Lexical and Derivational Meaning in Vector-Based Models of Relativisation

Michael Moortgat[1] and Gijs Wijnholds[2]

[1] Utrecht University, The Netherlands
m.j.moortgat@uu.nl
[2] Queen Mary University of London, United Kingdom
g.j.wijnholds@qmul.ac.uk

Abstract

Sadrzadeh et al (2013) present a compositional distributional analysis of relative clauses in English in terms of the Frobenius algebraic structure of finite dimensional vector spaces. The analysis relies on distinct type assignments and lexical recipes for subject vs object relativisation. The situation for Dutch is different: because of the verb final nature of Dutch, relative clauses are ambiguous between a subject vs object relativisation reading. Using an extended version of Lambek calculus, we present a compositional distributional framework that accounts for this derivational ambiguity, and that allows us to give a single meaning recipe for the relative pronoun reconciling the Frobenius semantics with the demands of Dutch derivational syntax.

1 Introduction

Compositionality, as a structure-preserving mapping from a syntactic source to a target interpretation, is a fundamental design principle both for the set-theoretic models of formal semantics and for syntax-sensitive vector-based accounts of natural language meaning, see [1] for discussion. For typelogical grammar formalisms, to obtain a compositional interpretation, we have to specify how the Syn-Sem homomorphism acts on *types* (basic and complex) and on *proofs* (derivations, again basic (axioms) or compound, obtained by inference steps). There is a tension here between lexical and derivational aspects of meaning: the derivational aspects relate to the composition operations associated with the inference steps that put together phrases out of more elementary parts; the atoms for this composition process are the meanings of the lexical constants associated with the axioms of a derivation.

Relative clause structures form a suitable testbed to study the interaction between these two aspects of meaning, and they have been well-studied in the formal and in the distributional settings. Informally, a restrictive relative clause ('books that Alice read') has an intersective interpretion. In the formal semantics account, this interpretation is obtained by modeling both the head noun ('books') and the relative clause body ('Alice read ␣') as (characteristic functions of) sets (type $e \to t$); the relative pronoun can then be interpreted as the intersection operation. In distributional accounts such as [2], full noun phrases and simple common nouns are interpreted in the same semantic space, say N, distinct from the sentence space S. In this setting, element-wise multiplication, which preserves non-null context features, is a natural candidate for an intersective interpretation; in the case at hand this means element-wise multiplication of a vector in N interpreting the head noun, with a vector interpretation obtained from the relative clause body. To achieve this effect, [9] rely on the Frobenius algebraic structure of **FVect**, which provides operations for (un)copying, insertion and deletion of vector information. A key feature of their account is that it relies on *structure-specific* solutions of the lexical equation: subject and object relative clauses are obtained from distinct type assignments to the relative

$$\frac{}{1_A : A \longrightarrow A} \qquad \frac{f : A \longrightarrow B \quad g : B \longrightarrow C}{g \circ f : A \longrightarrow C}$$

$$\frac{f : \Diamond A \longrightarrow B}{\triangledown f : A \longrightarrow \Box B} \qquad \frac{f : A \otimes B \longrightarrow C}{\triangleright f : A \longrightarrow C/B} \qquad \frac{f : A \otimes B \longrightarrow C}{\triangleleft f : B \longrightarrow A\backslash C}$$

$$\frac{g : A \longrightarrow \Box B}{\triangledown^{-1} g : \Diamond A \longrightarrow B} \qquad \frac{g : A \longrightarrow C/B}{\triangleright^{-1} g : A \otimes B \longrightarrow C} \qquad \frac{g : B \longrightarrow A\backslash C}{\triangleleft^{-1} g : A \otimes B \longrightarrow C}$$

$$\alpha^l_\diamond : \Diamond A \otimes (B \otimes C) \longrightarrow (\Diamond A \otimes B) \otimes C \qquad \alpha^r_\diamond : (A \otimes B) \otimes \Diamond C \longrightarrow A \otimes (B \otimes \Diamond C)$$
$$\sigma^l_\diamond : \Diamond A \otimes (B \otimes C) \longrightarrow B \otimes (\Diamond A \otimes C) \qquad \sigma^r_\diamond : (A \otimes B) \otimes \Diamond C \longrightarrow (A \otimes \Diamond C) \otimes B$$

Figure 1: $\mathbf{NL}_\diamond$. Residuation rules; extraction postulates.

pronoun (Lambek types $(n\backslash n)/(np\backslash s)$ vs $(n\backslash n)/(s/np)$), associated with distinct instructions for meaning assembly.

For a language like Dutch, such an account is problematic. Dutch subordinate clause order has the SOV pattern Subj–Obj–TV, i.e. a transitive verb is typed as $np\backslash(np\backslash s)$, selecting its arguments uniformly to the left. As a result, example (1)(a) is ambiguous between a subject vs object relativisation interpretation: it can be translated as either (b) or (c). The challenge here is twofold: at the syntactic level, we have to provide a *single* type assignment to the relative pronoun that can withdraw either a subject or an object hypothesis from the relative clause body; at the semantic level, we need a *uniform* meaning recipe for the relative pronoun that will properly interact with the derivational semantics.

a	mannen$_n$ die$_?$ vrouwen$_{np}$ haten$_{np\backslash(np\backslash s)}$	(ambiguous)	
b	men who hate women	(subject rel)	(1)
c	men who(m) women hate	(object rel)	

The paper is structured as follows. In §2, we present an extended version of Lambek calculus, and show how it accounts for the derivational ambiguity of Dutch relative clauses. In §3.1, we define the interpretation homomorphism that associates syntactic derivations with composition operations in a vector-based semantic model. The derivational semantics thus obtained is formulated at the type level, i.e. it abstracts from the contribution of individual lexical items. In §3.2, we bring in the lexical semantics, and show how the Dutch relative pronoun can be given a uniform interpretation that properly interacts with the derivational semantics. The discussion in §4 compares the distributional and formal semantics accounts of relativisation.

