :

- (45) a. *nog lang genoeg niet* → **genoeg lang NOG NIET**
 b. *al lang genoeg* → **genoeg lang AL**
 c. *nog lang genoeg* → **genoeg lang NOG**
 d. *(al) lang genoeg niet meer* → **genoeg lang NIET MEER**
- (46) a. *nog te lang niet* → **te lang NOG NIET**
 b. *al te lang* → **te lang AL**
 c. *nog te lang* → **te lang NOG**
 d. *(al) te lang niet meer* → **te lang NIET MEER**

Notice, incidentally, that the same *al* as in (37d) shows up to reinforce the *niet meer* in (45d) and (46d). In addition, the mechanism underlying (38b-c), which derives the negation of the modifier in L from the negation of the quantifier head in Dutch, works for all negative counterparts of (45-46). In the case of (45d) and (46d) it is the extra *al* which is considered to be the quantifier head. Consider the translation rules for the respective negations, some of whose input is marked with a question mark because of the reduced processibility :

- (47) a. *?niet lang genoeg meer niet* → **niet genoeg lang NOG NIET**
 b. *nog niet lang genoeg* → **niet genoeg lang AL**
 c. *niet lang genoeg meer* → **niet genoeg lang NOG**
 d. *?nog niet lang genoeg niet meer* → **niet genoeg lang NIET MEER**
- (48) a. *?niet te lang meer niet* → **niet te lang NOG NIET**
 b. *nog niet te lang* → **niet te lang AL**
 c. *niet te lang meer* → **niet te lang NOG**
 d. *?nog niet te lang niet meer* → **niet te lang NIET MEER**

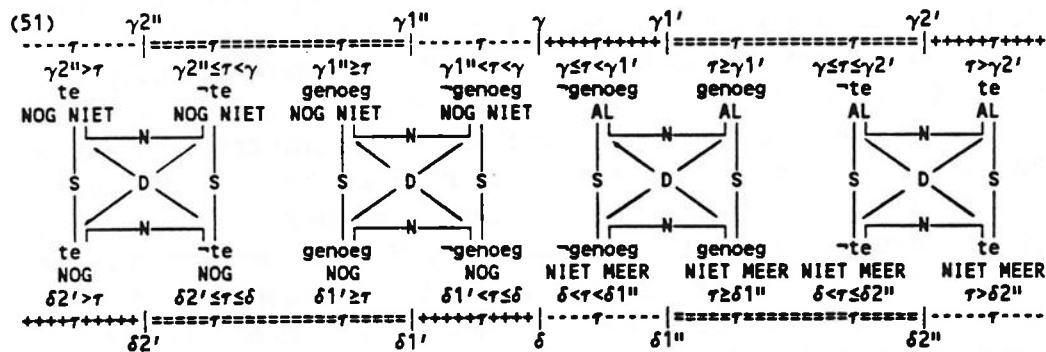
In order to locate these sixteen quantifiers with respect to the four admissibility areas, we introduce eight secondary poles, namely γ_1' , γ_2' , γ_1'' and γ_2'' for γ , and δ_1' , δ_2' , δ_1'' and δ_2'' for δ . Again, the single prime poles are located in the positive half of the frame and the double prime ones in the negative half :

- (49) a. $\gamma 1''(QFn[\neg Q_{norm}(p,t)]) = u1'' \in QFn : u1'' = \gamma\text{-imin}$
 $\gamma 2''(QFn[\neg Q_{norm}(p,t)]) = u2'' \in QFn : u2'' = \gamma\text{-imax}$
b. $\gamma 1'(QFn[\neg Q_{norm}(p,t)]) = u1' \in QFn : u1' = \gamma\text{-jmin}$
 $\gamma 2'(QFn[\neg Q_{norm}(p,t)]) = u2' \in QFn : u2' = \gamma\text{-jmax}$
c. $\delta 1'(QFn[\neg Q_{norm}(p,t)]) = v1' \in QFn : v1' = \delta\text{-kmin}$
 $\delta 2'(QFn[\neg Q_{norm}(p,t)]) = v2' \in QFn : v2' = \delta\text{-kmax}$
d. $\delta 1''(QFn[\neg Q_{norm}(p,t)]) = v1'' \in QFn : v1'' = \delta\text{-lmin}$
 $\delta 2''(QFn[\neg Q_{norm}(p,t)]) = v2'' \in QFn : v2'' = \delta\text{-lmax}$
with $\text{imin} > 0$, $\text{lmin} > 0$, $\text{xmin} \leq \text{xmax}$, fixed by context as admissibility boundaries⁹.

As was the case with the metrical frame, the difference in denotation between the positive quantifiers in (45-46) and their negations in (47-48) lies in the fact that the former only refer to a secondary pole, whereas the latter involve both a primary and a secondary pole¹⁰ :

- (50)
- | | | | |
|-----------------------|--------------------------|----------------------------|----------------------------------|
| a. [genoeg NOG NIET] | $= \gamma 1'' \geq \tau$ | [$\neg$ genoeg NOG NIET] | $= \gamma 1'' < \tau < \gamma$ |
| b. [genoeg AL] | $= \tau \geq \gamma 1'$ | [$\neg$ genoeg AL] | $= \gamma \leq \tau < \gamma 1'$ |
| c. [genoeg NOG] | $= \delta 1' \geq \tau$ | [$\neg$ genoeg NOG] | $= \delta 1' < \tau \leq \delta$ |
| d. [genoeg NIET MEER] | $= \tau \geq \delta 1''$ | [$\neg$ genoeg NIET MEER] | $= \delta < \tau < \delta 1''$ |
-
- | | | | |
|-------------------|-----------------------|------------------------|-------------------------------------|
| e. [te NOG NIET] | $= \gamma 2'' >$ | [$\neg$ te NOG NIET] | $= \gamma 2'' \leq \tau < \gamma$ |
| f. [te AL] | $= \tau > \gamma 2'$ | [$\neg$ te AL] | $= \gamma \leq \tau \leq \gamma 2'$ |
| g. [te NOG] | $= \delta 2' > \tau$ | [$\neg$ te NOG] | $= \delta 2' \leq \tau \leq \delta$ |
| h. [te NIET MEER] | $= \tau > \delta 2''$ | [$\neg$ te NIET MEER] | $= \delta < \tau \leq \delta 2''$ |

As for the duality relations among these sixteen quantifiers, we see that the NEG simply operates on the modifiers (lang) *genoeg* and *te* (lang), the SUBNEG affects the quantifier head, and the DUAL combines the effects of NEG and SUBNEG. The resulting Duality Square duplicates the structure of (42) :



As in (42) the NEG switches both the side of τ with respect to the secondary pole, and the restriction on the primary pole. For the SUBNEG of the positive quantifiers, which switches the secondary pole, the definition of (43) can be generalized as follows :

(52) a. $\gamma x'' \leftarrow SP \rightarrow \delta x'$ b. $\gamma x' \leftarrow SP \rightarrow \delta x''$

For the SUBNEG of the negative members of (51), which switches both the secondary and the primary pole, we can again use the $[\pm SP(C)]$ feature introduced for (44). As in (36) and (44), the DUAL can be defined as the combination of NEG and SUBNEG :

(53)

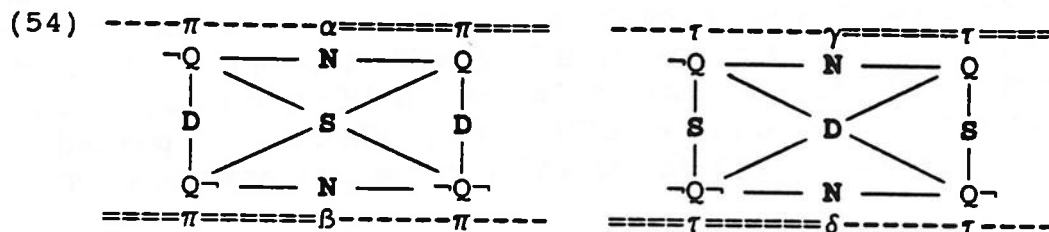
LOCATE τ	$[- SS/R]$	$[+ SS/R]$
$[- SP(C)]$	=	NEG
$[+ SP(C)]$	SUBNEG	DUAL

3. Conclusion

A first important difference between the nominal and the aspectual duality configurations presented in parts one and two respectively is their complexity. Although both the nominal basic group in (11) and the aspectual one in (35) have four members, the respective metrical and normative configurations are fundamentally different. In the metrical case, the nominal group in (16) contains four quantifiers, whereas its aspectual counterpart in (42) consists of eight quantifiers. The divergence is even bigger with the normative configurations : four nominal members in (22) versus sixteen aspectual ones in (51).

The Conservativity constraint, according to which anything outside the domain set A is irrelevant for the interpretation of the Q in $Q(A,B)$, is crucial for the construction of the nominal quantifier frames. As a consequence, the metrical and normative quantifiers can only specify distances in the area between α and β , and not in the non-A areas before α and after β . More informally, one could say that "no is no" and "all is all", and that further specification of the distance between the π and the pole is useless in these two cases. This does not hold at all, however, for the aspectual quantifiers. Their dynamic nature, which is due to the presupposition of state-change (and hence context-change), makes the notion of Conservativity inapplicable¹¹. For the construction of the aspectual quantifier frames the negative area in which the proposition does not hold is as important as the positive area in which it does. Hence, the metrical and normative quantifiers do specify the negative areas before γ and after δ . With the former group this results in a duplication of the basic duality configuration, and with the latter even in a quadruplication.

A second major difference between the nominal and the aspectual systems concerns the distribution of the duality relations across the frames. Although the external negation NEG establishes the horizontal connection on both the nominal and the aspectual squares, the internal negation SUBNEG constitutes the diagonal link on the nominal squares but the vertical link on the aspectual ones, whereas the DUAL is vertical in the nominal case but diagonal in the aspectual case. Schematically this gives :



Now the definitions of the quantifier denotations basically consist of three parts, namely a parameter (π or τ), a linear precedence relation, and a primary or secondary pole. As the parameter is the same for any two members of the same square, the duality relation between the two quantifiers can be characterized in terms of the precedence and the pole. Hence we introduced binary features using two basic switching operations, namely "switch side" [$\pm$ SS] and "switch pole" [$\pm$ SP]. Disregarding the more complex versions of these features, namely [$\pm$ SP/C] in (24/36), and [$\pm$ SP(C)] and [$\pm$ SS/R] in (44/53), the different duality distribution can be summarized as follows :

(55)

LOCATE π	[- SS]	[+ SS]
[- SP]	=	NEG
[+ SP]	DUAL	SUBNEG

LOCATE τ	[- SS]	[+ SS]
[- SP]	=	NEG
[+ SP]	SUBNEG	DUAL

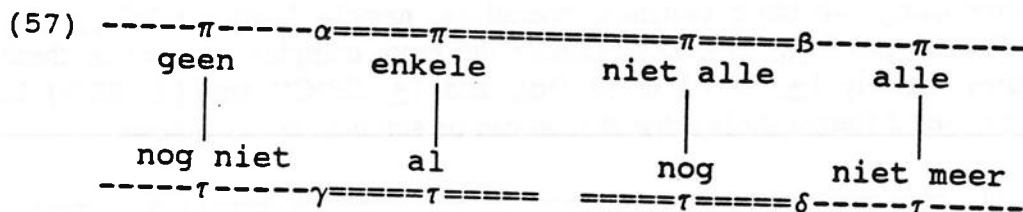
On the nominal frames the SUBNEG has the combined effect of NEG [+ SS] and DUAL [+ SP], whereas on the aspectual frames it is the DUAL which combines the [+ SS] of the NEG and the [+ SP] of the SUBNEG. In other words, there is a fundamental difference, as far as the complexity of the SUBNEG is concerned, between the nominal and the aspectual systems. Its [+SS/+SP] status in the nominal case is supported by the common practice to replace the inner negations $\exists \neg$ and $\forall \neg$ with the outer negations of the dual, namely $\neg \forall$ and $\neg \exists$ respectively. The less complex [-SS/+SP] status of the aspectual SUBNEG is confirmed by its overt morphological marking, not only in the pair *nog/nog niet*, but also with the addition of the extra *al* to *niet meer* in (37d), (45d) and (46d), which results in the pair *al .../al ... niet meer*. A

first step towards an explanation of this difference is to look at which constituent the SUBNEG operates on in the two systems :

- (56) a. nominal : $\neg Q (A, \neg B)$ b. aspectual : $\neg Q (\neg p, t)$

Due to the conservativity of the nominal quantifier in (56a), anything outside A is irrelevant for the denotation of Q, so that B is reduced to $A \cap B$. Hence, the SUBNEG that leads to the complement set $\neg B$ is not a basic operation in the nominal system, as is reflected in its [+SS/+SP] status in (55). The presupposition of the aspectual quantifiers, however, crucially involves the transitions between p and $\neg p$. Hence the SUBNEG, which affects the bare proposition p in (56b), does count as a basic [-SS/+SP] operation in the aspectual system.

