Tree Models and (Labeled) Categorial Grammar

Yde Venema
Department of Mathematics and Computer Science
Free University
De Boelelaan 1081a
1081 HV Amsterdam
e-mail: yde@cs.vu.nl

Abstract
This paper studies the relation between some extensions of the non-associative Lambek Calculus NL and their interpretation in tree models (free groupoids). We give various examples of sequents that are valid in tree models, but not derivable in NL. We argue why tree models may not be axiomatizable if we add finitely many derivation rules to NL, and proceed to consider labeled calculi instead.

We define two labeled categorial calculi, and prove soundness and completeness for interpretations that are 'almost' the intended one, namely for tree models where some branches of some trees may be resp. all branches of all trees must be infinitely extending. Extrapolating from the experiences in our quite simple systems, we briefly discuss some problems involved with the introduction of labels in categorial grammar, and argue that many of the basic questions are not yet understood.

1 Introduction

For a long time, the associative Lambek calculus has been the predominant formalism in categorial grammar, and language models (free semigroups, string models) its standard model-theoretic interpretation. Recent years however have seen a proliferation of both alternative calculi and alternative interpretations. The reasons for this development stem from both logic and linguistic origins. In logic for instance Lambek's calculus has found itself surrounded by a whole landscape of of so-called *substructural logics* (cf. DOŠEN & SCHRÖDER-HEISTER [7]), and also connections with modal logic have been investigated (cf. VAN BENTHEM [1]); in linguistics, it was realized that the Lambek calculus is not a suitable device for studying phenomena like discontinuous constituency or head dependency (cf. MOORTGAT [17]).

The aim of this paper is to contribute to both model theory and proof theory of categorial grammar (with product) by studying a very simple example in detail. In order to formulate the motivation for writing this paper more precisely, let us start with a formal definition of this problem:

Definition 1 *Given a set Pr of primitive types, the set $Tp(Pr)$ of types is formed by closing Pr under the binary connectives $\circ$ ('times'), $/$ ('over') and $\backslash$ ('under'). A sequent is of the form $X \longrightarrow A$ with X a term and A a type; here the set of* **terms** *is defined as the closure of $Tp(Pr)$ under the structural connective $(\cdot,\cdot)$.*

We are interested in the following semantics for this language. Consider a set L of elements called **leaves**. *$\mathtt{Tree}(L)$, the set of (ordered)* **trees over** *L, is defined as follows: any leaf is a tree, and if s and t are trees, then so is (st). A* **finite-tree model**, *or shortly: a* **fintree model** *is a pair $\mathfrak{M} = ((\mathtt{Tree}(L), V)$ where V is an* **interpretation** *mapping basic types to subsets of $\mathtt{Tree}(L)$. V can be extended to types and terms as follows:*

$$
\begin{array}{lcl}
V(A \circ B) & = & \{(st) \in \mathtt{Tree}(L) \mid s \in V(A), t \in V(B)\} \\
V(A/B) & = & \{s \in \mathtt{Tree}(L) \mid \ (st) \in V(A) \textit{ for all trees } t \textit{ with } t \in V(B)\ \} \\
V(A\backslash B) & = & \{s \in \mathtt{Tree}(L) \mid \ (ts) \in V(B) \textit{ for all trees } t \textit{ with } t \in V(A)\ \} \\
V(X,Y) & = & \{(st) \in \mathtt{Tree}(L) \mid s \in V(X), t \in V(Y)\}.
\end{array}
$$

We usually denote $s \in V(A)$ by $\mathfrak{M}, s \Vdash A$, or if no confusion arises, by $s \Vdash A$. We also use terminology from modal logic, like 'A is **true** *at s' for '$s \Vdash A$'. A sequent $X \longrightarrow A$* **holds** *in a model $\mathfrak{M}$, notation: $\mathfrak{M} \models X \longrightarrow A$, if $V(X) \subseteq V(A)$; it is* **valid** *in the class of finite-tree frames, notation: $\mathsf{T_f} \models X \longrightarrow A$, if it holds in every finite-tree model.*

These models have occurred under various names in the literature, like *bracketed strings*, *free groupoids*, *non-associative category hierarchies*, etc. The central problem of the paper can now be formulated concisely as follows.