2 Syntax

Our syntactic engine is $\mathbf{NL}_\diamond$ [6]: the extension of Lambek's [3] Syntactic Calculus with an adjoint pair of control modalities $\Diamond, \Box$. The modalities play a role similar to that of the exponentials of linear logic: they allow one to introduce controlled, rather than global, forms of reordering and restructuring. In this paper, we consider the controlled associativity and commutativity postulates of [7]. One pair, $\alpha^l_\diamond, \sigma^l_\diamond$, allows a $\Diamond$-marked formula to reposition itself on left branches of a constituent tree; we use it to model the SOV extraction patterns in Dutch. A symmetric pair $\alpha^r_\diamond, \sigma^r_\diamond$ would capture the non-local extraction dependencies in an SVO language such as English. Lambek [4] has shown how deductions in a syntactic calculus can be viewed as arrows in a category. Figure 1 presents $\mathbf{NL}_\diamond$ in this format.

For parsing, we want a proof search procedure that doesn't rely on cut. Consider the rules in Figure 2, expressing the monotonicity properties of the type-forming operations, and recasting the postulates in rule form. It is routine to show that these are *derived* rules of inference of $\mathbf{NL}_\diamond$. In [8] it is shown that by adding them to the residuation rules of Figure 1, one obtains a system equivalent to a display sequent calculus enjoying cut-elimination. By further restricting to *focused* derivations, proof search is free of spurious ambiguity.

We are ready to return to our example (1)(a). A type assignment $(n\backslash n)/(\Diamond\Box np\backslash s)$ to the relative pronoun 'die' accounts for the derivational ambiguity of the phrase. The derivations agree on the initial steps

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{n \longrightarrow n} \quad \overline{n \longrightarrow n}}{n\backslash n \longrightarrow n\backslash n}\ \backslash \qquad \dfrac{\vdots}{np \otimes (np\backslash(np\backslash s)) \longrightarrow \Diamond\Box np\backslash s}}{(n\backslash n)/(\Diamond\Box np\backslash s) \longrightarrow (n\backslash n)/(np \otimes (np\backslash(np\backslash s)))}\ /}{((n\backslash n)/(\Diamond\Box np\backslash s)) \otimes (np \otimes (np\backslash(np\backslash s))) \longrightarrow n\backslash n}\ \triangleright^{-1}}{n \otimes (((n\backslash n)/(\Diamond\Box np\backslash s)) \otimes (np \otimes (np\backslash(np\backslash s)))) \longrightarrow n}\ \triangleleft^{-1} \tag{2}$$

but then diverge in how the relative clause body is derived:

$$\dfrac{\dfrac{\overline{np \longrightarrow np} \qquad \boxed{\dfrac{\dfrac{\dfrac{\overline{np \longrightarrow np}}{\Box np \longrightarrow \Box np}\ \Box}{\Diamond\Box np \longrightarrow np}\ \triangledown^{-1} \quad \overline{s \longrightarrow s}}{np\backslash s \longrightarrow \Diamond\Box np\backslash s}\ \backslash}}{np\backslash(np\backslash s) \longrightarrow np\backslash(\Diamond\Box np\backslash s)}\ \backslash}{np \otimes (np\backslash(np\backslash s)) \longrightarrow \Diamond\Box np\backslash s}\ \triangleleft^{-1} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\boxed{\dfrac{\dfrac{\dfrac{\overline{np \longrightarrow np}}{\Box np \longrightarrow \Box np}\ \Box}{\Diamond\Box np \longrightarrow np}\ \triangledown^{-1} \quad \dfrac{\overline{np \longrightarrow np} \quad \overline{s \longrightarrow s}}{np\backslash s \longrightarrow np\backslash s}\ \backslash}{np\backslash(np\backslash s) \longrightarrow \Diamond\Box np\backslash(np\backslash s)}\ \backslash}}{\Diamond\Box np \otimes (np\backslash(np\backslash s)) \longrightarrow np\backslash s}\ \triangleleft^{-1}}{np \otimes (\Diamond\Box np \otimes (np\backslash(np\backslash s))) \longrightarrow s}\ \triangleleft^{-1}}{\Diamond\Box np \otimes (np \otimes (np\backslash(np\backslash s))) \longrightarrow s}\ \widehat{\sigma}^l_\diamond}{np \otimes (np\backslash(np\backslash s)) \longrightarrow \Diamond\Box np\backslash s}\ \triangleleft \tag{3}$$

In the derivation on the left, the $\Diamond\Box np$ hypothesis is linked to the *subject* argument of the verb; in the derivation on the right to the *object* argument, reached via the $\widehat{\sigma}^l_\diamond$ reordering step.