Finally, one issue, which was left implicit in this paper, definitely deserves further research, namely the correspondence between the elements on the nominal basic frame and those on the aspectual basic frame. Even though α and β are two poles on one nominal frame, whereas γ and δ each belong to their own aspectual frame, there is an obvious connection between α and γ , and β and δ , and thus between the elements located with respect to them :



The crucial property of the parallellism in (57) is that it disregards the morphological criterion of overt negative marking on the β and δ poles. The latter position, which is adopted by Ter Meulen (1990) and Löbner (1990/1992), would associate *alle* with *nog*, and *niet alle* with *niet meer*. Löbner furthermore uses the idea of a markedness hierarchy to account for the parallellism. Two questions arise in this connection. Do the markedness criteria, which are well motivated for the nominal (and modal) quantifiers, simply carry over to the aspectual group ? And are there any other criteria which would lend support to the parallellism in (57) ?

Notes

1. I would like to thank G. Craps, F. Droste, D. Geeraerts, S. Löbner, L. Melis, D. Speelman and A. Ter Meulen for their comments on earlier versions of this paper.
2. At this point an extra step could be added by positing a lattice homomorphism from $\rho(E)$ to the Boolean lattice consisting of all the individual sums of members of E (In the sense of Link, 1983:306). For the reduction of this second lattice to the quantifier frame, the definition in (1) would have to be modified by replacing the intersection with some meet operation, and the cardinality with the measure function μ of Link (1987:162). This would then allow us to incorporate a considerable part of Link's analysis of plurals and mass nouns (Link, 1983/1987). The full complexity of the lattices, which is crucial for the analysis of (in)definite NPs and partitives, will play no further role in this paper.
3. This representation is slightly misleading as far as *geen* and *alle* are concerned : with the former there is only one point on QF preceding α , namely A_0 , and with the latter the π can only coincide with β and not strictly follow it. Nevertheless, we will stick to this format for the sake of comparison with the other frames developed later on. Furthermore, the restrictions on *geen* and *alle* will prove to be relevant for the discussion in part 3.
4. The restrictions on i and j in the metrical definition of (15), which prevent the coincidence of α and α' , β and β' , and α' and β' , do not hold for *min* and *max*. If $A_{\min} = A_1$ then *te weinig* = *geen*, whereas if $A_{\max} = A_{n-1}$, then *te veel* = *alle*. It is also possible that the admissibility has no lower or upper boundary, in which case $A_{\min} = A_0$ or $A_{\max} = A_n$ respectively. In addition, $A_{\min} = A_{\max}$ in those cases where only one exact cardinality is admissible.
5. The relation of Implication lends extra support to this position. On the basic frame *alle* implies *enkele* because $(\pi \geq \beta)$ implies $(\pi \geq \alpha)$. Similarly, on the metrical frame *de meeste* implies *veel* because $(\pi \geq \beta')$ implies $(\pi \geq \alpha')$. The implication from *te veel* to *genoeg* on the normative frame does not hold, however, because $(\pi \geq \beta')$ does not imply $(\beta' \geq \pi \geq \alpha')$.
6. To make the comparison between the nominal and the aspectual systems fully explicit the latter would have to be given a lattice-theoretical account similar to that of the former system in part one. This extra step will have to be postponed to a later paper.
7. This I° constitutes a join-semilattice which fulfills a role similar to the Boolean lattice $\rho(E)$ of part one, in the sense that the quantifier frame can again be defined as a flattening or reduction of the relevant sublattice of I° .
8. The notion of RSI introduced here differs from the closely related concept of Zulässiges Intervall (ZI) or Admissible Interval (AI) in Löbner (1990:117 etc) in that the latter need not contain the polarity switch.
9. The remarks made in note 4 about the difference between the definition of i/j in (15) and *min*/*max* in (20), more particularly about the possible absence of one of the admissibility boundaries, and the possible coincidence of *min* and *max* carry over, *mutatis mutandis*, to the *nmin* and *nmax* of the aspectual normative frame.
10. To keep things manageable the *lang* and the two arguments (p, t) of the quantifier are dropped. Notice that, in contrast to the bipolar definition of its nominal counterpart in (25b), the aspectual *genoeg* gets a unipolar definition because it is taken to be a primary normative modifier. The reason is that the *te*, which is basic in the nominative case, does not occur with the negation of *lang*, namely *kort* : **(niet) te kort*. However, the alternative bipolar definition would not affect any of the subsequent claims about the duality relations in any crucial way.
11. I wish to thank Alice Ter Meulen for making this clear to me.

References

- Barwise, J. and Cooper, R.: 1981, Generalized Quantifiers and Natural Language, *Linguistics and Philosophy* 4(1), 159-219
- Eijck, J. v.: 1988, Quantification, To appear in A. von Stechow and D. Wunderlich (eds.), *Handbook of Semantics*
- Keenan, E.L.: 1990, Case, Quantification and Anaphora, Paper read at the Logic and Language Universals Conference in Paris (June 1990)
- Link, G.: 1983, The Logical Analysis of Plurals and Mass Terms : A Lattice-theoretical Approach, In Bäuerle et al. (eds.), *Meaning, Use and Interpretation of Language*, de Gruyter, Berlin, 302-323
- Link, G.: 1987, Generalized Quantifiers and Plurals, In P. Gaerdenfors (ed.), *Generalized Quantifiers*, Reidel Publishing Company, Dordrecht, 151-180
- Löbner, S.: 1990, *Wahr neben Falsch. Duale Operatoren als die Quantoren natürlicher Sprache*, Linguistische Arbeiten 244, Max Niemeyer Verlag, Tübingen
- Löbner, S.: 1992, The structure of quantifier and other duality groups, To appear in P. Dekker and M. Stokhof (eds), *Proceedings of the Eighth Amsterdam Colloquium, December 19-22, 1991*, ITLI, Amsterdam
- McCawley, J.D.: 1981, *Everything that linguists have always wanted to know about Logic (but were ashamed to ask)*, University of Chicago Press, Chicago
- Partee, B.H., Ter Meulen, A. and Wall, R.E.: 1990, *Mathematical Methods in Linguistics*, Kluwer, Dordrecht
- Peterson, P.L.: 1979, On the Logic of 'few', 'many', and 'most', *Notre Dame Journal of Formal Logic* 20(1), 155-179
- Ter Meulen, A.: 1990, English Aspectual Verbs as Generalized Quantifiers, in M. Stokhof and L. Torenvliet (eds), *Proceedings of the Seventh Amsterdam Colloquium, December 19-22, 1989*, ITLI, Amsterdam, 303-315
- Westerstahl, D.: 1989, Quantifiers in formal and natural languages, In D. Gabbay and F. Guenther (eds.), *Handbook of Philosophical Logic, Volume IV*, D. Reidel Publishing Company, Dordrecht, 1-131

On the Complexity of Feature Structures in Categorical Unification Grammar

Marten Trautwein*

Department of Mathematics and Computer Science
University of Amsterdam

February 28, 1992

Abstract

The classical categorical grammar has been enriched with features, a unification and a reentrance mechanism as known from other grammatical systems (HPSG, FUG). Such extensions have been proposed to account for modifiers. We show that the recognition problem of the proposed grammar is NP-complete. The reentrance mechanism and the infinite set of feature structures cause this complexity. We conclude with some suggestions for constraints that can be put on both the reentrance mechanism and the set of features.

1 Introduction

The reentrance mechanism for features is used in a variety of grammars, like Head-driven Phrase Structure Grammar (Pollard and Sag 1988), Functional Unification Grammar (Kay 1985) and Categorical Unification Grammar. In this article, we will only be concerned with the use of the reentrance mechanism in categorical grammar. The grammar we are considering is a categorical grammar as proposed in (Bouma 1988). Bouma has used these utilities to describe modifiers like adjectives. This proposal in turn is inspired by (Bach 1983b).

The complexity theory measures the difficulty of a particular problem. Problems of equal difficulty are placed in the same class. Feasible problems are members of the complexity class P. The class NP consists of those problems. The complexity of a grammar can be measured by the complexity of the recognition problem of that grammar. The recognition problem is thus defined: For some grammar, is a string an utterance from the language generated by that grammar?

We will show that the recognition problem of this grammar is NP-complete. Hereto we first prove that this recognition problem is in the class NP and secondly make a polynomial time reduction from the well-known NP-complete 3SAT problem. This reduction will give us some insights in which components of the grammar are hard. We suggest some limitations that can be placed on them.

Before entering the above outlined path, we will discuss the components of the complexity theory that we will need further-on.

*This research was supported by the Linguistic Research Foundation, which is funded by the Netherlands organization for scientific research, NWO

2 3SAT and Reductions

The complexity theory divides problems up into complexity classes. The classes we are considering are P, NP and NP-complete. The class P consists of those problems that can be solved in polynomial time (we call them feasible problems). The class NP contains those problems for which it is possible to guess a solution and check whether it is a solution in polynomial time (hard problems). It is obvious that the class P is a subclass of the class NP. However, it is unlikely that the classes P and NP are equal. In the sequel we will assume P and NP to be unequal.

The hardest problems of NP are in the subclass NP-complete. If we would find a polynomial time solution for a NP-complete problem then we would have a polynomial time solution for all NP problems. And hence the classes NP and P would be equal. Therefore it is unlikely that polynomial solutions exist for NP-complete problems.

A well-known NP-complete problem is 3-satisfiability problem (3SAT). Suppose a propositional logic formula (conjunctive normal form, disjunctions of three literals) of the form $(x'_1 \vee y'_1 \vee z'_1) \wedge \dots \wedge (x'_m \vee y'_m \vee z'_m)$ is given, where x'_i, y'_i, z'_i are positive or negative occurrences of some variables. Is this formula satisfiable? Positive occurrences of a variable (x_i) are called positive literals; negative occurrences ($\bar{x}_i$), negative literals; disjunctions of three literals, clauses; and conjunctions of clauses simply conjunctions.

It is easy to verify whether the formula is true given an assignment for the variables. Finding an assignment that makes the formula true is hard, we have to try all possible assignments for the variables and see whether the formula is true.

The complexity of a grammar is measured by the complexity of its recognition problem. This recognition problem is thus defined: Given a grammar G . Is a string w accepted by the grammar; i.e. is the string w in the language generated by G ?

We will prove the NP-completeness of the recognition problem of the grammar given in section 3 by a reduction from 3-satisfiability (3SAT), a known NP-complete problem. Under the assumption $P \neq NP$, 3SAT is not solvable in polynomial time. Suppose a polynomial time reduction from 3SAT to this recognition problem can be found. Then, no polynomial time solution can exist for this recognition problem, since that would yield a polynomial time solution for 3SAT. NP-completeness is proven by also showing that the recognition problem is in NP.

For further information on complexity classes, 3SAT, reductions, etc. I refer the reader to (Hopcroft and Ullman 1979).

3 Extended Categorical Grammar

Bach (Bach 1983b; Bach 1983a) has proposed a categorical grammar with feature structures. This was done to account for morphological information. Bouma (Bouma 1988) has proposed a slight variant of this categorical grammar to handle modifiers, by adding a reentrance mechanism. The grammar Bouma proposed, we are interested in. Bouma's grammar can be divided in the following two components. On the one hand there is the ordinary classical categorical grammar, on the other hand the grammar uses a feature system. The feature system defines the set of features; the way feature structures are combined (unification); and the way to other feature structures is referred (reentrance). These methods are also used in other grammars, like HPSG and FUG. But we will concentrate on the categorical grammar, in this article.

Feature System. The feature system has as primitives a set of feature names, e.g. {pers, plur} and a set of feature values, e.g. {1, 2, 3, -, +}, besides a combination mechanism. Features are pairs of feature names and feature values, e.g. (pers, 3); also called feature pairs. A set of features is called a feature structure, denoted by [...]; e.g. [(pers,3), (plur,-)].

A feature structure is called inconsistent if it contains two features with the same feature name but with different feature values. Otherwise the feature structure is consistent.

Features are combined by unification. In the simple case this is ordinary set union. However, if the union would yield an inconsistent feature structure, the unification fails. The more complicated case arises from the use of reentrance.

Reentrance is used to denote that two feature structures will always be equal; i.e. if one of the feature structures changes the other will change in exactly the same way. Reentrance is accomplished by reentrance symbols, such a symbol is attached to two or more feature structures.

The feature structures that result from this feature system are attached to the categories given by the categorial grammar. This is specified in the definition of the categories. The rules to combine feature structures are invoked by the combination of categories only, which is given by the categorial grammar.

Categories. The grammar contains a set of primitive categories, PCat, and a set of feature structures, Feat. We have two notion of categories, Cat and FCat, as given below. The set Cat denotes all ordinary categories, the set FCat all categories to which feature structures are attached.