Problem 1 *Can we find a 'nice' calculus recursively enumerating all sequents* Γ *for which* $\mathsf{T_f} \models \Gamma$*?*

Here 'nice' refers to properties like cut-elimination or decidability. Note that Kandulski showed in [13] that the analogous problem for the the *product-free* language allows a quite easy positive solution, cf. Theorem 3.

Let us hasten to remark that this problem itself will be too simple to be of *direct* linguistic interest, althought it is obviously important from the perspective of both linguistics and mathematics. It is interesting to note that there are some recent developments that witness the relevance of Problem 1 for the linguistic side of the categorial grammar framework. The example that we are referring to is that of the treatment of discontinuous constituency in the framework of categorial grammar. It would go a bit too far to discuss this phenomenon in detail here, so we confine ourselves to a brief explanation.

To overcome the difficulties of the traditional Lambek calculus in handling this phenomenon, extensions of the categorial language with new type constructors have been proposed, cf. Moortgat [17]. It turned out that these new connectives do not find a natural surrounding in a string-based approach, cf. Versmissen [30]. For instance, the associativity of the structural connective in Lambek's calculus seems to make it impossible to formulate a pair of natural left and right operational rules for Moortgat's infixation ($\downarrow$) and extraction ($\uparrow$) operators. The basic problem seems to be that a *string* of words does not have a unique point where a second string can be inserted. However, if we study finite-tree models in which every node of a tree has a distinguished *head* daughter, we can equip any tree with such a unique insertion point, viz. immediately before or immediately after the head of the tree. In [21], Moortgat and Oehrle give a nice inductive definition of a head wrapping operation on trees, thus providing a unified categorial framework for headedness and discontinuous constituency. Note that a solution to our Problem 1 would be a first step towards a proof calculus for Moortgat & Oehrle's system.

However, the above mentioned example is definitely not the only source of inspiration for studying Problem 1. There are in fact *two* more *kinds* of motivation.

The first one is a purely mathematical one, inspired by developments in the model theory of categorial grammars. For a long time it has been one of the outstanding open questions in this field whether the associative Lambek calculus is not only sound but also complete with respect to the interpretation in language models. Recently, this question has been answered affirmatively by M. Pentus (cf. [26]). The obvious counterpart of this question is whether a similar completeness result holds for the non-associative Lambek calculus *NL* with respect to tree models (free groupoids). Now *this* problem has been solved already a few years ago — in the negative, cf. Došen [8]), but this negative answer now triggers the question whether we can *extend NL* with some simple axioms and/or derivation rules in order to obtain completeness. Note that this really is an instance of our Problem 1.

The last kind of motivation takes us to an area of logical proof theory which has has become rather active lately, viz. that of labeled deductive systems. The basic idea of a labeled deductive system is that the *structure* of the 'database' of assumptions $A_1, \ldots, A_n$ in a consequence relation

$$A_1, \ldots, A_n \longrightarrow B$$

is made *explicit* by labeling the types:

$$x_1 : A_1, \ldots, x_n : A_n \longrightarrow y : B.$$

This idea has been around in categorial grammar for some time already (cf. Buszkowski [2]), but seems to be taking off after Gabbay introduced his Labeled Deductive Systems as a general framework for reasoning with labels (cf. Gabbay [9]), and Oehrle suggested a multi-dimensional approach to formal linguistics (cf. [24]), in which linguistic objects are represented as tuples, with each coordinate providing information on a specific aspect like prosodic form or semantic meaning. In a categorial grammar framework labels seem to be the perfect vehicles to carry information other than the syntactic type, as was observed by Moortgat (cf. [19]). Grosso modo, there is the following important distinction to be made here as to the impact of the labels in the calculus.