$$\dfrac{f : A \longrightarrow B}{\Diamond f : \Diamond A \longrightarrow \Diamond B} \qquad \dfrac{f : A \longrightarrow B}{\Box f : \Box A \longrightarrow \Box B}$$

$$\dfrac{f : A \longrightarrow B \quad g : C \longrightarrow D}{f \otimes g : A \otimes C \longrightarrow B \otimes D} \qquad \dfrac{f : A \longrightarrow B \quad g : C \longrightarrow D}{f/g : A/D \longrightarrow B/C} \qquad \dfrac{f : A \longrightarrow B \quad g : C \longrightarrow D}{f\backslash g : B\backslash C \longrightarrow A\backslash D}$$

$$\dfrac{f : (\Diamond A \otimes B) \otimes C \longrightarrow D}{\widehat{\alpha}^l_\diamond f : \Diamond A \otimes (B \otimes C) \longrightarrow D} \qquad \dfrac{f : B \otimes (\Diamond A \otimes C) \longrightarrow D}{\widehat{\sigma}^l_\diamond f : \Diamond A \otimes (B \otimes C) \longrightarrow D}$$

Figure 2: $\mathbf{NL}_\diamond$. Monotonicity; leftward extraction (rule version).

3 From source to target

3.1 Derivational semantics

Compositional distributional models are obtained by defining a homomorphism sending types and derivations of a syntactic source system to their counterparts in a symmetric compact closed category (sCCC); the concrete model for this sCCC then being finite dimensional vector spaces (**FVect**) and (multi)linear maps. Such interpretation homomorphisms have been defined for pregroup grammars, Lambek calculus and CCG in [2, 5]. We here define the interpretation for $\mathbf{NL}_\diamond$, starting out from [10].

Recall first that a *compact closed category* (CCC) is monoidal, i.e. it has an associative $\otimes$ with unit I; and for every object there is a left and a right adjoint satisfying

$$A^l \otimes A \xrightarrow{\epsilon^l} I \xrightarrow{\eta^l} A \otimes A^l \qquad A \otimes A^r \xrightarrow{\epsilon^r} I \xrightarrow{\eta^r} A^r \otimes A$$

In a *symmetric* CCC, the tensor moreover is commutative, and we can write A^* for the collapsed left and right adjoints.

In the concrete instance of **FVect**, the unit I stands for the field $\mathbb{R}$; identity maps, composition and tensor product are defined as usual. Since bases of vector spaces are fixed in concrete models, there is only one natural way of defining a basis for a *dual space*, so that $V^* \cong V$. In concrete models we may collapse the adjoints completely.

The ϵ map takes inner products, whereas the η map (with $\lambda = 1$) introduces an identity tensor as follows:

$$\begin{array}{lllll} \epsilon_V : V \otimes V \to \mathbb{R} & \text{given by} & \sum_{ij} v_{ij}(\vec{e}_i \otimes \vec{e}_j) & \mapsto & \sum_i v_{ii} \\ \eta_V : \mathbb{R} \to V \otimes V & \text{given by} & \lambda & \mapsto & \sum_i \lambda(\vec{e}_i \otimes \vec{e}_i) \end{array}$$

Interpretation: types At the type level, the interpretation function $\lceil\cdot\rceil$ assigns a vector space to the atomic types of $\mathbf{NL}_\diamond$; for complex types we set $\lceil\Diamond A\rceil = \lceil\Box A\rceil = \lceil A\rceil$, i.e. the syntactic control operators are transparent for the interpretation; the binary type-forming operators are interpreted as

$$\lceil A \otimes B\rceil = \lceil A\rceil \otimes \lceil B\rceil \quad \lceil A/B\rceil = \lceil A\rceil \otimes \lceil B\rceil^* \quad \lceil A\backslash B\rceil = \lceil A\rceil^* \otimes \lceil B\rceil$$

Interpretation: proofs From the linear maps interpreting the premises of the $\mathbf{NL}_\diamond$ inference rules, we want to compute the linear map interpreting the conclusion. Identity and composition are immediate: $\lceil 1_A\rceil = 1_{\lceil A\rceil}$, $\lceil g \circ f\rceil = \lceil g\rceil \circ \lceil f\rceil$. For the residuation inferences, from the map $\lceil f\rceil : \lceil A\rceil \otimes \lceil B\rceil \longrightarrow \lceil C\rceil$ interpreting the premise, we obtain

$$\begin{array}{l} \lceil \triangleright f\rceil = \lceil A\rceil \xrightarrow{1_{\lceil A\rceil} \otimes \eta_{\lceil B\rceil}} \lceil A\rceil \otimes \lceil B\rceil \otimes \lceil B\rceil^* \xrightarrow{\lceil f\rceil \otimes 1_{\lceil B\rceil^*}} \lceil C\rceil \otimes \lceil B\rceil^* \\ \lceil \triangleleft f\rceil = \lceil B\rceil \xrightarrow{\eta_{\lceil A\rceil} \otimes 1_{\lceil B\rceil}} \lceil A\rceil^* \otimes \lceil A\rceil \otimes \lceil B\rceil \xrightarrow{1_{\lceil A\rceil^*} \otimes \lceil f\rceil} \lceil A\rceil^* \otimes \lceil C\rceil \end{array}$$