Definition of Cat. • $A \in \text{PCat} \rightarrow A \in \text{Cat}$

- $A, B \in \text{Cat} \rightarrow (A/B), (B \setminus A) \in \text{Cat}$
- Nothing else is a member of Cat.

Definition of FCat. • $A \in \text{PCat}, [F] \in \text{Feat} \rightarrow \frac{A}{[F]} \in \text{FCat}$

- $A \in \text{PCat}, B \in \text{FCat}, [F] \in \text{Feat} \rightarrow \left(\frac{A}{[F]} / \frac{B}{[F]} \right), \left(\frac{B}{[F]} \setminus \frac{A}{[F]} \right) \in \text{FCat}$
- Nothing else is a member of FCat.

Note: $\frac{A}{[F]} / \frac{B}{[G]} \frac{B}{[H]}$ is not a valid category of FCat; whereas $\frac{A}{[G]} / \frac{B}{[H]}$ is.

Rules. The only rule to combine categories is application. Application can only be performed if the unification of feature structures succeeds; i.e. the conditions on feature structures are passed.

$$\begin{array}{l} \text{Right application: } \frac{X}{[F]} / \frac{Y[G]}{[G']} \frac{Y[G']}{[F']} \rightarrow \frac{X}{[F']} \\ \text{Left application: } \frac{Y[G']}{[F']} \frac{Y[G]}{[F]} \setminus \frac{X}{[F]} \rightarrow \frac{X}{[F']} \end{array}$$

The conditions on feature structures state that $[G]$ unifies with $[G']$. This unification of $[G]$ and $[G']$ might change the reentrance symbols of $[G]$ (and $[G']$). If this has happened, the reentrance symbols of $[F]$ should also be changed accordingly. This transforming feature structure $[F]$ into $[F']$.

Example. An example may clear the use of the reentrance mechanism. Consider a functor without the reentrance mechanism $\frac{B}{[(b, +)]} \frac{A}{[(a, -)]}$, and a argument category $\frac{A}{[(a, -)]}$. Application will result in $\frac{B}{[(b, +)]}$. Using the reentrance mechanism, the functor becomes

$\frac{B}{\langle 1 \rangle [(b, +)]} \frac{A_{\langle 1 \rangle}}{[(a, -)]}$ and yields the result category $\frac{B}{\langle 1 \rangle [(a, -), (b, +)]}$. Since, it is of no use to assign a reentrance symbol to a feature structure without having a second occurrence of that reentrance symbol we may discard it. Application forces the unification of $\langle 1 \rangle$ and $[(a, -)]$, therefore feature structure $[(a, -)]$ is assigned to reentrance symbol $\langle 1 \rangle$.

4 Recognition is in NP-complete

This section is divided in two subsections. In the first part, we will show that the recognition problem of the grammar of section 3 is in NP. In the second, we will show that the problem is in NP-hard. And hence we have proved that the recognition problem of our grammar is in NP-complete.

4.1 Recognition in NP

The only rule the grammar G , as described in section 3 contains, is application. In application two categories are combined into one new category. If a string is derivable, the categories of that string can all be combined into one resulting category, the category to which the string belongs. We will refer to this category as the root category (of the string). For a string of n categories to be accepted by the grammar precisely $(n-1)$ application steps are required. Below we show that it is easy to verify one application step. And therefore it is possible to check in polynomial time whether a given sequence of application steps results in a derivation of the string.

An application step consists of two substeps. We will show that each of these substeps is easy to perform. First, we have to determine the resulting category. Given a functor and argument category, the resulting category is known. If the domain category of the functor is equal to the argument category, the result is the range category of the functor, otherwise the functor cannot be applied to the argument.

The second step consists of the unification of the feature structures. For simplicity, first suppose that neither feature structure contains reentrance symbols. Then to unify (the feature structure of the domain and the feature structure of the argument) we take the union of the feature structures. If this union is consistent, it is the result of the unification, otherwise the unification fails. The more difficult situation arises when we take the reentrance mechanism into account. The unification process may alter the reentrance symbols. Therefore we have to check all feature structures of both categories.

There is only a linear amount of reentrance symbols in each category. The time spent to perform the union of two feature structures is the product of the sizes these feature structures. Since the amount of feature names is fixed for each grammar, and we demand feature structures to be consistent, this union is $O(m^2)$ for m feature names. Hence we need $O(km^2)$ time for a grammar with largest category size k and m feature names.

Application steps are easy to verify, the derivation consists of n application steps for a string of n categories. Hence, a derivation is polynomial in the length of the string. And given a derivation we can verify that the derivation is correct in polynomial time.

4.2 Recognition in NP-hard

The recognition problem for the grammar G is NP-hard if a reduction from a known NP-complete problem to the recognition problem exists. However, this reduction must be performed in polynomial time. A well known NP-complete problem is 3SAT.

We start this section by giving the reduction of 3SAT to the recognition problem. This consists of the construction of the grammar G , and the translation of the 3SAT formula to a string of G . Then, a proof is given that the reduction can be performed in polynomial time. And finally, we prove that the 3SAT problem is equivalent to the recognition problem. This is done in a straightforward way, by showing that 3SAT implies recognition, and showing that recognition implies 3SAT.

Reduction. The reduction is best understood by an example. Suppose we have a 3SAT formula $f = (x_1 \vee \overline{x_3} \vee x_4) \wedge (x_3 \vee x_4 \vee x_5)$. This formula f is satisfiable when we assign the variable x_4 the value true. (The assignment for the other variables thereby being irrelevant.)

The string w_f the formula f translates to is $x_1 \vee \overline{x_3} \vee' x_4 \wedge x_3 \vee x_4 \vee' x_5$. More general: leave out all parenthesis and change the second disjunct ($\vee$) of each clause to $\vee'$. It should be clear that the time cost to perform this translation is linear in the length of the formula f , $O(|f|)$.

Using our intuitions and their great resemblances we will call the elements of the string after their corresponding objects in the formula f ; i.e. we will speak of literals, odd and even disjunctions, clauses, etc. of the string w_f .

The grammar consists of a fixed part and a part that depends of the formula. The fixed part gives the lexicon entries for the disjunction and conjunction symbols. Furthermore, it lists the set of primitive categories, PCat; the set of feature names and feature values; and the combination rules for categories and feature structures. In table 1 the fixed lexicon entries are specified (table 3 for the variable entries); the set of primitive categories is $\{F, T, PF, PT, S\}$; the set of features is $\{(t_i, v) | i \in \mathbb{N}, v \in \{+, -\}\}$; the application rules are given in section 3.

$\vee$:	$\frac{T_{(1)} \setminus (PT / T_{(1)})}{(1)}$	$\vee'$:	$\frac{(PT_{(1)} \setminus S)}{(1)} / T_{(1)}$
$\vee$:	$\frac{T_{(1)} \setminus (PT / F_{(1)})}{(1)}$	$\vee'$:	$\frac{(PT_{(1)} \setminus S)}{(1)} / F_{(1)}$
$\vee$:	$\frac{F_{(1)} \setminus (PT / T_{(1)})}{(1)}$	$\vee'$:	$\frac{(PF_{(1)} \setminus S)}{(1)} / T_{(1)}$
$\vee$:	$\frac{F_{(1)} \setminus (PF / F_{(1)})}{(1)}$	$\wedge$:	$\frac{(S_{(1)} \setminus S)}{(1)} / S_{(1)}$

Table 1: Fixed entries of the lexicon.

Let us explain the intended meaning behind these formalisms. The most simple ones are the set of features and the set of primitive categories. The feature $(t_i, +)$ expresses that variable x_i is true; likewise $(t_i, -)$ expresses that x_i is false. The categories F and T express that a literal is false, respectively true. The category PF , that the disjunction of the first two literals of a clause is false (partially false); PT that this disjunction is true (partially true). S expresses that a clause or conjunction of clauses is true (or satisfiable). Armed with this knowledge table 1 can be understood. It is easy to see that the assignment to the disjunction symbols is such that clauses are left-associative. The word $\vee$ combines the left-most two literals; $\vee'$ that right-most literal and the disjunction of $\vee$. The word $\wedge$ combines

only satisfied clauses. The reentrance symbols of the functors are used to pass all feature structures of their arguments up to the resulting categories.

Table 2 shows that every clause that contains a literal of category T has a derivation towards the category S. It is left to the reader to show that a clause containing three literals of category F cannot be combined. This corresponds to a clause with three false literals being unsatisfiable.

First	Literals		Disjunctions	
	Second	Third	Odd	Even
T	T	T	$T \backslash (PT/T)$	$(PT \backslash S)/T$
T	T	F	$T \backslash (PT/T)$	$(PT \backslash S)/F$
T	F	T	$T \backslash (PT/F)$	$(PT \backslash S)/T$
T	F	F	$T \backslash (PT/F)$	$(PT \backslash S)/F$
F	T	T	$F \backslash (PT/T)$	$(PT \backslash S)/T$
F	T	F	$F \backslash (PT/T)$	$(PT \backslash S)/F$
F	F	T	$F \backslash (PF/F)$	$(PF \backslash S)/T$

Table 2: Category assignments for disjunctions.

So far for the fixed part of our grammar. The next part involves the lexicon for the items corresponding to the literals of the formula. It is important that the reader understands the difference between variables and literals. The truth-value of variable i is given by the feature value associated with the feature name t_i . The truth-value of a positive (negative) literal with index i is given by the category assigned to the word x_i ($\bar{x}_i$).

For each literal two entries are introduced: according to the one entry, the literal is assigned category T, according to the other the literal is assigned category F. The feature structure attached to these categories depends on the kind of literal (positive or negative).

$$\left. \begin{array}{l} x_i : T[(t_i, +)] \\ x_i : F[(t_i, -)] \end{array} \right\} \text{ for (positive) literal } x_i$$

$$\left. \begin{array}{l} \bar{x}_j : T[(t_j, -)] \\ \bar{x}_j : F[(t_j, +)] \end{array} \right\} \text{ for (negative) literal } \bar{x}_j$$

Table 3: Variable entries of the lexicon (for each literal).

Intuitively this means the literal x_i is true (T) iff the variable is true $((t_i, +))$, likewise x_i is false (F) iff the variable is false $((t_i, -))$. Analogously for the truth-value of negative literals ($\bar{x}_i$) and the truth-value of the variable (t_i).

Given the construction recipe we can compute its time cost. For a string with n literals $2n$ entries are created. Therefore the total cost of constructing the grammar is $O(|f|)$.

Reduction can be performed in polynomial time. We have arrived at the point where we will prove that we have a polynomial time reduction. The reduction consist of two steps, translation of the formula and construction of the grammar. For the translation we have to walk through the formula once, for the grammar construction we have to create two entries for each literal in the formula. Hence, both the translation and the construction require linear, $O(|f|)$, time. The total time the reduction takes, is therefore linear, $O(|f|)$.

Continuing the example. We have already given the translation of our example formula f , $x_1 \vee \bar{x}_3 \vee' x_4 \wedge x_3 \vee x_4 \vee' x_5$. The corresponding grammar, G_f , we will present here. The