If the labels just *follow* the proof, for instance in order to generate the meaning of a sentence fragment, we are just confronting (a generalization of) the well-known Curry-Howard isomorphism. On the other hand, in most of the recently developed systems the labels play a far more active role. For instance, Morrill & Solias [22] and Hepple [11] intend to solve precisely the above-mentioned problem of discontinuous constituency by formulating, in the equational theory of the label algebra, side conditions on the application of operational rules. The logical aspects of *such* applications of Labeled Deductive Systems are as yet largely unknown, although some first exercises have been carried out, witness (besides the papers cited above), Chau [4], Kurtonina [14], Roorda [27, 28].

The main aim of this paper is to take a few more steps in the area of labeled categorial grammar; We will investigate some model-theoretical and proof-theoretical properties of a few labeled calculi.

Overview In the next section we will approach our Problem from a naive point of view. The basic idea in this section is to investigate whether a simple extension of the non-associative Lambek calculus might yield the desired completeness result. First we will define a hierarchy of frame classes generalizing the class of fintree models, for instance T (tree frames, i.e. where some trees may have infinite branches) and T_∞ (inftree frames, i.e. where all branches of a tree are infinite). In the first part of the section we will review some nice completeness results for *NL* itself, but then we give various examples as to why *NL* and some of its intuitive finite extensions will not be complete with respect to any class of tree frames. We leave it as an open problem whether our classes of tree frames allow a *finite* axiomatization in a 'pure' sequent calculus.

In section 3 we turn to *labeled* categorial grammars instead. For both the classes T_∞ and T we will develop sound and complete labeled calculi LC_∞ and LC_t, LC_∞ being nice in the sense that it allows a cut elimination theorem. Our main problem, viz. whether the class of frames of *finite* trees has a (nice) complete axiomatization, remains open. We finish with a short section discussing some problems involved with the introduction of labels in categorial grammar, our main conclusion being that the logical foundations of the area seem to be unexplored as yet. We briefly return to the linguistic questions that brought about the research reported on in this paper.

2 Incompleteness for calculi without labels

In this section we will start looking for a complete calculus for (fin)tree frames by various 'naive' adaptations of the non-associative Lambek calculus *NL*. In the first subsection we will give some positive results concerning *NL*, in the last part of the section we will argue why this naive approach is unlikely to work.

2.1 The non-associative Lambek calculus

As we already mentioned in the introduction, the non-associative Lambek Calculus seems to be the natural starting point to look for a complete calculus — below we give a formal definition of this calculus *NL*. In the formulation of the derivation rules, $X[B]$ denotes a term X with a distinguished occurrence of the type B; when used in the same rule, $X[Y]$ denotes the term X with the term Y substituted for the distinguished occurrence of B. Outermost parenthesis of terms will frequently be omitted.

Definition 2 *The sequent derivation system of the non-associative Lambek Calculus NL is given by the following* **logical axiom** *and* **logical rule***:*

$$\frac{}{A \longrightarrow A}\ [Id] \qquad \frac{X \longrightarrow A \quad Y[A] \longrightarrow B}{Y[X] \longrightarrow B}\ [Cut]$$

and the following **operational rules** *for the three connectives:*

$$\frac{X[B] \longrightarrow C \quad Y \longrightarrow A}{X[Y, A\backslash B] \longrightarrow C}\ [\backslash L] \qquad \frac{(A, X) \longrightarrow B}{X \longrightarrow A\backslash B}\ [\backslash R]$$

$$\frac{X[B] \longrightarrow C \quad Y \longrightarrow A}{X[B/A, Y] \longrightarrow C}\ [/L] \qquad \frac{(X, A) \longrightarrow B}{X \longrightarrow B/A}\ [/R]$$

$$\frac{X[A, B] \longrightarrow C}{X[A \circ B] \longrightarrow C}\ [\circ L] \qquad \frac{X \longrightarrow A \quad Y \longrightarrow B}{(X, Y) \longrightarrow A \circ B}\ [\circ R]$$

Note that NL has **no structural rules**. *Finally, notions like* **derivability** *and* **theorems** *are defined as usual.*

NL is the weakest logic in the landscape of so-called substructural logics, cf. DOŠEN [7] (at least, if one does not take systems like the head-dependency calculus of MOORTGAT & MORRILL [20] or ZIELONKA [31] into account). In fact it can be seen as the pure system of *residuation*, cf. the algebraic inequalities below:

$$A \longrightarrow C/B \text{ iff } A \circ B \longrightarrow C \text{ iff } B \longrightarrow A\backslash C.$$

It is well-known that the above schema has a natural reading in the power set algebra of relational structures. This gives an easy completeness result for *NL*, but an interesting one, as it forms the basis for our further investigations in this section.