For the inverses, from maps $\lceil g\rceil : \lceil A\rceil \longrightarrow \lceil C/B\rceil$, $\lceil h\rceil : \lceil B\rceil \longrightarrow \lceil A\backslash C\rceil$ for the premises, we obtain

$$\begin{array}{l} \lceil \triangleright^{-1} g\rceil = \lceil A\rceil \otimes \lceil B\rceil \xrightarrow{\lceil g\rceil \otimes 1_{\lceil B\rceil}} \lceil C\rceil \otimes \lceil B\rceil^* \otimes \lceil B\rceil \xrightarrow{1_{\lceil C\rceil} \otimes \epsilon_{\lceil B\rceil}} \lceil C\rceil \\ \lceil \triangleleft^{-1} h\rceil = \lceil A\rceil \otimes \lceil B\rceil \xrightarrow{1_{\lceil A\rceil} \otimes \lceil h\rceil} \lceil A\rceil \otimes \lceil A\rceil^* \otimes \lceil C\rceil \xrightarrow{\epsilon_{\lceil A\rceil} \otimes 1_{\lceil C\rceil}} \lceil C\rceil \end{array}$$

Monotonicity. The case of parallel composition is immediate: $\lceil f \otimes g \rceil = \lceil f \rceil \otimes \lceil g \rceil$. For the slash cases, from $\lceil f \rceil : \lceil A \rceil \longrightarrow \lceil B \rceil$ and $\lceil g \rceil : \lceil C \rceil \longrightarrow \lceil D \rceil$, we obtain

$$\lceil f/g \rceil = \qquad\qquad \lceil f \backslash g \rceil =$$

$$\begin{array}{rc}
 & \lceil A \rceil \otimes \lceil D \rceil^* \\
\lceil f \rceil \otimes \eta_{\lceil C \rceil} \otimes 1_{\lceil D \rceil^*} & \downarrow \\
 & \lceil B \rceil \otimes \lceil C \rceil^* \otimes \lceil C \rceil \otimes \lceil D \rceil^* \\
1_{\lceil B \rceil \otimes \lceil C \rceil^*} \otimes \lceil g \rceil \otimes 1_{\lceil D \rceil^*} & \downarrow \\
 & \lceil B \rceil \otimes \lceil C \rceil^* \otimes \lceil D \rceil \otimes \lceil D \rceil^* \\
1_{\lceil B \rceil \otimes \lceil C \rceil^*} \otimes \epsilon_{\lceil D \rceil} & \downarrow \\
 & \lceil B \rceil \otimes \lceil C \rceil^*
\end{array}
\qquad
\begin{array}{cl}
\lceil B \rceil^* \otimes \lceil C \rceil & \\
\downarrow & 1_{\lceil B \rceil^*} \otimes \eta_{\lceil A \rceil} \otimes \lceil g \rceil \\
\lceil B \rceil^* \otimes \lceil A \rceil \otimes \lceil A \rceil^* \otimes \lceil D \rceil & \\
\downarrow & 1_{\lceil B \rceil^*} \otimes \lceil f \rceil \otimes 1_{\lceil A \rceil^* \otimes \lceil D \rceil} \\
\lceil B \rceil^* \otimes \lceil B \rceil \otimes \lceil A \rceil^* \otimes \lceil D \rceil & \\
\downarrow & \epsilon_{\lceil B \rceil} \otimes 1_{\lceil A \rceil^* \otimes \lceil D \rceil} \\
\lceil A \rceil^* \otimes \lceil D \rceil &
\end{array}$$

Interpretation for the extraction structural rules is obtained via the standard associativity and symmetry maps of **FVect**: $\lceil \widehat{\alpha}^l_\diamond f \rceil = f \circ \alpha$ and $\lceil \widehat{\sigma}^l_\diamond f \rceil = f \circ \alpha^{-1} \circ (\sigma \otimes 1_A) \circ \alpha$ and similarly for the rightward extraction rules.

Simplifying the interpretation Whereas the syntactic derivations of $\mathbf{NL}_\diamond$ proceed in cut-free fashion, the interpretation of the inference rules given above introduces detours (sequential composition of maps) that can be removed. We use a generalised notion of Kronecker delta, together with Einstein summation notation, to concisely express the fact that the interpretation of a derivation is fully determined by the identity maps that interpret its axiom leaves, realised as the ϵ or η identity matrices depending on their (co)domain signature.

Recall that vectors and linear maps over the real numbers can be equivalently expressed as (multi-dimensional) arrays of numbers. The essential information one needs to keep track of are the coefficients of the tensor: for a vector $\mathbf{v} \in \mathbb{R}^n$ we write v_i (with i ranging from 1 to n), an $n \times m$ matrix $\mathbf{A}$ is expressed as A_{ij}, an $n \times m \times p$ cube $\mathbf{B}$ as B_{ijk}, with the indices each time ranging over the dimensions. The Einstein summation convention on indices then states that in an expression involving multiple tensors, indices occurring once give rise to a tensor product, whereas indices occurring twice are contracted. Without explicitly writing a tensor product $\otimes$, the tensor product of a vector $\mathbf{a}$ and a matrix $\mathbf{A}$ thus can be written as $a_i A_{jk}$; the inner product between vectors $\mathbf{a}, \mathbf{b}$ is $a_i b_i$. Matrix application $\mathbf{Aa}$ is rendered as $A_{ij} a_j$, i.e. the contraction happens over the second dimension of $\mathbf{A}$ and $\mathbf{a}$. For tensors of arbitrary rank we use uppercase to refer to lists of indices: we write a tensor $\mathbf{T}$ as T_I. Tensor application then becomes $T_{IJ} R_J$, for some tensor $\mathbf{R}$ of lower rank.

