THE ROLE OF LISTS IN A CATEGORIAL ANALYSIS OF COORDINATION

MICHAEL SCHIEHLEN

Institute for Natural Language Processing
University of Stuttgart
Michael.Schiehlen@ims.uni-stuttgart.de

The paper proposes categorial analyses for coordination with multiple conjuncts, correlative coordination, and *respectively* coordination. It argues that in a categorial setting these phenomena can only be adequately analysed if a data structure of lists is introduced. To this purpose the Lambek Calculus is extended with the Kleene star, a connective that has already been explored in other substructural logics. Correspondingly, the λ calculus is extended with list-forming operators as motivated by the analysis of the coordination phenomena.

1. Introduction

Like other syntactic theories, categorial grammar is concerned with the composition of form and meaning, i.e. the definition of grammatically well-formed strings and their interpretation. In categorial grammar, the Curry–Howard correspondence makes sure that the two processes are so tightly coupled that they constrain each other in a non-trivial way. Formally, the object of investigation in grammar is the set $\mathcal{L}$ of grammatically well-formed strings of words or categories. Categorial grammar provides an infinite supply of categories inductively defined from a finite set of atomic categories and two type-forming connectives, the leftward slash (\) and the rightward slash (/). The compound categories can be defined in terms of the more basic categories (1); here $\cdot$ denotes string concatenation.

(1) a. $X\backslash Y = \{\, x \in \mathcal{L} : \forall y \in Y : x \cdot y \in X \,\}$

b. $X/Y = \{\, x \in \mathcal{L} : \forall y \in Y : y \cdot x \in X \,\}$

Standing for "incomplete" expressions, compound categories can be interpreted as functions: $\tau(X\backslash Y) = \tau(X/Y) = \tau(X)^{\tau(Y)}$. Another characteristic of categorial grammar is its view on syntactic derivation: Parsing is seen as a form of reasoning that can be couched in a deductive system like the Associative Lambek Calculus (ALC) (Lambek 1958). ALC provides for every type-forming connective a left rule (which eliminates the connective in one of the premises) and a right rule (which deals with the connective in the conclusion). Right rules are associated with operations that construct new data structures (e.g. functions in ALC); left rules correspond to operations that deconstruct.

(2) a. $\dfrac{}{X : t \Rightarrow X : t}$ Id

b. $\dfrac{T \Rightarrow Y : t' \quad \Gamma[X : t(t')] \Rightarrow Z : t''}{\Gamma[X/Y : t,\ T] \Rightarrow Z : t''}$ L/ $\qquad \dfrac{T,\ Y : x \Rightarrow X : t}{T \Rightarrow X/Y : \lambda x.t}$ R/

c. $\dfrac{T \Rightarrow Y : t' \quad \Gamma[X : t(t')] \Rightarrow Z : t''}{\Gamma[T,\ X\backslash Y : t] \Rightarrow Z : t''}$ L\ $\qquad \dfrac{Y : x,\ T \Rightarrow X : t}{T \Rightarrow X\backslash Y : \lambda x.t}$ R\

This paper argues for the need of an additional type-forming connective in categorial grammar, which stands for lists. Lists of this kind have already been used in other substructural logics (Restall 2000) for the purpose of the verification of loops in programs.In the computational literature, the connective has been expressed by the Kleene star, a practice that we will follow. The paper argues that certain cases of coordination cannot be analysed in a manner compliant to the Curry–Howard correspondence unless a data structure for lists is assumed during syntactic composition.

2. Multi–Conjunct Coordination

The first puzzle is the ability of coordinating conjunction to take an arbitrary number of arguments. As each subcategorization frame is expressed with a different lexical entry in categorial grammar, this means that infinitely many entries are associated with a coordinating conjunction.

(3) **and**: $(((X\backslash X)\backslash X)\backslash X)/X : \lambda x_4 \lambda x_3 \lambda x_2 \lambda x_1 . x_1 \sqcap x_2 \sqcap x_3 \sqcap x_4$

To ensure a finite lexicon, a connective is required that can represent category lists of arbitrary length (also cf. (Morrill 1994, 212)): the Kleene star. In (4a), individual list items are separated by commas (just as conjuncts are in written language).