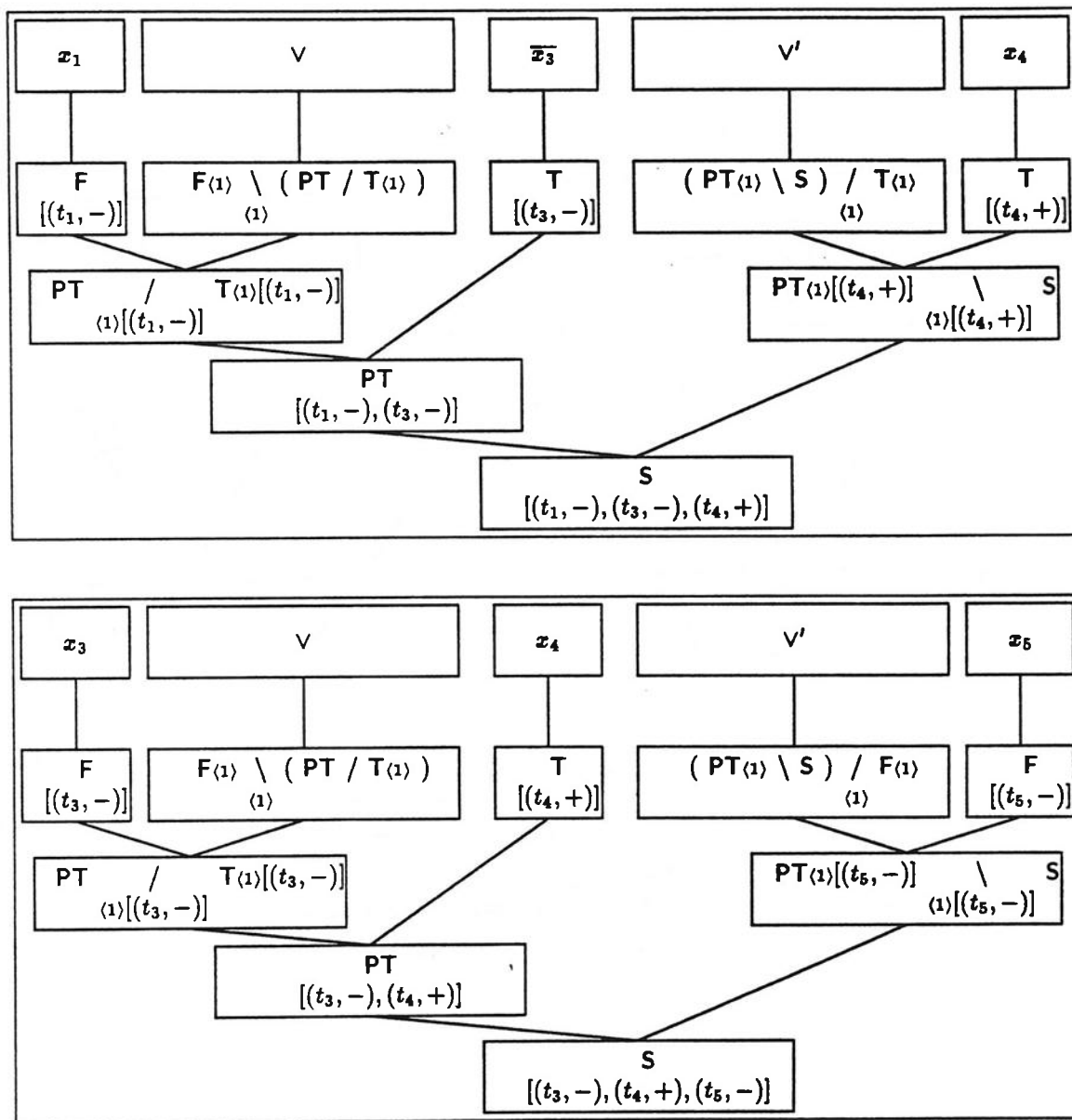


Figure 1: A derivation tree (clauses).

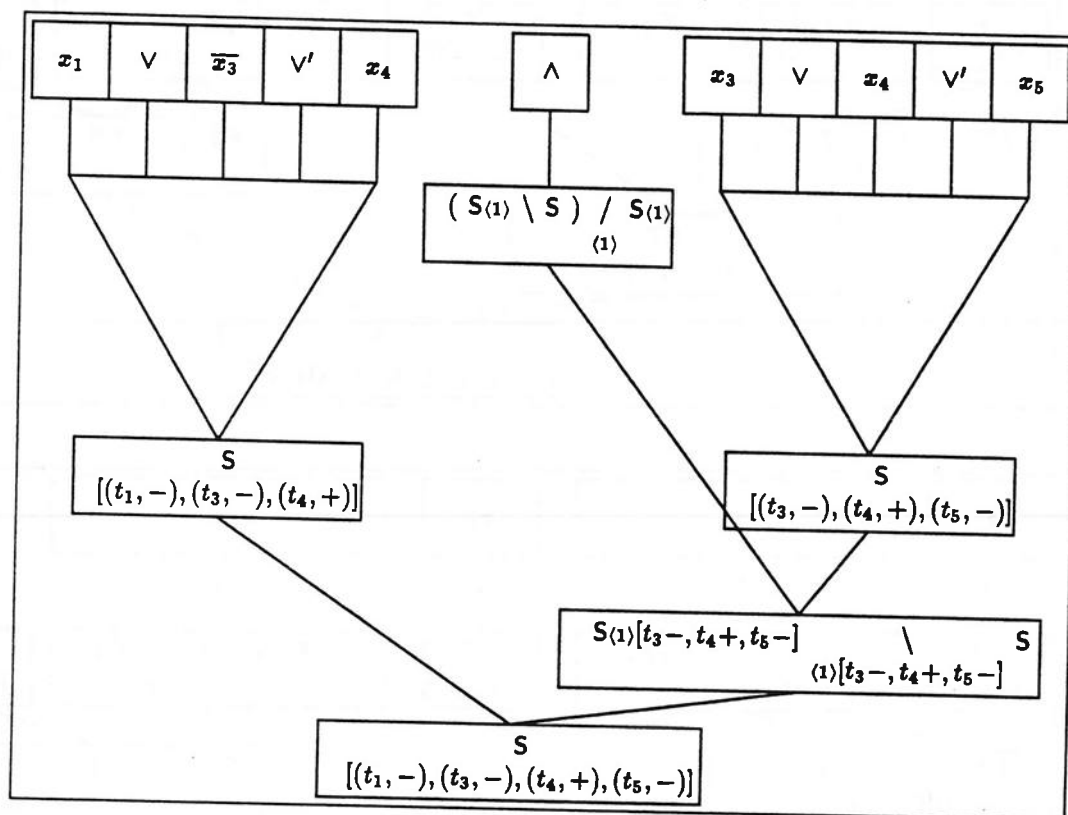


Figure 2: A derivation tree.

fixed part is as discussed above. The variable part is the entries for the positive and negative literals (x_i and $\bar{x}_j$). The indices i and j are in the sets $\{1, 3, 4, 5\}$ and $\{3\}$ respectively.

We have mentioned that the formula f is satisfiable for $g(x_4) = \text{true}$. Here, we show a derivation for the string w_f . In 3SAT we are free to choose the assignment for our variables. Likewise, in the recognition the categories for the literals are free to choose. The categories for the disjunctions and conjunction are set (cf. table 2). A possible assignment is given in table 4. On the left side the words and categories appear that are used in the first clause; on the right, those used in the second clause. At the bottom the conjunction is found. Note that the word $\vee'$ is ambiguous, it is assigned two different meanings in the string.

First clause		Second clause	
Word	Category	Word	Category
x_1 :	$F[(t_1, -)]$	x_3 :	$F[(t_3, -)]$
$\vee$:	$F_{(1)} \setminus (PT / T_{(1)})$	$\vee$:	$F_{(1)} \setminus (PT / T_{(1)})$
	$^{(1)}$		$^{(1)}$
$\bar{x}_3$:	$T[(t_3, -)]$	x_4 :	$T[(t_4, +)]$
$\vee'$:	$(PT_{(1)} \setminus S) / T_{(1)}$	$\vee'$:	$(PT_{(1)} \setminus S) / F_{(1)}$
	$^{(1)}$		$^{(1)}$
x_4 :	$T[(t_4, +)]$	x_5 :	$F[(t_5, -)]$
$\wedge$:		$(S_{(1)} \setminus S) / S_{(1)}$	
		$^{(1)}$	

Table 4: A category assignment.

The corresponding derivation tree is given in figure 1 and 2. Due to typographic limitations the tree is split in two parts. The detailed derivations of the clauses are given in the first figure; the whole derivation tree in the second one.

From the feature structure that is assigned to the root of the derivation tree we can see that the formula f is satisfiable for the assignment $g(x_i) = \begin{cases} \text{true} & i \in \{4\} \\ \text{false} & i \in \{1, 3, 5\} \end{cases}$

3-Satisfiability implies recognition. A solution to 3SAT implies that an assignment exists which is consistent; i.e. each literal is assigned precisely one truth-value. Hence, each variable is assigned exactly one truth-value. Moreover, every clause is satisfiable; i.e. every clause contains a literal with value true.

We have to prove that whenever a 3SAT formula f is satisfiable, the corresponding string w_f is accepted by the grammar G_f . The correspondence between satisfiable formulas f and accepted strings w_f is expressed by the following statements.

1. We are free to choose the categories for the literals. Make these choices: a literal is assigned category T if the literal is true, assign category F otherwise. (Thereby determining the feature structure of that literal).
2. For every clause that contains a category T, assignments to $\vee$ and $\vee'$ are possible such that the clause is of category S (see table 2).
3. Conjunctions are assigned category $(S \setminus S)/S$.

We need not worry about the feature structures in this part of the proof. Because the feature structures correspond to the assignment of the 3SAT formula, an inconsistent feature

structure would express that the assignment has given some variable both the truth-values true and false. We know that each variable of the formula is assigned exactly one truth-value, so all feature structures are consistent.

Because all clauses are satisfiable, we have a literal of category T in each clause (statement (1)). By statement (2), each clause has a derivation to S. Together with statement (3) we know that a derivation for string w_f exists with intermediate stage:

$$\begin{array}{ccccccc} S & (S \setminus S)/S & S & (S \setminus S)/S & \dots & (S \setminus S)/S & S \\ \Delta & | & \Delta & | & & | & \Delta \\ \text{Clause}_1 & \wedge & \text{Clause}_2 & \wedge & \dots & \wedge & \text{Clause}_m \end{array}$$

From this it is obvious that the string w_f has a derivation to S. So, we have proved that we can recognize every 3-satisfiable formula with our grammar. Next, we will prove that a recognized 3SAT formula is satisfiable.

Recognition implies 3-satisfiability. We will prove that if the grammar accepts a translated 3SAT formula f , then the formula f is satisfiable. Because the grammar G_f has accepted the string w_f , a derivation for w_f exists. First, we will show that if all clauses have a derivation to S, then the string w_f has a derivation to S. Next, if a clause has a derivation, it is to S. And finally, that all clauses of the string w_f have a derivation. From these three lemmas it is easy to prove that an assignment exists such that every clause of formula f is true and each variable is assigned exactly one truth-value.

Observe that the only way the grammar is able to alter feature structures is via the functors. Functors take an argument with a feature structure and yield a result category with a feature structure. In general, the application step does not pose any limitations on the feature structures the functor yields. In our grammar, however, no functor removes any feature from the feature structure of its argument, but at most adds new ones.

Lemma 1 *If a string w_f has a derivation and all clauses have a derivation to S, then the string w_f is of category S.*

Proof: All clauses have a derivation to S. Hence the following derivation can be made.

$$\begin{array}{ccccccc} S & (S \setminus S)/S & S & (S \setminus S)/S & \dots & (S \setminus S)/S & S \\ \Delta & | & \Delta & | & & | & \Delta \\ \text{Clause}_1 & \wedge & \text{Clause}_2 & \wedge & \dots & \wedge & \text{Clause}_m \end{array} \quad (1)$$

From here there are numerous many ways to derive category S, where each possibility will assign the same feature structure to the root category. The functors $(S \setminus S)/S$ (conjunctions), pass all feature structures from their arguments up to their result category (by means of the reentrance symbols). Because a derivation exists, the feature structures of the clauses are consistent and all unification steps succeed. This proves the lemma.

Lemma 2 *If a clause has a derivation, it is to category S, and its feature structure is consistent.*

Proof: A clause consists of three literals and two disjunction symbols. The disjunction symbol $\vee$ can be assigned four different functor categories. For the disjunction symbol $\vee'$ three categories can be chosen. This results in 12 pairs of categories for the disjunction symbols of a clause.

The reader can easily verify that only 7 of these 12 combinations result in a derivation, by properly selecting the categories for the three literals of the clause; i.e. according to table 2. Note that the derivation requires the union of the feature structures of the literals to be

consistent. All of these seven derivations lead to category S, and the feature structure of the clause is this union. Therefore the lemma holds.

Lemma 3 *The string w_f has a derivation such that all clauses have a derivation to category S.*

Proof: Although a derivation for string w_f exists, it is not by forehand necessary that clauses form a constituent in the derivation. We will have to show that a conjunction cannot be applied to the any of the elements of a clause.

The only arguments a conjunction can be applied to are of category S. This also holds for the result of this application, the category $S \setminus S$. No intermediate steps in the derivation of a clause exist that result in category S. So, a conjunction symbol cannot be applied to either basic or compound elements of a clause. (Note that also no functor exists that takes the conjunction as an argument.) Therefore, clauses do form constituents in the derivation of string w_f . And by lemma 2 the string has a derivation like (1).

Resuming we have: If a derivation for a string w_f exists each clause is of category S and their feature structures are consistent. Hence each clause contains at least one literal of category T (by table 2). The feature structure in the root category is the unification of all feature structures (which is consistent due to the existence of the derivation). The corresponding formula f is true under the assignment given in the root (where the assignment assigns exactly one truth-value to each variable) and every clause of f contains a true literal.

4.3 Recapitulating

From section 4.2 we have the reduction of 3SAT to the recognition problem of our categorial grammar with features. This reduction takes polynomial time, and in the last two paragraphs we saw that a 3SAT formula f is satisfiable if and only if the string translation w_f of the formula is accepted by the grammar G_f . Therefore, recognition is NP-hard.

In section 4.1 we proved recognition in NP. Hence, we have proven that recognition is NP-complete.

5 Conclusions

The unification process is rather simple. It is in the simple case ordinary set union of sets whose sizes are linearly bounded by the grammar size. Because features are not recursive and feature structures should be consistent sets, the number of feature structures is limited, given the grammar. The complexity of our categorial unification grammar with feature structures is largely caused by the reentrance mechanism and the unbounded set of features.