The identity matrix is given by the Kronecker delta (left), the identity tensor by its generalisation (right):

$$\delta^i_j = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases} \qquad \delta^I_J = \begin{cases} 1 & I_k = J_k \text{ for all } k \\ 0 & \text{otherwise} \end{cases}$$

The attractive property of the (generalised) Kronecker delta is that it expresses unification of indices: $\delta^i_j a_i = a_j$, which is simply a renaming of the index; the inner product can be computed by $\delta^i_j a_i b_j = a_j b_j$. Left on its own, it is simply an identity matrix/tensor.

With the Kronecker delta, the composition of matrices $\mathbf{B} \circ \mathbf{A}$ is expressible as $\delta_k^j A_{ij} B_{kl}$, which is the same as $A_{ij} B_{jl}$ (or $A_{ik} B_{kl}$). We can show that order of composition is irrelevant:

$$\delta_k^j A_{ij} \delta_m^l B_{kl} C_{mn} = A_{ij} B_{jl} C_{ln} = \delta_m^l \delta_k^j A_{ij} B_{kl} C_{mn}$$

The special cases of tensor product of generalised Kronecker deltas is given by concatenating the index lists:

$$\delta_J^I \otimes \delta_L^K = \delta_{JL}^{IK}$$

expressing the fact that $1_A \otimes 1_B = 1_{A \otimes B}$.

Since the generalised Kronecker delta is able to do renaming, take inner product, and insert an identity tensor, depending on the number of arguments placed behind it, it will represent precisely the $1_A, \epsilon_A, \eta_A$ maps discussed above. In this respect, the interpretation can be simplified and we can label the proof system (with formulas already interpreted) with these generalised Kronecker deltas. The effect of the residuation rules and the structural rules is to only change the (co)domain signature of a Kronecker delta, whereas the rules for axioms and monotonicity also act on the Kronecker delta itself:

$$\frac{}{A_I \xrightarrow{\delta_J^I} A_J}\ 1_A$$

$$\frac{A \xrightarrow{\delta_J^I} B \quad C \xrightarrow{\delta_L^K} D}{A \otimes C \xrightarrow{\delta_{JL}^{IK}} B \otimes D}\ \otimes \qquad \frac{A \xrightarrow{\delta_J^I} B \quad C \xrightarrow{\delta_L^K} D}{A \otimes D \xrightarrow{\delta_{JL}^{IK}} B \otimes C}\ / \qquad \frac{A \xrightarrow{\delta_J^I} B \quad C \xrightarrow{\delta_L^K} D}{B \otimes C \xrightarrow{\delta_{JL}^{IK}} A \otimes D}\ \backslash$$

In the full version of this paper (arXiv:1711:11513) we show that this labelling is correct for the general interpretation of proofs in §3.1.

3.2 Lexical semantics

For the general interpretation of types and proofs given above, a proof $f : A \longrightarrow B$ is interpreted as a linear map $\lceil f \rceil$ sending an element belonging to $\lceil A \rceil$, the semantic space interpreting A, to an element of $\lceil B \rceil$. The map is expressed at the general level of types, and completely abstracts from *lexical* semantics. For the computation of concrete interpretations, we have to bring in the meaning of the lexical items. For $A = A_1 \otimes \cdots \otimes A_n$, this means applying the map $\lceil f \rceil$ to $\mathbf{w}_1 \otimes \cdots \otimes \mathbf{w}_n$, the tensor product of the word meanings making up the phrase under consideration, to obtain a meaning $M \in \lceil B \rceil$, the semantic space interpreting the goal formula.

With the index notation introduced above, $\lceil f \rceil$ is expressed in the form of a generalised Kronecker delta, which is applied to the tensor product of the word meanings in index notation to produce the final meaning in $\lceil B \rceil$. In (4) we illustrate with the interpretation of some proofs derived from the same axiom leaves, $np \longrightarrow np$ and $s \longrightarrow s$. Assuming $\lceil np \rceil = \mathsf{N}$ and $\lceil s \rceil = \mathsf{S}$, these correspond to identity maps on N and S. We use the convention that the formula components of the endsequent are labelled in alphabetic order; the correct indexing for the Kronecker delta is obtained by working back to the axiom leaves.