(4) $X^* = \bigcup_{n \geq 1} X^n$ where $X^1 = X$ and $X^{n+1} = \{x_1 \cdot, \cdot x_2 : x_1 \in X, x_2 \in X^n\}$

Starred categories are interpreted as n-tuples of category denotations (5).

(5) $\tau(X^*) = \bigcup_{n \geq 1} \tau(X)^n$

For interpretation, we need to equip the semantic representation language (i.e. the lambda calculus) with operators for constructing and deconstructing list objects. Lists are constructed inductively with two operators (single-item lists with $\langle\!\langle . \rangle\!\rangle$, multiple-item lists with list concatenation $+$). A single item-list denotes its sole member ($[\![\langle\!\langle x \rangle\!\rangle]\!] = [\![x]\!]$); list concatenation is defined in (6).

(6) $[\![L_1 + L_2]\!] = \{\ \langle x_1 \ldots x_n, y_1 \ldots y_m \rangle : \quad \langle x_1 \ldots x_n \rangle \in [\![L_1]\!] \quad \wedge$
$\qquad \langle y_1 \ldots y_m \rangle \in [\![L_2]\!] \quad \}$

List objects can be reconverted into basic objects with the **join** operation. The **join**-operator uses the binary operation o to reconnect the list items. An inductive definition is given in 7. The two clauses also function as β reduction rules.

(7) $\textbf{join}(o, \langle\!\langle x \rangle\!\rangle) \Rightarrow x$
$\textbf{join}(o, L_1 + L_2) \Rightarrow o(\textbf{join}(o, L_1), \textbf{join}(o, L_2))$

With all these operators available, we can state the lexical entry of coordinating conjunctions (8). In contrast to entry (3), the entry (8) only introduces exactly one occurrence of the conjunction connective $\sqcap$. Multiplication of this connective is carried out by β–reduction (7). It is often useful to include the last conjunct in the conjunct list as well. This can be achieved by wrapping the conjunction around the last conjunct, e.g. with the help of an additional polymorphic variable Y (8).

(8) **and**: $(X\backslash(X^*/Y))/Y : \lambda t \lambda P.\textbf{join}(\sqcap, P(t))$

In compliance with the Curry–Howard correspondence, each operator is associated with a proof rule. The two right rules **R*** and **M*** are adopted from Restall (2000, 55f).

(9) a. $$\frac{T \Rightarrow X : x}{T \Rightarrow X^* : \langle\!\langle x \rangle\!\rangle}\ \text{R*}$$

b. $$\frac{P \Rightarrow X^* : L_1 \quad Q \Rightarrow X^* : L_2}{P,\ Q \Rightarrow X^* : L_1 + L_2}\ \text{M*}$$

c. $$\frac{X : a,\ X : b \Rightarrow X : c \quad \Gamma[X : \textbf{join}(\lambda b \lambda a.c, L)] \Rightarrow Z : z}{\Gamma[X^* : L] \Rightarrow Z : z}\ \text{L*}_1$$

3. Partial Distribution in Multi–Conjunct Coordination

By now, we have two operators for constructing lists and one operator for deconstructing lists. A fourth operator can be used to apply functions to lists directly. The **map**-operator, inductively defined in (10), modifies a list L by applying a function f to each list item. Again the two clauses also function as β reduction rules.

(10) $\textbf{map}(f, \langle\!\langle x \rangle\!\rangle) \Rightarrow \langle\!\langle f(x) \rangle\!\rangle$
$\textbf{map}(f, L_1 + L_2) \Rightarrow \textbf{map}(f, L_1) + \textbf{map}(f, L_2)$

The proof rule for the **map** operator, again a left rule, is given in (11).