We have two ways to alter the NP-completeness result of our recognition problem. First we can restrict our recognition problem. Secondly we can restrict our grammar. In the first case, we can think about fixing the grammar, thus fixing the lexicon. Fixing the lexicon means that we fix the number of variables that we can use in our 3SAT formula. Clearly we can come up with a large 3SAT formula that we cannot distinguish with our grammar. This is a trivial solution to the problem from a formal point of view. Linguistically, a thus restricted grammar is unable to handle arbitrarily long sentences, with arbitrarily many anaphora.

Another solution in this line of work is dynamic programming. Dynamic programming divides a problem in a couple of subproblems. It is hoped that these subproblems are easier to solve and that combining these subproblems into the original problem is not too

difficult either. In our recognition problem this means that we could parse substrings, like noun phrases, verb phrases, relative clauses, etc. separately. And thus make use of local information (e.g. noun phrases do not contain verbs).

The other option is to restrict the grammar, instead of posing a different question for the recognition problem. Our reduction showed that if we want the recognition problem of our grammar to be in the class P, we have to place some constraints on the reentrance mechanism and the set of features. This leads to the following constraints:

- fix the number of feature names in advance, use typed feature structures;
- use reentrance symbols for modifiers only, the sources they were invented for in the first place.

We will shortly examine these possibilities.

Fix features. The features are used to encode the assignments that are made for the variables. For each variable in the 3SAT formula we need a new feature name. Fixing the number of features means that we can only encode this information for a limited number of variables. It is obvious that we cannot recognize an 3-satisfiable formula with too many variables.

Restrict reentrance. The reentrance mechanism is invented for modifiers, categories of the form N/N , like adjectives. We could restrict the use of the reentrance mechanism to these kinds of functors, and allow only two occurrences of a reentrance symbol within these functors.

In our reduction conjunctions should pass arbitrary feature structures up to their mother node. So, we have to use the reentrance mechanism for them. However, this requires the category of conjunctions to be the same as that of the modifiers, N/N . This is insufficient, because conjunctions are binary instead of unary operators. Therefore a reduction like the one we gave cannot be created and probably no reduction can. Every reduction will have to check the assignments of all variables. So, all features have to be passed up to the root category. The only way to do this is via the reentrance mechanism, which we showed to be inadequate.

To reduce the complexity of the recognition problem of our categorial unification grammar with its feature structures and its reentrance mechanism, we have to put restrictions on the grammar. The most interesting restriction is limiting the use of the reentrance mechanism. Two other, well-known, grammars that use the reentrance mechanism are HPSG and FUG. In neither system did we find syntactic evidence that the reentrance mechanism is needed for other types than modifiers. Therefore we expect this restriction to be the most promising.

Further Research Two main topics for further research emerge from the previous possibilities. The first topic aims at the set of feature structures, the second at the reentrance mechanism.

We claim that our reduction is still valid if we forbid an infinite set of feature-values, but allow recursive use of features. A recursive feature can take itself as a value. Recursive features are for instance used in PATR-2 (Shieber et al. 1983). We can encode the truth-value of the variables by their unary encoding. For instance instead of $(t_3, +)$ we use $(t, (t, (t, +)))$.

However, this complexity could be reduced if we could borrow a constraint from Lexical Functional Grammar (Bresnan 1982) called functional locality. This constraint only allows comparisons of features up to a fixed depth (suggested depth is 2 or 3). This means in the

case of recursive features that we cannot compare features that contain too many occurrences of itself; e.g. we cannot distinguish variable t_{100} from t_{101} . Thus we have again restricted the number of variables of the 3SAT formulas we can recognize with our grammar. However, it is neither clear what the maximum depth of features should be, nor what the linguistic argument for this depth is.

The other topic in which further research is required is the restriction of the reentrance mechanism. First, it should be shown that restricting the reentrance mechanism to modifiers indeed reduces the complexity to less than NP-complete. Secondly, it should be investigated whether linguistic phenomena exist which need a more extensive use of the reentrance mechanism than modifiers. As a final remark we conjecture that both FUG and HPSG are at least as hard as NP-complete. A slightly modified version of our reduction will work on these grammars as well, unless it can be shown that the use of the reentrance mechanism can be restricted.

References

- Bach, E. 1983a. Generalized Categorical Grammars and the English Auxiliaries. *Linguistic Categories: Auxiliaries and Related Puzzles 2*. eds. F. Heny and B. Richards. Dordrecht-Boston-London: D. Reidel. pp. 101-120.
- Bach, E. 1983b. On the Relationship between Word-Grammar and Phrase-Grammar. *Natural Language and Linguistic Theory* 1:65-89.
- Barton Jr., G. E., Berwick, R. C. and Ristad, E. S. 1987. *Computational Complexity and Natural Language*. Cambridge, MA: MIT Press.
- Bouma, G. 1988. Modifiers and Specifiers in Categorical Unification Grammar. *Linguistics* 26:21-46.
- Bouma, G. 1990. Defaults in Unification Grammar. Paper delivered at Research Institute for Knowledge Systems, Maastricht, the Netherlands, Second European Summer School in Language Logic and Information Katholieke Universiteit Leuven July 30-August 10, 1990.
- Bresnan, J., ed. 1982. *The Mental Representation of Grammatical Relations*. Cambridge, MA: MIT Press.
- Garey, M. R. and Johnson, D. S. 1979. *Computers and Intractability: a Guide to the Theory of NP-Completeness*. San Francisco, CA: W. H. Freeman and Company.
- Hopcroft, J. E. and Ullman, J. P. 1979. *Introduction to Automata Theory, Languages, and Computation*. Reading, MA: Addison Wesley.
- Kay, M. 1985. *Natural Language Parsing*. Cambridge, England: Cambridge Univ. Press.
- Lambek, J. 1988. The Mathematics of Sentence Structure. *Categorical Grammar*. eds. W. Buszkowski, W. Marciszewski and J. van Benthem. Linguistic and Literary Studies in Eastern Europe (LLSEE). Vol. 25. Amsterdam-Philadelphia: John Benjamins Publishing Company. pp. 153-172.
- Moortgat, M. 1988. *Categorical Investigations: Logical and Linguistic Aspects of the Lambek Calculus*. Groningen-Amsterdam Studies in Semantics (GRASS). Vol. 9. Dordrecht-Providence RI: Foris.
- Pollard, C. J. and Sag, I. A. 1988. *Information-Based Syntax and Semantics: Volume 1 - Fundamentals*. CSLI Lecture Notes. Vol. 13. Stanford: CSLI.

- Sells, P. 1985. *Lectures on Contemporary Syntactic Theories*. CSLI Lecture Notes. Vol. 3. Stanford: CSLI.
- Shieber, S., Uszkoreit, H., Pereira, F., Robinson, J. and Tyson, M. 1983. The Formalism and Implementation of PATR-II. *Research on Interactive Acquisition and the Use of Knowledge*. eds. B. Grosz and M. Stickel. Menlo Park, CA: SRI International.
- Shieber, S. M. 1986. *An Introduction to Unification-Based Approaches to Grammar*. CSLI Lecture Notes. Vol. 4. Stanford: CSLI.

EXISTENTIAL SENTENCES AND PREDICATION

Alessandro Zucchi
Stanford University

0. Introduction

I take it that a semantic account the contrast in (1)-(2), often referred to as "definiteness effect" (DE),

- (1)a. ?? There is every student in the garden
- b. ?? There are all students in the garden
- c. ?? There is the student in the garden
- d. ?? There are most students in the garden
- (2)a. There is a student in the garden
- b. There are three students in the garden
- c. There are no students in the garden

must fulfill these tasks:

- (i) it must provide a semantic definition of the class of NPs which are allowed in postverbal position of there-sentences;
- (ii) it must show how DE can be derived from this semantic definition and the interpretation of there-sentences.

Following Milsark (1977), I'll use the term weak determiners and NPs to refer to determiners and NPs allowed in postverbal position of there-sentences, and the term strong determiners and NPs to refer to determiners and NPs which are not allowed in the same position.

1. Two Ways of Testing Analyses of the DE

In this section, I introduce two tests of adequacy for accounts of DE.

FIRST TEST: an adequate account of DE must explain why there-sentences discriminate between complex determiners like the ones underlined in (3)-(4):

- (3) There were either zero or else more than zero students at the party
- (4) ?? There were either all or else not all students at the party

This test poses a challenge for any attempt to derive DE from the model-theoretic properties of the denotation of determiners alone, since the complex determiners in (3)-

(4) seem to denote exactly the same function in every world, the function which maps each set onto the power set of the domain. They seem to denote this function because for each set A it is always the case that any set B either includes or does not include A, and the intersection of A with B is either empty or nonempty. Since the theoretical significance of the contrast in (3)-(4) was pointed out by Keenan (1987), I'll call this first test Keenan's test.

An adequate account of DE must be compatible with the fact that sometimes the postverbal material (or coda) of there-sentences is not analyzed as a bare NP, but as an NP followed by a separate constituent. Evidence for this analysis is provided for example by Barwise and Cooper's (1981) observation that the string two students who object to that enrolled in the course in (7)

- (7) There are two students who object to
 that enrolled in the course

cannot be analyzed as an NP, since it cannot occur in NP-position:

- (8) ?? Two students who object to that
 enrolled in the course came in

Further syntactic evidence that sometimes the coda of there-sentences is not a single constituent is provided by Keenan's observation that the coda cannot be relativized in some cases, as (9b) shows:

- (9)a. Don't worry, John will help himself to
 whatever there is in the fridge
 b. * Don't worry, John will help himself to
 [whatever in the fridge] there is

The contrast in (9) is explained if the PP in the fridge does not form an NP constituent with whatever. Following Keenan, I'll refer schematically to the separate constituent which may follow the NP in there-sentences as "the XP-constituent" (where X can be P, A, etc.). The syntactic evidence I've reviewed provides thus a further adequacy test for accounts of DE:

SECOND TEST: an adequate account of DE must be compatible with the NP-XP analysis of the coda.

2. An Evaluation of Some Accounts of DE

2.1. Milsark (1974, 1977)

According to Milsark, the determiners that occur in postverbal position of there-sentences (weak determiners) are cardinality markers (number words), items whose function is "to express the size of the set of entities denoted by the nominal with which they are construed." Milsark's explanation of why only cardinality markers are allowed in there-sentences is this. While determiners like every, all, most, the are quantificational, cardinal determiners are nonquantificational. Since there be is an existential quantifier, a there-sentence containing a postverbal quantified NP "would have two quantifications on the NP...[which] should certainly be expected to be anomalous." On the other hand, if the postverbal NP is nonquantificational, as in the case of number words, no anomaly arises, since no vacuous quantifier is present.

Milsark's proposal has been observed by various authors¹ to run into problems with sentences like (10)

(10) there is no justice

which are not analyzable in terms of widest scope existential quantification as Milsark would have it. But there is another problem with Milsark's approach. Heim's (1982) analysis of the contrast in (11a)-(12a)

(11)a.* if every/most/all men own a
 donkey, beat it

(12)a. if a man/some man owns a donkey he
 beats it

is based on the assumption that the NPs in (i) are inherently quantificational and the NPs in (ii) are not:

(i) every man, most men, all men

(ii) a man, some man

Notice, however, that NPs of the form [_{NP} no Nⁿ] do not support donkey anaphora and NPs of the form [_{NP} the Nⁿ] do, as (11b)-(12b) show:

(11)b. ?? if no man_i owns a donkey he_i beats it

(12)b. if the farmer_i we met owns a donkey he_i
 beats it

Thus, if Heim is right, no is quantificational and the is not. Thus, Milsark's account predicts incorrectly that no should be barred from postverbal position of there-

sentences, and the should be allowed. In other words, the distinction between quantificational and non-quantificational NPs that emerges from the study of donkey anaphora does not coincide with the distinction between weak and strong NPs.

2.2. Barwise and Cooper (1981)

Barwise and Cooper's account of DE fails both tests of adequacy in 1. Their account predicts incorrectly that both (3) and (4) should be ruled out. Moreover, their semantics for there-sentences is based on the assumption that the coda is a bare NP. I refer the reader to Keenan (1987) for a more detailed discussion of these points.

2.3. Keenan (1987)

Keenan makes the following claim:

KEENAN'S THESIS: The class of determiners allowed in postverbal position of existential there-sentences is the class of existential determiners.