$$\begin{array}{llll}
a & \mathsf{dream}^{np\backslash s} \longrightarrow np\backslash s & \mathbf{dream}_{i,j}^{\mathsf{N}\otimes\mathsf{S}} \xrightarrow{\delta_{i,l}^{k,j}} T_{k,l}^{\mathsf{N}\otimes\mathsf{S}} & \\
b & \mathsf{poets}^{np} \otimes \mathsf{dream}^{np\backslash s} \longrightarrow s & \mathbf{poets}_i^{\mathsf{N}} \otimes \mathbf{dream}_{j,k}^{\mathsf{N}\otimes\mathsf{S}} \xrightarrow{\delta_{j,l}^{i,k}} V_l^{\mathsf{S}} & (4) \\
c & \mathsf{poets}^{np} \longrightarrow s/(np\backslash s) & \mathbf{poets}_i^{\mathsf{N}} \xrightarrow{\delta_{k,j}^{i,l}} R_{j,k,l}^{\mathsf{S}\otimes\mathsf{N}\otimes\mathsf{S}} &
\end{array}$$

(4)(a) expresses the linear map from $\mathbf{dream} \in \mathsf{N} \otimes \mathsf{S}$ to a tensor $T \in \mathsf{N} \otimes \mathsf{S}$. Because we have $T = \delta_{i,l}^{k,j}\mathbf{dream}_{i,j} = \mathbf{dream}_{k,l}$, this is in fact the identity map. (4)(b) computes a vector $V \in \mathsf{S}$ with $V = \delta_{j,l}^{i,k}\mathbf{poets}_i \otimes \mathbf{dream}_{j,k} = \mathbf{poets}_j \otimes \mathbf{dream}_{j,l}$. In (4)(c) we arrive at an interpretation $R \in \mathsf{S} \otimes \mathsf{N} \otimes \mathsf{S}$ with $R = \delta_{k,j}^{i,l}\mathbf{poets}_i = \delta_j^l \mathbf{poets}_k$. Note that we wrote the tensor product symbol $\otimes$ explicitly.

In the case of our relative clause example (1), the derivational ambiguity of (3) gives rise to two ways of obtaining a vector $\mathbf{v} \in \mathsf{N}$. They differ in whether l, the index of the $\Diamond\Box np$ hypothesis in the relative pronoun type, contracts with index p for the subject argument of the verb (5) or with the direct object index o (6).

$$\begin{aligned} & \mathbf{mannen}_i \otimes \mathbf{die}_{jklm} \otimes \mathbf{vrouwen}_n \otimes \mathbf{haten}_{opq} \xrightarrow{\delta_{j,r,p,q,o}^{i,k,l,m,n}} \mathbf{v}_r^{subj} \in \mathsf{N} \\ \mathbf{v}_j^{subj} = \; & \mathbf{mannen}_i \otimes \mathbf{die}_{ijkl} \otimes \mathbf{vrouwen}_m \otimes \mathbf{haten}_{mkl} \qquad \text{(relabeled)} \end{aligned} \tag{5}$$

$$\begin{aligned} & \mathbf{mannen}_i \otimes \mathbf{die}_{jklm} \otimes \mathbf{vrouwen}_n \otimes \mathbf{haten}_{opq} \xrightarrow{\delta_{j,r,o,q,p}^{i,k,l,m,n}} \mathbf{v}_r^{obj} \in \mathsf{N} \\ \mathbf{v}_j^{obj} = \; & \mathbf{mannen}_i \otimes \mathbf{die}_{ijkl} \otimes \mathbf{vrouwen}_m \otimes \mathbf{haten}_{kml} \qquad \text{(relabeled)} \end{aligned} \tag{6}$$

The picture in Figure 3 expresses this graphically.

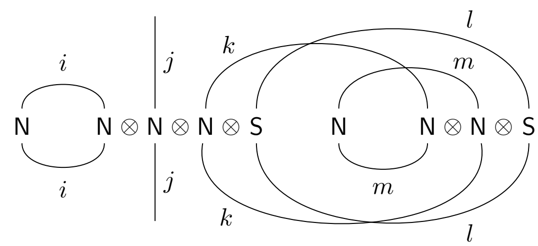

Figure 3: Matching diagrams for Dutch derivational ambiguity. Object relative (top), mannen_i die_{ijkl} $\mathsf{vrouwen}_m$ haten_{kml} versus subject relative (bottom) mannen_i die_{ijkl} $\mathsf{vrouwen}_m$ haten_{mkl}.

Open class items vs function words For open class lexical items, concrete meanings are obtained distributionally. For function words, the relative pronoun in this case, it makes more sense to assign them an interpretation independent of distributions. To capture the intersective interpretation of restrictive relative clauses, Sadrzadeh et al [9] propose to interpret the relative pronoun with a map that extracts a vector in the noun space from the relative clause body, and then combines this by elementwise multiplication with the vector for the head noun. Their account depends on the identification $\lceil np \rceil = \lceil n \rceil = \mathsf{N}$: noun phrases and simple common nouns are interpreted in the same space; it expresses the desired meaning recipe for the relative pronoun with the aid of (some of) the Frobenius operations that are available in a compact closed category:

$$\Delta : A \to A \otimes A \quad \mu : A \otimes A \to A \quad \iota : A \to I \quad \zeta : I \to A \tag{7}$$

In the case of **FVect**, Δ takes a vector and places its values on the diagonal of a square matrix, whereas μ extracts the diagonal from a square matrix. The ι and ζ maps respectively sum the coefficients of a vector or introduce a vector with the value 1 for all of its coefficients.