(11) $$\frac{\Gamma[X : x] \Rightarrow Y : y}{\Gamma[X^* : L] \Rightarrow Y^* : \textbf{map}(\lambda x.y, L)}\ \text{L*}_2$$

In an analysis of coordination, the **map** operator can be used to distribute over conjuncts. In cases that require such distribution, a functor f which syntactically applies to a coordination semantically applies to each of the conjuncts. The idea of the analysis is as follows: The list types allow to separate the task of collecting the conjuncts (rules **R*** and **M***) from the application of the conjunction functor (rule $\mathbf{L*}_1$). Hence a scopal item can get scope over all conjuncts without leaving the scope of conjunction: by applying to the list (rule $\mathbf{L*}_2$).

A syntactic account of distribution has certain advantages with respect to effects that concern the interplay of syntax and semantics. Larson (1985, 220) has observed that the placement of the correlate of a coordinating conjunction fixes the conjunction's semantic scope. So, the position of the correlate in example (12) enforces wide scope for the disjunction. The effect follows if we assume that the correlate is picked up as the last argument of the conjunction (so that e.g. *or* would get the entry $((X\backslash \mathit{either})\backslash(X^*/Y))/Y$).

(12) Mary is either looking for a maid or a cook.

Hudson (1989, 89) discusses cases where a functor is only partially distributed, i.e. distributed not over the entire conjunct list, but only over a contiguous sublist (13).

(13) a. in the United States, (the) Netherlands and in England

b. either in England, in the United States or (the) Netherlands

Cases of partial distribution provide strong motivation for the assumption that list formation and conjunct interpretation should be separated. Only such a factorization allows the propagation of the conjunction connective to sublists that do not include the coordinating conjunction. To analyse partial distribution, a list must be decomposable into arbitrary sublists. Hence, a general operation of list composition (as in rule **M***) is required; it would not suffice to only consider lists of a string-like structure, i.e. lists where a always a single element is prepended or appended.

4. Respectively–Coordination

A third argument for a process of list formation is provided by *respectively*–coordination (ResC). In ResC several surface conjunctions are conflated to a single functor on the semantic form (14).

(14) John and Peter love Mary and hate Sue, respectively.
love(j,m) $\wedge$ hate(p,s)

An occurrence of ResC consists of at least two coordinations. All but one coordination are modified by the adverb *respectively*. We will call the unique coordination without *respectively* the governing member, and all other coordinations the dependent members. Occurrences of ResC with more than two members (cf. (15) from (Schachter 1973, 390)) can be regarded as recursive applications of binary ResC.

(15) [John and Bill went to New York and Chicago respectively] on Monday and Wednesday respectively.

Each member coordination must have exactly the same number of conjuncts (or, as we shall say, the same arity). In the interpretation, the conjuncts of the members are correlated so that every i-th conjunct relates to the other i-th conjuncts. The correlating behaviour is reminiscent of the scalar product (16).

(16) $\langle x_1, \ldots, x_n \rangle \cdot \langle y_1, \ldots, y_n \rangle = x_1 y_1 + \ldots + x_n y_n$

The scalar product will be taken as a guideline in developping a theory of ResC. The fact that ResC can be stacked (cf. example (15)) makes necessary an adjustment, however. The linguistic operation, which will be called **vecp** for vector product, transforms its two argument lists not into a basic object but into yet another list. The operation is inductively be defined in (17). Again the definition clauses of (17) also serve as β–reduction rules.

(17) $\textbf{vecp}(f, \langle\!\langle x_1 \rangle\!\rangle, , \langle\!\langle x_2 \rangle\!\rangle) \Rightarrow \langle\!\langle f(x_1)(x_2) \rangle\!\rangle$
$\textbf{vecp}(f, \langle\!\langle x_1 \rangle\!\rangle + L_1, \langle\!\langle x_2 \rangle\!\rangle + L_2) \Rightarrow \langle\!\langle f(x_1)(x_2) \rangle\!\rangle + \textbf{vecp}(f, L_1, L_2)$
$\textbf{vecp}(f, L_1, (L_2 + L_3) + L_4) \Rightarrow \textbf{vecp}(f, L_1, L_2 + (L_3 + L_4))$
$\textbf{vecp}(f, (L_1 + L_2) + L_3, L_4) \Rightarrow \textbf{vecp}(f, L_1 + (L_2 + L_3), L_4)$

vecp is the only list operator for which order is relevant. But for **vecp** and ResC, multisets could have been used instead of lists. By the Curry–Howard correspondence, **vecp** can be correlated with the following left rule (18).