Existential determiners are either basic determiners denoting existential functions or determiners built up from such basic determiners via a certain set of operations (such as Boolean combination, composition with adjectives, and the exception determiner operator). An existential function is a function f from sets to families of sets such that for every set A and B , $B \in f(A)$ iff the universal property $\in f(A \cap B)$. Intuitively, a determiner D denotes an existential function iff sentences of the form in (E) are valid (where N_1 and N_2 are set-denoting expressions):

(E) DN_1 is/are N_2 iff DN_1 who is/are N_2 exist(s)

According to (E), some denotes an existential function since (14) is valid:

(14) some students are vegetarian iff some students that are vegetarian exist

On the other hand, every does not denote an existential function since (15) is not valid:

(15) every student is vegetarian iff every student who is vegetarian exists

Notice that existential determiners are not defined by Keenan as the determiners that denote existential functions. The definition of existential determiner is more elaborate than that. Keenan requires that existential determiners be either syntactically simple determiners denoting existential functions or complex determiners built up from simple determiners that denote existential functions. The need to define existential determiners in this more elaborate way is motivated by contrast (3)-(4). If we define existential determiners simply as determiners that denote existential functions, then both either zero or else more than zero and either all or else not all would be existential determiners since they denote existential functions.² By Keenan's thesis, it would follow that both these determiners should thus be allowed in postverbal position of there-sentences, which is incorrect. According to Keenan's definition of existential determiner, on the other hand, either zero or else more than zero is existential since it is formed from simple existential determiners, but either all or else not all is not existential. Thus, contrast (3)-(4) follows from Keenan's thesis.

The semantics Keenan assumes for there-sentences is this:

KEENAN'S SEMANTICS FOR THERE-SENTENCES

a VP_[there] of the form [be NP XP] is true iff [XP] ∈ [NP]

It follows as a theorem from this semantics that

- (a) a sentence of the form there be NP XP is logically equivalent to a sentence of the form NP XP exist iff the determiner of the NP is always interpreted by an existential function.

I see two main problems with Keenan's account. First of all, Keenan provides no explanation of why only existential NPs are allowed in postverbal position of there-sentences. Keenan's analysis provides no reason to rule out (1) on syntactic grounds and Keenan's semantics provides a perfectly good interpretation for (1). For example, (1a) is assigned the same interpretation as (16):

- (16) every student is in the garden

What Keenan's analysis does is predicting that equivalence (a) holds, but this leaves us with the problem we started out with, namely the problem why (1a) is deviant. Indeed, Keenan fails to give a reason for the ill-formedness of (4) as well. Thus, Keenan's analysis of there-sentences does not pass Keenan's Test.³

There's another problem. By adopting the more elaborate definition of existential determiner reported above Keenan succeeds in characterizing the class of existential determiners in such a way as to classify either zero or else more than zero as existential and either all or else not all as nonexistential. Paired with Keenan's thesis this predicts correctly that either zero or else more than zero should be allowed in there-sentences and either all or else not all should not be. But, even if the right class of NPs is individuated by Keenan's definition, his way of characterizing this class cries out for an explanation: why is the class of complex determiners allowed in there-sentences individuated on the basis of the denotations of the simple determiners that make them up rather than on the basis of the denotations of the complex determiners?

2.4. Higginbotham (1987)

Higginbotham claims that the NPs allowed in postverbal position of there-sentences are the NPs whose determiners are symmetric. DE is derived from the assumption that the semantics of there-sentences requires determiners to have an absolute interpretation:

$[D]([N]) = 1$ iff $|N| \in$ the (nonempty) set K of thresholds for D

A determiner can get an absolute interpretation only if it has a nonempty set of thresholds, where thresholds are defined in this way:

A threshold for a determiner D is a cardinal number n such that for all models M and for every set A and B , if $|A \cap B| = n$, then $[D]^M(A, B) = 1$.

According to this definition, every cannot have an absolute interpretation since there are no thresholds for every, but some can have an absolute interpretation since it has a nonempty set of thresholds. The ill-formedness of (1) is thus claimed to follow from the fact (1) is uninterpretable, since the determiners in (1) lack a nonempty set of thresholds.

Higginbotham's analysis fails both Keenan's test and the syntactic test of adequacy. It fails the syntactic test, since his semantics for there-sentences assumes that the coda of there-sentences is a bare NP. It fails Keenan's test, since both determiners in (3)-(4) have nonempty sets of thresholds (every cardinal number is a

threshold) and thus are predicted to be allowed in there-sentences. Notice, moreover, that the identification of weak determiners with the symmetric ones is incorrect, since it is a theorem that, if a determiner D denotes an existential function, then D is symmetric.⁴ Thus, Higginbotham predicts that, if a determiner denotes an existential function, it should be allowed in there-sentences. But this is false, as (3) and (4) show.

3. The Ingredients of Definiteness

The name DE for the contrast in (1)-(2) originates from the hypothesis that NPs barred from postverbal position of there-sentences are, in some sense, definite. As Milsark (1977) observed, however, the attempt to derive the distribution of NPs in there-sentences from a prohibition against definite NPs runs into the difficulty of explaining in which sense NPs barred from there-sentences are all definite. For example, in which sense are universally quantified NPs like every student definite? What is it that these NPs have in common with NPs traditionally regarded as definite, like the students? This difficulty induced Milsark to give up the characterization strong NPs as definite and to attempt reconstructing the strong/weak distinction through the notions quantificational/nonquantificational. In this section, I want to suggest that Milsark's decision to abandon the hypothesis that strong NPs are, in some sense, definite was premature. Heim (1982) has suggested that there are two related, but not interdeducible, components to definiteness, the Familiarity Condition and the Descriptive Content Condition. In Heim's system, universally quantified NPs turn out to be indefinite, since they fail to meet the Familiarity Condition, namely they introduce novel referents. The assumption that universally quantified NPs introduce novel discourse referents is needed to prevent the quantifier everyone from binding the pronoun her in (17):

(17) everyone who gave a cat to her got it back

There is, however, a property of definite NPs which is shared by universally quantified NPs and, more generally, by NPs barred from postverbal position of there-sentences. The Descriptive Content Condition requires the context of utterance of NPs of the form in (a)

(a) $[_{NPi} \text{ DET}^{[+def]} N^n]$

to entail that the set denoted by N^n include the individual assigned to index i by the variable assignment

g.⁵ For example, the context of utterance of [_{NPI} the cat] must entail that the set of cats include the individual assigned to *i* by the variable assignment *g*. Thus, DCC entails that definite NPs have the characteristic property (P):

- (P) The set denoted by the highest nonmaximal projection of N is presupposed to be nonempty.

Examples (18)-(19) from Lumsden (1988) show that NPs barred from postverbal position of there-sentences share property (P) with definite NPs:⁶

- (18) ?? There is/are the/every/all/most/both/
all the mistakes
(19) If you find the/every/all/most/both
mistake(s), I'll give you a fine
reward

The interpretation of the strong NPs in (19) is characterized by the expectation of the hearer that the set of mistakes is nonempty. However, no such presupposition needs to arise for NPs allowed in postverbal position of there-sentences, as (20)-(21) show:

- (20) There is/are a/some/three/zero/many/a
lot of/no mistakes
(21) If you find a/some/three/zero/
many/a lot of/no mistake(s), I'll
give you a fine reward

The data in (18)-(21) suggest thus this answer to Milsark's question:

- (22) NPs barred from postverbal position of there-sentences, unlike NPs allowed in the same position, share with definite NPs the presupposition that the set denoted by the noun, more precisely by N^n , is nonempty.⁷

There is further evidence, in addition to the one provided by Lumsden in (18)-(21), that the presuppositional characterization of strong NPs is on the right track. Let's consider again the determiners either all or else not all and either zero or else more than zero. As we saw, only the last determiner is able to occur in there-sentences. If the presuppositional hypothesis is correct, we should expect either all or else not all, unlike either zero or else more than zero,

to presuppose that the set denoted by the the noun is not empty. The prediction is borne out. Intuitively, (24) carries the presupposition that the set of mistakes is nonempty, but (23) does not:

- (23) if you find either zero or else more
than zero mistakes, I'll give you a
fine reward
- (24) if you find either all or else not
all mistakes, I'll give you a fine
reward

Notice that (22), besides identifying correctly the class of NPs barred from postverbal position of there-sentences, also explains (what was missing from Keenan's account) why the acceptability of complex determiners in there-sentences depends on the basic determiners that make them up. Given the plausible hypothesis that presuppositions are projected from lexical items, complex determiners should carry the same presuppositions as the basic determiners that make them up.

4. Some Unsuccessful Attempts to Derive DE from the Presuppositional Characterization of Strong NPs

I still haven't explained why NPs carrying the presupposition that the set denoted by the noun is nonempty should be barred from the postverbal position of there-sentences. What is it that goes wrong? One possible explanation of DE based on the presuppositional characterization of strong NPs is this. Consider (25) and (26):

- (25) There is a king of France
- (26) ?? There is the king of France

Sentence (25) asserts the existence of a king of France. On the basis of (25), one might conjecture that the role of there-sentences is that of predicating the property of existing of the denotation of the postverbal NP. In a generalized quantifier analysis, this amounts to saying that there-sentences assert that the domain of discourse belongs to the set of sets that contain a king of France as a member. Since the NP "a king of France" does not presuppose the existence of a king of France, then an appropriate context of utterance for (25) does not have to entail that the domain of discourse is a member of the denotation of the NP. Thus, an appropriate context of utterance for (25) need not entail what (25) asserts. The postverbal NP in (26), however, presupposes the existence of a king of France. Thus, what (26) asserts, namely that the domain of discourse is a member of the family of sets

denoted by the NP, must be already entailed by any context appropriate for the utterance of (26). Thus, (26) asserts nothing more than what is presupposed. This is why (26) is deviant.⁸ There are some problems with this explanation. Consider (27)-(28):

- (27) the king of France is a king of
 France
- (28) the king of France exists

Examples (27)-(28) are cases in which the presupposition of the subject NP entails the content of the whole assertion. Yet, (27)-(28) are more acceptable than (29):

- (29) ?? there is the king of France

According to the analysis of DE we are considering, however, (27)-(28) should be ruled out exactly for the same reason (29) is ruled out.

The account I'm considering raises also some questions in connection with there-sentences whose coda is not analyzed as a bare NP. For example, the material following the verb in (30) must be analyzed as in (31):

- (30) ?? there is the student who objects
 to that enrolled in the course
- (31) ?? there is [_{NP}the student who objects
 to that] [_{XP}enrolled in the course]

The proposed analysis of DE work doesn't rule out (30) because it is uninterpretable. What should be wrong with (30) is that the content of (30) is entailed by the presupposition of the postverbal NP. But what is the content of (30)? If we assume with Keenan that the XP is predicated of the postverbal NP, then the XP enrolled in the course should be predicated of the NP the student who objects to that, and (30) should have the same content as (32):

- (32) the student who objects to that
 is enrolled in the course

Thus, in order for the presupposition of the postverbal NP in (30) to entail the content of (30), the NP the student who objects to that must presuppose the content of (32). But how can this be, given that the phrase enrolled in the course is not even part of the NP? If, on the other hand, we do not assume that the XP-constituent is predicated of the postverbal NP, then one is entitled

to ask: what is the semantic role of the XP in there-sentences?

The first problem I pointed out shows that DE cannot be derived simply from the presupposition of strong NPs together with general conversational principles such as "Do not assert what is already presupposed." Sentences (27)-(29) suggest that there must be something specific to there-sentences which is in conflict with the presupposition of strong NPs. Grice (1975) introduced the notion of conventional implicature to denote those implicatures which have been grammaticized, namely implicatures that are part of the conditions of use of lexical items and syntactic constructions, and are not derivable from the truth-conditional meanings of these items and constructions together with general conversational maxims. The presupposition of strong NPs described in (22) may be seen as a conventional implicature in Grice's sense. Conventional implicatures carried by a particular lexical item or syntactic construction are often treated as felicity conditions for the use of that item or construction, namely as requirements imposed by that item or construction on contexts appropriate for its use. An expression α is assumed to denote in a context c only if c meets the felicity conditions of α . A possible hypothesis about DE is that the deviant character of strong NPs in there-sentences is due to a conflict between the felicity conditions of there-sentences and the presupposition (that is, the felicity conditions) of strong NPs. Suppose that there-constructions are characterized by the felicity conditions in (33):

(33) FELICITY CONDITIONS OF THERE-SENTENCES
(tentative formulation)

There-sentences are felicitous only in contexts which entail neither that the set denoted by the N^n of the postverbal NP is empty nor that it is nonempty.

The ill-formedness of (29)

(29) ?? there is the king of France

may then be explained as follows. Strong NPs require that the context entail the denotation of the noun to be nonempty. However, a context which satisfies this requirement cannot meet the felicity conditions of there-sentences in (33). Thus, there is no appropriate context for a sentence like (29), the felicity conditions of strong NPs and of there-sentences are in conflict with

one another. This is why (29) is deviant. Notice that this proposal avoids the difficulty posed by (27)-(28), since the deviant character of (29) is traced back to a grammaticized presupposition specific to there-constructions.