$$
\begin{array}{llccc}
\Delta_V : V \to V \otimes V & \text{given by} & \sum_i v_i \vec{e}_i & \mapsto & \sum_i v_i (\vec{e}_i \otimes \vec{e}_i) \\
\iota_V : V \to \mathbb{R} & \text{given by} & \sum_i v_i \vec{e}_i & \mapsto & \sum_i v_i \\
\mu_V : V \otimes V \to V & \text{given by} & \sum_{ij} v_{ij} (\vec{e}_i \otimes \vec{e}_j) & \mapsto & \sum_i v_{ii} \vec{e}_i \\
\zeta_V : \mathbb{R} \to V & \text{given by} & \lambda & \mapsto & \sum_i \lambda \vec{e}_i
\end{array}
$$

The analysis of [9] uses a pregroup syntax and addresses relative clauses in English. It relies on distinct pronoun types for subject and object relativisation. In the subject relativisation case, the pronoun lives in the space $\mathsf{N} \otimes \mathsf{N} \otimes \mathsf{S} \otimes \mathsf{N}$, corresponding to $n^r\, n\, s^l\, np$, the pregroup translation of a Lambek type $(n\backslash n)/(np\backslash s)$; for object relativisation, the pronoun lives in $\mathsf{N} \otimes \mathsf{N} \otimes \mathsf{N} \otimes \mathsf{S}$, corresponding to $n^r\, n\, np^{ll}\, s^l$, the pregroup translation of $(n\backslash n)/(s/np)$.

For the case of Dutch, the homomorphism $\lceil \cdot \rceil$ of §3.1 sends the relative pronoun type $(n\backslash n)/(\Diamond\Box np\backslash s)$ to the space $\mathsf{N}\otimes\mathsf{N}\otimes\mathsf{N}\otimes\mathsf{S}$. This means we can import the pronoun interpretation for that space from [9], which now will produce both the subject and object relativisation interpretations through its interaction with the derivational semantics.

$$\mathbf{die} \quad = \quad (1_{\mathsf{N}} \otimes \mu_{\mathsf{N}} \otimes 1_{\mathsf{N}} \otimes \zeta_{\mathsf{S}}) \circ (\eta_{\mathsf{N}} \otimes \eta_{\mathsf{N}}) \tag{8}$$

Intuitively, the recipe (8) says that the pronoun consists of a cube (in $\mathsf{N} \otimes \mathsf{N} \otimes \mathsf{N}$) which has 1 on its diagonal and 0 elsewhere, together with a vector in the sentence space S with all its entries 1. Substituting this lexical recipe in the tensor contraction equations of (5) and (6) yields the desired final semantic values (9) and (10) for subject and object relativisation respectively. We write $\odot$ for elementwise multiplication; the summation over the S dimension reduces the rank-3 $\mathsf{N} \otimes \mathsf{N} \otimes \mathsf{S}$ interpretation of the verb to a rank-2 matrix in $\mathsf{N} \otimes \mathsf{N}$, with rows for the verb's object, columns for the subject. This matrix is applied to the vector **vrouwen** either forward in (10), where 'vrouwen' plays the subject role, or backward in (9) before being elementwise multiplied with the vector for mannen.

$$(5) \quad = \quad \mathbf{mannen} \odot \Big[\Big(\sum_S \mathbf{haten}\Big)^{\mathsf{T}} \mathbf{vrouwen}\Big] \tag{9}$$

$$(6) \quad = \quad \mathbf{mannen} \odot \Big[\Big(\sum_S \mathbf{haten}\Big) \mathbf{vrouwen}\Big] \tag{10}$$

Returning to English, notice that the pregroup type assignment $n^r\, n\, np^{ll}\, s^l$ for object relativisation in [9] is restricted to cases where the 'gap' in the relative clause body occupies the final position. To cover these non-subject relativisation patterns in general, also with respect to positions internal to the relative clause body, we would use an $\mathbf{NL}_\Diamond$ type $(n\backslash n)/(s/\Diamond\Box np)$ for the pronoun, together with the rightward extraction postulates $\alpha^r_\Diamond, \sigma^r_\Diamond$ of Figure 1. For English subject relativisation, the simple pronoun type $(n\backslash n)/(np\backslash s)$ will do, as this pattern doesn't require any structural reasoning.

4 Discussion

We briefly compare the distributional and the formal semantics accounts, highlighting their similarities. In the formal semantics account, the interpretation homomorphism sends syntactic types to their semantic counterparts. Syntactic types are built from atoms, for example s, np, n for sentences, noun phrases and common nouns; assuming semantic atoms e, t and function types built from them, one can set $\lceil s\rceil = t$, $\lceil np\rceil = e$, $\lceil n\rceil = e \to t$, and $\lceil A/B\rceil = \lceil B\backslash A\rceil = \lceil B\rceil \to \lceil A\rceil$. Each semantic type A is assigned an interpretation domain D_A, with $D_e = E$, for some non-empty set E (the discussion domain), $D_t = \{0,1\}$ (truth values), and $D_{A\to B}$ funtions from D_A to D_B.

In this setup, a syntactic derivation $A_1, \ldots A_n \Rightarrow B$ is interpreted by means of a linear lambda term M of type $\lceil B\rceil$, with parameters x_i of type $\lceil A_i\rceil$ — linearity resulting from the fact that the syntactic source doesn't provide the copying/deletion operations associated with the structural rules of Contraction and Weakening.