(18) $$\frac{X : x,\ Y : y \Rightarrow U : u \quad \Gamma[U^* : \textbf{vecp}(\lambda x \lambda y.u, L_1, L_2)] \Rightarrow Z}{\Gamma[X^* : L_1,\ Y^* : L_2] \Rightarrow Z}\ \text{L*}_3$$

The analysis has to come to grips with the fact that in the final representation the coordinating conjunctions of all member coordinations are conflated to just one instance. We assume that this instance is triggered by the governing member. Conjunctions in dependent coordinations merely pass on their conjunct lists (19).

(19) **and** $\quad ((X^* \mid \textit{respectively}) \backslash (X^*/Y))/Y : \lambda t \lambda P \lambda r.P(t)$

The conjunction in the governing coordination has a local effect (essentially null) and a global effect (introducing its meaning). In this respect it is similar to quantifiers, and the techniques used for quantifier raising may be applied (e.g. wrapping, polymorphism, or a lexical entry communicating with a unary rule).

(20) **and** locally: $(X^* \backslash (X^*/Y))/Y : \lambda t \lambda P.P(t)$
globally: $X \mid X^* : \lambda L.\textbf{join}(\sqcap, L)$

We assume that some modality or feature mechanism controls the communication between the rules and lexical entries in (18, 19, 20) and ensures e.g. the presence of exactly one governing coordination and at least two members.

The consequences of ResC for the complexity of syntax have been hotly debated in the literature. ResC has been used as an argument that natural language is not even mildly context-sensitive (Kac 1987). To salvage context-freeness, it is important that the same–arity restriction be tested not before in semantics. That semantics plays an important role is obvious in constructions with plural noun phrases (21): The number of partition cells, though relevant for ResC, has no reflex in syntax (Pullum and Gazdar 1982, 500.fn(10)).

(21) during the period of squabbling between court factions supporting Russia or Japan respectively (BNC)

In the analysis presented here, the arity restriction is not checked during composition but rather during β–reduction (17). For invalid sentences, the parser produces a result, but the result includes unreduced lists.

5. Conclusion

The paper has argued for the necessity of including lists in the categorial analysis of coordination. First, without lists, either the phrase structure rule base or the categorial lexicon will cease to be finite, as a single coordinating conjunction can connect arbitrarily many conjuncts. Second, lists allow for an explicit modelling of distribution over conjuncts. Such a treatment is needed e.g. to adequately express the mutual restrictions between syntax and options for distribution. It is also needed to account for partial distribution, i.e. distribution over only a subset of conjuncts in a multi–conjunct coordination. Finally, lists are a handy tool in analysing *respectively* coordination. I would like to thank Hans Kamp, Kristina Spranger, and an anonymous reviewer for their helpful comments.

Bibliography

Hudson, R. A.: 1989, Gapping and grammatical relations, *Journal of Linguistics* 25

Kac, M. B.: 1987, Surface Transitivity, *Respectively* Coordination and Context-Freeness, *Natural Language and Linguistic Theory* 5, 441–452

Lambek, J.: 1958, The mathematics of sentence structure, *American Mathematical Monthly* 65, 154–170,

Larson, R. K.: 1985, On the Syntax of Disjunction Scope, *Natural Language and Linguistic Theory* 3, 217–265

Morrill, G. V.: 1994, *Type Logical Grammar*, Kluwer

Pullum, G. and Gazdar, G.: 1982, Natural Languages and Context-Free Languages, *Linguistics and Philosophy* 4, 471–504

Restall, G.: 2000, *An Introduction to Substructural Logics*, Routledge

Schachter, P.: 1973, Conjunction, in R. P. Stockwell, P. Schachter, and B. H. Partee (eds.), *The Major Syntactic Structures of English*, pp 294–418, Holt, Rinehart & Winston