This analysis of DE still runs into problems with there-sentences whose coda cannot be analyzed as a bare NP. Consider the conversation in (34):

- (34) A: The students who object to that will
come to the meeting tomorrow.
B: There are some students who object to
that enrolled in the course

Imagine that, by the time B utters his line, the assertion made by A is incorporated in the common ground, namely is assumed to be true by the participants in the conversation. In this case, the context of utterance for B's sentence entails that the set of students who object to that is nonempty. Since the phrase enrolled in the course does not form a constituent with the complex noun students who object to that, we are forced to analyze B's sentence as in (35):

- (35) there are [_{NP}some [_{NN}students who
object to that] [_{XP}enrolled in the
course]

The felicity conditions for there-sentences in (33) require thus that the context of utterance of B's sentence do not entail the set of students who object to that to be nonempty. Since the context of utterance of B's sentence does entail this set to be nonempty, we should expect B's sentence to be infelicitous in this conversation. Yet, no suggestion of infelicity arises.

Conversation (34) suggests that the felicity conditions for there-sentences proposed in (33) are incorrect. Indeed, in view of (34), it would seem that the felicity conditions in (36) are a more suitable candidate as felicity conditions of there-sentences:

- (36) FELICITY CONDITIONS OF THERE-SENTENCES
(revised formulation)

There-sentences are felicitous only in contexts which entail neither that the intersection of the set denoted by the Nⁿ of the postverbal NP with the set denoted by the XP is empty nor that it is nonempty.

Conversation (34) does not pose a problem for the revised felicity conditions in (36), since the context of utterance of B's sentence entails only that the set of students who object to that is nonempty. Notice, however, that, while the revised formulation of the felicity conditions of there-sentences in (36) avoids the problem posed by (34), it no longer allows us to derive DE. Consider, (1a) again:

(1)a. ?? there is every student in the garden

According to the account of DE we are considering, (1a) should be deviant because of a conflict between the felicity conditions of strong NPs and the felicity conditions of there-sentences. Since (1a) allows the analysis in (1a')

(1)a' ?? there is [_{NP} every student] [_{xP} in the garden],

the revised conditions in (36) require that the context be neutral as to the emptiness of the set of students in the garden. This is compatible with the requirement imposed the felicity conditions of the strong NP every student that the context of utterance of (1a) entail that the set of students be nonempty. Thus, no conflict arises for (1a), and (1a) is incorrectly predicted to be acceptable.

Finally, the second way of deriving DE I considered runs also into problems with there-sentences whose coda is a bare NP. Take (37):

(37) There are some mistakes. Indeed, there are five mistakes.

The context of utterance of the second sentence in (37) entails that the set of mistakes is nonempty, since the first sentence in the discourse has informed us of that. Thus, this account of DE predicts incorrectly that the second sentence in (37) should not be appropriate once the first sentence has been uttered.

5. Deriving DE

The discussion in the previous section shows that deriving the distribution of NPs in there-sentences from the presupposition of strong NPs is no simple matter. Should we give up attempting to produce this derivation? The presuppositional characterization of strong NPs gives a rather striking result in connection with Keenan's

Test. The attempt is worth pursuing more. I am going to argue that the problems I described for deriving DE from the presuppositional definition of strong NPs arise from an inadequate understanding of the semantics of there-sentences, in particular, of the semantic role of the XP-coda. Before I do that, however, I need to take a detour.

5.1. Contextual Domains

It has often been observed that quantified sentences of natural languages are understood as quantifying over contextually furnished subsets of the domain of discourse. For example, sentence (38)

(38) every student was at the party

is naturally understood as saying that every student belonging to a certain contextually furnished set was at the party and not as saying that every student in the universe was at the party. The semantic framework I assume is the one developed in Heim (1982:Ch.2, 1987). In Heim (1982), NPs are obligatorily adjoined to S by an NP-raising rule. In order to account for narrow scope readings, Heim (1987) suggests that NPs may also be interpreted in situ. This means that LFs may now contain chunks of the form (i)-(ii) (where the NP is not a trace):

(i) $[_{VP} V NP]$ (ii) $[_{NP} DET N^n]$

The interpretation rules for (i)-(ii) are the following:

$$\begin{aligned} S_{VP} \cdot [[_{VP} V NP]]^g &= [V]^g_{M,c}([NP]^g_{M,c}) \\ S_{NP} \cdot [[_{NP} DET N^n]]^g_{M,c} &= [DET]^g_{M,c}([N]^g_{M,c}) \end{aligned}$$

In these rules, as in the other LF-interpretation rules, the interpretation function $[]$ is relative to a context, a model, and a variable assignment. We may think of a context c as an n-tuple including, among other things, a set of propositions representing the common ground of c, and a function that assigns a value to free variables in LF-structures. The observation that quantified sentences are always evaluated with respect to a contextually furnished subset of the universe of discourse may be incorporated in this way. Let's assume that a context specifies also a subset $D(c)$ of the universe of discourse. The desired restriction on the interpretation of (38) may now fall out of the assumption that in any given context c , the interpretation function $[]^g_{M,c}$ assigns as denotations sets that are built out of $D(c)$. This means that the denotation of the noun student and of the NP every student in a context c will be specified in this

way:

$$\begin{aligned} [\text{student}]_{M,c} &= \{x \subseteq E \mid x \text{ is a man and } x \in D(c)\} \\ [\text{every student}]_{M,c} &= [\text{every}]_{M,c}^g([\text{student}]_{M,c}^g) = \\ &= \{X \subseteq E \mid [\text{student}]_{M,c} \subseteq X\} \end{aligned}$$

It follows that (38) will be true in a context c just in case the set of entities at the party includes the set of students in the contextually furnished set $D(c)$.

Notice that the presupposition associated with strong NPs is also understood as relative to the contextually furnished domain $D(c)$. Suppose I want you to proofread a paper of mine and I tell you:

(19') If you find every mistake, I'll give you a fine reward

When I say "every mistake" in this context, I mean every mistake in my paper. Thus, the contextually furnished domain relevant for interpreting (19') will not contain mistakes that are not in my paper. The presupposition carried by the strong NP every mistake is also about this contextually furnished set: in the context at hand, (19') presupposes that the set of mistakes in the paper is not empty, and not simply that somewhere, in some paper other than mine, there are mistakes. This fact follows from the semantics adopted here and from the presupposition of strong NPs in (22): in the context we described, the head noun mistakes denotes the set of mistakes in my paper. Thus, the presupposition carried by the strong NP every mistake will be the presupposition that the set of mistakes in my paper is nonempty.

5.2. Intuitive Sketch of the Derivation

In the previous discussion, I suggested the felicity conditions for there-sentences reported here in 1:

1. FELICITY CONDITIONS OF THERE-SENTENCES
There-sentences are felicitous only in contexts which entail neither that the intersection of the set denoted by the N^n of the postverbal NP with the set denoted by the XP is empty nor that it is nonempty.

Whenever no XP constituent is present, let's assume that the common ground must be neutral about the nonemptiness of the intersection of the denotation of the Noun of the postverbal NP with the domain of discourse. The presupposition of strong NPs in (22) may now be expressed in terms of felicity conditions as in 2:

2. FELICITY CONDITIONS OF STRONG NPs
 NPs whose determiner is every/the/both/all/etc. are felicitous only in contexts whose common grounds entail that the denotation of N^n is nonempty.

I noticed before that 1-2 are not sufficient to derive DE. For LF (1a')

(1)a' ?? there is [_{NP} every [_{Nn} student]] [_{PP} in the garden]

no conflict seems to arise between the felicity conditions of there-sentences in 1 and the felicity conditions of strong NPs in 2. The felicity conditions of there-sentences require that the context of utterance of (1a') do not entail that the set of students in the garden is nonempty. This is compatible with the requirement imposed by the strong NP every student that the context entail that the set of students is nonempty. Thus, 1-2 seem to provide no clue as to what causes the deviancy of (1a'). In drawing this conclusion, however, I have implicitly assumed that the XP constituent plays no role in the interpretation of the postverbal NP. Indeed, if the XP-constituent in (1a') is predicated of the postverbal NP, as in (39)

(39) every student is in the garden,

there seems to be no reason why 1 and 2 should be in conflict. In (1a'), however, the XP in the garden is not in canonic predicative position as in (39). This lends some plausibility to the hypothesis that, from a semantic standpoint, the role of the XP-constituent is not that of being predicated of the postverbal NP. I suggest that the semantic role of the XP is this:

3. SEMANTIC ROLE OF XP-CODA
 The denotation of XP provides the contextual domain to interpret the postverbal NP.

Given the felicity conditions of strong NPs in 2, assumption 3 will have the consequence that (1a') is appropriate only in contexts in which the set of students in the garden is assumed to be nonempty. But the felicity-conditions of there-sentences require that an appropriate context for (1a') do not entail that the set of students in the garden is nonempty. Thus, (1a') imposes contradictory requirements on what an appropriate context for its utterance should be.

I still need to deal with the problem posed by (37):

- (37) There are some mistakes. Indeed,
 there are five mistakes.

The felicity conditions for there-sentences I assumed require the context of utterance of there-sentences whose coda is a bare NP to be neutral as to the emptiness of the set denoted by the noun. Applied to (40)

- (40) there are five mistakes

this seems to imply that an appropriate context of utterance for (40) should be neutral about the existence of mistakes. But discourse (37) is felicitous, despite the fact that the set of mistakes is entailed to be nonempty by the context at the time when (40) is uttered. The source of the problem here is the assumption that number words like five are located in specifier position of NPs and thus lie outside the highest nonmaximal projection of N (N^n). Milsark (1977) pointed out that number words act semantically like cardinality predicates on the set denoted by the noun with which they combine. In terms of recent theories of plurals, we may see number words like five, for example, as combining with a set A to yield the subset of A whose members are pluralities made up by five atoms. Besides allowing an adjectival interpretation in this sense, number words have been noticed to display adjectival behavior from a syntactic point of view as well, as shown by (41)-(43):

- (41) the five boys left
(42) those five boys left
(43) the boys are five

These facts have prompted some authors to suggest that number words do not occupy the specifier position of NP and that they are dominated instead by a nonmaximal projection of N, the way adjectives are.⁹ According to this analysis, cardinal NPs like five students are NPs with an empty specifier position with the number word five dominated by N^n :

- (i) [_{NPi} [_{SPEC} $\emptyset$] [_{Nn} five students]]

Notice that, since this analysis claims that the specifier position of cardinal NPs is empty, it predicts that cardinal NPs have no quantificational power of their own. If Heim's (1982) account of donkey anaphora is correct, we should thus expect that cardinal NPs, like indefinite NPs, whose specifier is semantically inert for

the purposes of interpretation, should support donkey anaphora. As Reinhart (1987) has observed, this prediction is borne out, as you can see from (44):

- (44) every vampire who invited three
guests for dinner was through with
them by midnight

Under this analysis of cardinal NPs, moreover, discourse (37) is no longer in conflict with the felicity conditions for there-sentences in 1. These felicity conditions require now the context of utterance of the second sentence in (37) to be neutral about the existence of pluralities consisting of five mistakes. This requirement is met, since the first sentence entails only that there are mistakes.

5.3. Formal Implementation

Following Heim (1987), I assume that postverbal NPs in there-sentences are interpreted in situ at LF. This assumption is supported by the lack of a wide scope reading of the indefinite in (45) (contrasting with the presence of a wide scope reading in (46)):

- (45) Ralph believes that someone is spying
on him
(46) Ralph believes that there is someone
spying on him

Since indefinite NPs are interpreted in situ, namely in the position they are found at S-structure, we need to provide a higher type denotation for them. Heim (1987) suggests interpreting indefinite NPs like $[_{NP_i} a \text{ cat}]$ as generalized quantifiers of the kind in (47):

- (47) $[[_{NP_i} a \text{ cat}]]_{M,c} = \{X \subseteq E \mid g(x_i) \in [cat]_{M,c} \ \& \ g(x_i) \in X\}$

This interpretation is consistent with Heim's (1982) claim that indefinite NPs like a cat have no quantificational force of their own. We may generalize Heim's suggestion in (47) by adopting the rule in (48) for CN-headed NPs whose SPEC is empty or invisible for the purposes of semantic interpretation:¹⁰

- (48) $[[_{NP_i} [_{SPEC} \emptyset [N^n]]]_{M,c} = \{X \subseteq E \mid g(x_i) \in ([N^n]_{M,c} \cap X)\}$

I assume an operation of existential closure adjoining an unselective existential quantifier to VP. The LF

corresponding to (2a) (in the NP-XP analysis) will thus be (2a'):

(2a') $[_s [_{NP} \text{there}] [_{VP} \exists_i [_{VP[there]} \text{is} [_{NP} \text{a student}] [_{XP} \text{in the garden}]]]]]$

In the LF for (1a),

(1a) ?? there is every student in the garden,

where the postverbal NP is quantificational, the unselective existential quantifier will be vacuous for the purposes of interpretation. The felicity conditions of strong NPs and of there-sentences are reformulated in a slightly more formal fashion below:

FC. ^{strong NPs}
 $[[_{NP_i} [_{DET} \text{every/the/both/all/etc.}] N^n]]_{M,c}^g$ is defined only if, prior to the utterance of NP_i , $cg(c)$ entails that $[N^n]_{M,c}^g \neq \emptyset$

FC1. ^{there}
 $[[_{VP[there]} \text{be} [_{NP} (D) N^n] XP]_{M,c}^g$ is defined only if neither $cg(c)$ entails $[XP]_{M,c}^g \cap [N^n]_{M,c}^g \neq \emptyset$ nor $g(c)$ entails $[XP]_{M,c}^g \cap [N^n]_{M,c}^g$

FC2. ^{there}
 $[[_{VP[there]} \text{be} [_{NP} (D) N^n]]]_{M,c}^g$ is defined only if neither $cg(c)$ entails $E \cap [N^n]_{M,c}^g \neq \emptyset$ nor $cg(c)$ entails $E \cap [N^n]_{M,c}^g = \emptyset$

I assume with Keenan that the bearer of a truth-value in there-sentences is the VP. Let c and c' be two contexts such that $D(c') \subseteq D(c)$. Let's say context c' is a contraction of context c iff the variable assignment $g(c')$ coincides with $g(c)$ for every individual in $D(c')$ and for every persistent proposition p , if $cg(c')$ entails p , then $g(c)$ entails p . The truth-conditions of $VP_{[there]}$ are given below:

S1. ^{$VP_{[there]}$}
 $[[_{VP[there]} \text{be} NP_i XP]_{M,c}^g = 1$ just in case $E \in [NP_i]_{M,c'}^g$, where $D(c') = [XP]_{M,c}^g$ and c' is a contraction of c

S2. ^{$VP_{[there]}$}
 $[[_{VP[there]} \text{be} NP_i]_{M,c}^g = 1$ just in case $E \in [NP_i]_{M,c}^g$

As announced in 5.2., the crucial part of these truth-

conditions is that the set denoted by the XP provides the context set for interpreting the postverbal NP. According to these truth-conditions, LF (1a')

(1a') $[_s [_{NP} \text{there}] [_{VP} [_{there}] \text{is} [_{NP} \text{every student}]] [_{XP} \text{in the garden}]]]$

will denote in M, c if and only if NP_i denotes in M, c'. But according to the felicity conditions for strong NPs, NP_i denotes in M, c' iff the common ground of c' entails that the set of students in the garden is nonempty. Given the way c' is defined (c' is a contraction of c), if c' entails that the set of students in the garden is nonempty, c also does. Thus, (1a') will denote in M, c only if the set of students in the garden is entailed to be nonempty by the common ground of c. But the felicity conditions of VP_{there} require c not to have a such an entailment. Thus, a conflict is generated by the felicity conditions of VP_{there} and the felicity conditions projected by the strong NP in (1a').

ACKNOWLEDGEMENTS: I thank Cleo Condoravdi, Geoff Nunberg, Barbara Partee, and my Rochester students for valuable comments and discussions.

Notes

1. See von Stechow (1980), Heim (1987), Higginbotham (1987).
2. We can see that both these determiners denote an existential function in this way. We know that either all or else not all denotes the function which maps each set onto the power set of the domain. Thus, for every set A and B, $B \in [\text{either all or else not all}](A)$, and $B \in [\text{either all or else not all}](A \cap B)$. Thus, either all or else not all denotes an existential function. But either zero or else more than zero and either all or else not all denote the same function in all models. Thus, either zero or else more than zero denotes an existential function as well.
3. Keenan is aware that he is giving no explanation of the contrast in (1)-(2). See p. 299 of Keenan's paper on this.
4. See Keenan (1987:317).
5. The Descriptive Content Condition accounts for the intuition that definite NPs presuppose the existence of an individual satisfying the descriptive content of the NP.
6. This presupposition of strong NPs was originally pointed by Barwise & Cooper. Barwise and Cooper's derivation of DE, however, is based on a different property of strong NPs.

7. This hypothesis has also been suggested in de Jong (1987), de Jong & Verkuyl (1985).

8. As far as I understand, this explanation was suggested, among others, by Lumsden.

9. Reinhart (1987) has made the more radical suggestion that all weak NPs are dominated by N-double bar.

10. In this analysis, NPs like three students have thus the following interpretation:

$[_{NP_i} \text{three students}_i]_{M,c} = \{X \subseteq E \mid g(x_i) \in [\text{students}]_{M,c} \ \& \ g(x_i) \text{ is made up by three atoms} \ \& \ g(x_i) \in X\}$

References

- Barwise, J. and Cooper, R.: 1981, Generalized Quantifiers and Natural Language, Linguistics and Philosophy, 4(1), 159-219
- Heim, I.: 1982, The Semantics of Definite and Indefinite NPs, PhD Dissertation, UMass, Amherst
- Heim, I.: 1987, Where Does the Definiteness Restriction Apply? Evidence from the Definiteness of Variables, in E. J. Reuland and A. B. ter Meulen (eds.)
- Higginbotham, J.: 1987, Indefiniteness and Predication, in E. J. Reuland and A. B. ter Meulen (eds.)
- Jong, F.M.G. de: 1987, The Compositional Nature of (In)definiteness, in E. J. Reuland and A. B. ter Meulen (eds.)
- Keenan, E. L.: 1987, On the Semantic Definition of 'Indefinite NP', in E. J. Reuland and A. B. ter Meulen (eds.)
- Lumsden, M.: 1988, Existential Sentences, Croom Helm
- Milsark, G.: 1977, Toward an Explanation of Certain Peculiarities in the Existential Construction in English, Linguistic Analysis, 3, 1-30
- Reinhart, T.: 1987, Specifier and Operator Binding, in E. J. Reuland and A. B. ter Meulen (eds.)
- Reuland, E. J. and ter Meulen, A. J. (eds.): 1987, The Representation of (In)definiteness, The MIT Press, Cambridge, Mass.
- Stechow A. v.: 1980, Modification of NPs: A Challenge for Compositional Semantics, Theoretical Linguistics, 7, 57-109

Adresses of the authors

Pieter Adriaans

Syllogic B.V.

Postbus 26, 3990 DA Houten, The Netherlands

pieter@syllogic.nl

David Beaver

Centre for Cognitive Science, University of Edinburgh

2 Buccleuch Place, Edinburgh EH8 9LW, Scotland

Department of Philosophy, University of Amsterdam

Nieuwe Doelenstraat 15, 1012 CP Amsterdam, The Netherlands

dib@cogsci.ed.ac.uk & dib@illc.uva.nl

Filippo Beghelli

Department of Linguistics, UCLA

405 Hilgard Avenue, Los Angeles CA 90024, United States

izzyau3@mvs.oac.ucla.edu

Patrick Blackburn

Department of Philosophy, University of Utrecht

Heidelberglaan 8, 3584 CS Utrecht, The Netherlands

patrick@phil.ruu.nl

Michael Böttner

I.M.S.S.S. Ventura Hall, Stanford University

Stanford CA 94305-4115, United States

boettner@csli.stanford.edu

Paul Dekker

ILLC/Department of Philosophy, University of Amsterdam

Nieuwe Doelenstraat 15, 1012 CP Amsterdam, The Netherlands

dekker@illc.uva.nl

Peter van Emde Boas

ILLC/Department of Mathematics and Computer Science, University of Amsterdam

Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands

peter@fwi.uva.nl

Jan van Eijck

CWI, Amsterdam, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

OTS, Utrecht, Trans 10, 3512 JK Utrecht, The Netherlands

jve@cwi.nl

Frank van Eynde

Departement Linguistiek, Katholieke Universiteit Leuven
Maria Theresiastraat 21, 3000 Leuven, Belgium
frank@et.kuleuven.ac.be

Dov Gabbay

Department of Computer Science, Imperial College
180 Queen's Gate, London SW7 2BZ, England
dg@doc.ic.ac.uk

Jonathan Ginzburg

Centre for Cognitive Science, University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW, Scotland
ginzburg@cogsci.ed.ac.uk

Herman Hendriks

ILLC/Department of Philosophy, University of Amsterdam
Nieuwe Doelenstraat 15, 1012 CP Amsterdam, The Netherlands
hendriks@illc.uva.nl

Petra Hendriks

Department of Dutch, University of Groningen
Oude Kijk in 't Jatstraat 26, 9712 EK Groningen, The Netherlands
hendriks@let.rug.nl

Mark Hepple

Computer Laboratory, University of Cambridge
New Museums Site Pembroke Street, Cambridge CB2 3QG, England
mrh@cl.cam.ac.uk

Zhisheng Huang

Center for Computer Science in Organization and Management (CCSOM),
University of Amsterdam
Oude Turfmarkt 151, 1012 GC Amsterdam, The Netherlands

Pauline Jacobson

Department of Cognitive and Linguistic Sciences, Brown University
Box 1978, Providence RI 02912, United States
li700013@brownvm.brown.edu

Theo M.V. Janssen

Department of Mathematics and Computer Science, University of Amsterdam
Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands
`theo@fwi.uva.nl`

Edward L. Keenan

Department of Linguistics, UCLA
405 Hilgard Avenue, Los Angeles CA 90024, United States
`iyw8elk@mvs.oac.ucla.edu`

Ruth Kempson

Department of Computer Science, Imperial College
180 Queen's Gate, London SW7 2BZ, England
`rmk@doc.ic.ac.uk`

Jean-Yves Lerner

Department of Computational Linguistics, University of Saarbrücken
W-6600 Saarbrücken, Germany
`lerner@coli.uni-sb.de`

Lin Liang

Ventura Hall, Stanford University
Stanford CA 94305-4115, United States
`liang@sunrise.stanford.edu`

Márta Maleczki

Department of General and Applied Linguistics
Attila József University Hungary
Egyetem u. 2, Szeged H-6723, Hungary
`h3738mal@ella.hu`

J.-J.Ch. Meyer

Free University Amsterdam & University of Nijmegen
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
`jules@cs.vu.nl`

Wilfried P.M. Meyer Viol

OTS Utrecht, Trans 10, 3512 JK Utrecht, The Netherlands
CWI Amsterdam, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
`wilfried@cwi.nl`

Michael Moortgat

OTS Utrecht, Trans 10, 3512 JK Utrecht, The Netherlands
CWI Amsterdam, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
Michael.Moortgat@let.ruu.nl

John Nerbonne

Deutsches Forschungszentrum für Künstliche Intelligenz
Saarbrücken
nerbonne@dfki.uni-sb.de

Valeria de Paiva

Computer Laboratory, University of Cambridge
New Museums Site Pembroke Street, Cambridge CB2 3QG, England
vcvp@cl.cam.ac.uk

Jaroslav Peregrin

Institute for Theoretical and Computational Linguistics
Faculty of Philosophy, Charles' University
Celetná 13, 110 00 Praha 1, Czechoslovakia
pirott@dknkurz1.bitnet

Manfred Pinkal

Department of Computational Linguistics, University of Saarbrücken
W-6600 Saarbrücken, Germany
pinkal@coli.uni-sb.de

Tanya Reinhart

Linguistics, Tel Aviv University
Tel Aviv, Israel
f32@taunos.bitnet

Uwe Reyle

Institute for Computational Linguistics, University of Stuttgart
Azenbergstraße 12, D-7000 Stuttgart 1, Germany
uwe@adler.philosophie.uni-stuttgart.de

Mark Ryan

Department of Computing, Imperial College
180 Queen's Gate, London SW7 2BZ, England
mdr@doc.ic.ac.uk

Kjell Johan Sæbø

Germanistisk Insitutt, University of Oslo

Postboks 1004, N-0315 Oslo, Norway

`kjelljs@hedda.uio.no`

Roger Schwarzschild

English Department, Hebrew University of Jerusalem

Mt. Scopus, Jerusalem 91905, Israel

`hcurs@hujivml.bitnet`

Hans Smessaert

Departement Linguistiek, Katholieke Universiteit Leuven

Blijde Inkomststraat 21, 3000 Leuven, Belgium

`ffaab01@blekul11.bitnet`

Edith Spaan

Faculteit der Wiskunde en Informatica, Universiteit van Amsterdam

Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands

`edith@fwi.uva.nl`

Edward P. Stabler

Department of Linguistics, UCLA

405 Hilgard Avenue, Los Angeles CA 90024, United States

`stabler@cognet.ucla.edu`

Patrick Suppes

Ventura Hall, Stanford University

Stanford CA 94305-4115, United States

`suppes@csli.stanford.edu`

Marten Trautwein

Department of Mathematics and Computer Science, University of Amsterdam

Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands

`mtrautwe@fwi.uva.nl`

Fer-Jan de Vries

CWI Amsterdam, The Netherlands

Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

`ferjan@cw.nl`

Alessandro Zucchi

Department of Linguistics, Stanford University

Stanford, CA 94305-2150, United States

`zucchi@csli.stanford.edu`