As in the distributional model discussed here, the proof term M is an instruction for meaning assembly that abstracts from lexical semantics. In (11) below, one finds the proof terms for English subject (a) and object (b) relativisation. The parameter w stands for the head noun, f for the verb, y and z for its object and subject arguments; parameter x for the relative pronoun has type $(e \to t) \to (e \to t) \to e \to t$.

$$\begin{array}{lll} (a) & n, (n\backslash n)/(np\backslash s), (np\backslash s)/np, np \Rightarrow n & (x_{who}\ \lambda z^e.(f^{e\to e\to t}\ y^e\ z^e)\ w^{e\to t}) \\ (b) & n, (n\backslash n)/(s/np), np, (np\backslash s)/np \Rightarrow n & (x_{who}\ \lambda y^e.(f^{e\to e\to t}\ y^e\ z^e)\ w^{e\to t}) \end{array} \tag{11}$$

To obtain the interpretation of 'men who hate women' vs 'men who(m) women hate', one substitutes lexical meanings for the parameters of the proof terms. In the case of the open class items 'men', 'hate', 'women', these will be non-logical constants with an interpretation depending on the model. For the relative pronoun, we substitute an interpretation independent of the model, expressed in terms of the logical constant $\wedge$, leading to the final interpretations of (13), after normalisation.

$$x_{who} := \lambda x^{e\to t}\lambda y^{e\to t}\lambda z^e.((x\ z) \wedge ((y\ z)) \tag{12}$$

$$\begin{array}{ll} (a) & \lambda x.((\textsc{men}\ x) \wedge (\textsc{hate women}\ x)) \\ (b) & \lambda x.((\textsc{men}\ x) \wedge (\textsc{hate}\ x\ \textsc{women})) \end{array} \tag{13}$$

Notice that the lexical meaning recipe for the relative pronoun goes beyond linearity: to express the set intersection interpretation, the bound z variable is copied over the conjuncts of $\wedge$. By encapsulating this copying operation in the lexical semantics, one avoids compromising the derivational semantics. In this respect, the formal semantics account makes the same design choice regarding the division of labour between derivational and lexical semantics as the distributional account, where the extra expressivity of the Frobenius operations is called upon for specifying the lexical meaning recipe for the relative pronoun.

5 Acknowledgments

We thank Giuseppe Greco for comments on an earlier version. The second author would also like to thank Mehrnoosh Sadrzadeh for the many discussions on compositional distributional semantics and Frobenius operations, and Rob Klabbers for his interesting remarks on index notation. The second author gratefully acknowledges support by a Queen Mary Principal's Research Studentship, the first author the support of the Netherlands Organisation for Scientific Research (NWO, Project 360-89-070, *A composition calculus for vector-based semantic modelling with a localization for Dutch*).

References

[1] Marco Baroni, Raffaela Bernardi, and Roberto Zamparelli. Frege in Space: a Program for Compositional Distributional Semantics. *Linguistic Issues in Language Technology*, 9:241–346, 2014.

[2] Bob Coecke, Edward Grefenstette, and Mehrnoosh Sadrzadeh. Lambek vs. Lambek: Functorial Vector Space Semantics and String Diagrams for Lambek Calculus. *Annals of Pure and Applied Logic*, 164(11):1079–1100, 2013.

[3] Joachim Lambek. On the Calculus of Syntactic Types. In Roman Jakobson, editor, *Structure of Language and its Mathematical Aspects*, volume XII of *Proceedings of Symposia in Applied Mathematics*, pages 166–178. American Mathematical Society, 1961.

[4] Joachim Lambek. Categorial and Categorical Grammars. In Richard T. Oehrle, Emmon Bach, and Deirdre Wheeler, editors, *Categorial Grammars and Natural Language Structures*, volume 32 of *Studies in Linguistics and Philosophy*, pages 297–317. Reidel, 1988.

[5] Jean Maillard, Stephen Clark, and Edward Grefenstette. A Type-Driven Tensor-Based Semantics for CCG. In *Proceedings of the Type Theory and Natural Language Semantics Workshop*, pages 46–54. EACL, 2014.

[6] Michael Moortgat. Multimodal Linguistic Inference. *Journal of Logic, Language and Information*, 5(3-4):349–385, 1996.

[7] Michael Moortgat. Constants of Grammatical Reasoning. In G Bouma, E Hinrichs, G.-J. Kruijff, and R.T. Oehrle, editors, *Constraints and Resources in Natural Language Syntax and Semantics*, pages 195–219. CSLI, 1999.

[8] Michael Moortgat and Richard Moot. Proof Nets for the Lambek-Grishin Calculus. In Chris Heunen, Mehrnoosh Sadrzadeh, and Edward Grefenstette, editors, *Quantum Physics and Linguistics: A Compositional, Diagrammatic Discourse*, pages 283–320. Oxford University Press, 2013.

[9] Mehrnoosh Sadrzadeh, Stephen Clark, and Bob Coecke. The Frobenius Anatomy of Word Meanings I: Subject and Object Relative Pronouns. *Journal of Logic and Computation*, pages 1293–1317, 2013.

[10] Gijs Wijnholds. Categorical Foundations for Extended Compositional Distributional Models of Meaning. *MSc. Thesis*, Universiteit van Amsterdam, 2014.