Definition 3 *A* **(relational) frame** *is a pair* $\mathfrak{F} = (W, R)$ *with* R *a ternary* **accessibility relation** *on* W*. Adding an* **interpretation** $V : Pr \mapsto \mathcal{P}(W)$*, we obtain a* **(relational) model**. **Truth** *of types (and terms) is defined as follows:*

$$\begin{array}{lcl} V(A \circ B) & = & \{x \in W \mid (\exists yz)\ Rxyz, y \in V(A)\ \&\ z \in V(B)\} \\ V(A/B) & = & \{y \in W \mid (\forall xz)\ Rxyz\ \&\ z \in V(B) \textit{ imply } x \in V(A)\} \\ V(A\backslash B) & = & \{z \in W \mid (\forall xy)\ Rxyz\ \&\ y \in V(A) \textit{ imply } x \in V(B)\} \\ V(Y, Z) & = & \{x \in W \mid (\exists yz)\ Rxyz, y \in V(Y)\ \&\ z \in V(Z)\} \end{array}$$

The class of relational models is denoted by R.

The trivial but crucial connection with finite-tree models is that a finite-tree frame becomes a relational frame by putting

$$Rstu \iff s = (tu).$$

So, the following theorem states that *NL* is at least sound with respect to $\mathsf{T_f}$ and complete with respect to a *superclass* of it:

Theorem 1 *NL is sound and complete with respect to* R.

Proof.
Soundness is left to the reader. The rather easy proof of the completeness direction goes by a canonical model construction. Let W be the set of deductively closed sets of types (i.e. $\alpha \in W$ iff $A' \in \alpha$ whenever $A \in \alpha$ and $NL \vdash A \longrightarrow A'$); the accessibility relation R is defined by

$$R\alpha\beta\gamma \iff A \in \alpha \text{ whenever } B \in \beta, C \in \gamma \text{ and } NL \vdash B \circ C \longrightarrow A,$$

and the interpretation V by $V(p) = \{\alpha \mid p \in \alpha\}$. By induction to the complexity of A it is proved that $A \in \alpha \iff \alpha \Vdash A$. This implies that the canonical model is a counter model for every non-theorem of NL. □

So, to find a calculus for tree models, it might be a useful strategy to try and bridge the gap between R and $\mathsf{T_f}$. Let us define some new classes of frames:

Definition 4 *A* **groupoid** *is a pair $\mathfrak{G} = (G, \cdot)$ with $\cdot$ a binary operation on G. As before with tree frames, we may see $\mathfrak{G}$ as a special kind of relational frame by putting $Rstu \iff s = tu$. The class of groupoid frames is denoted by G. If in a groupoid frame, $x = yz$, we call y a* **left** *and z a* **right daughter** *of x.*

A **tree frame** *is a groupoid frame satisfying* **unique splittability**

$$(US) \qquad (Rwuv \;\&\; Rwu'v') \Rightarrow (u = u' \;\&\; v = v')$$

and **acyclicity**

$$(AC) \qquad \textit{for no distinct } x_0, \ldots, x_n \textit{ do we have } x_0Ex_1 \ldots x_{n-1}Ex_nEx_0$$

where E is the relation defined by xEy iff x is a daughter of y, or y of x.

A tree frame is a **fintree frame** *if it satisfies* **converse wellfoundedness** *(CW): there are no infinite paths $x_0Mx_1Mx_2M\ldots$ where xMy if y is a daughter of x. A tree frame is an* **inftree frame** *if every node has daughters, i.e. if it satisfies (S): $\forall x \exists yz\,(x = yz)$.*

The classes of tree frames, fintree frames and inftree frames are denoted by resp. T, $\mathsf{T_f}$ and $\mathsf{T_\infty}$.

We leave it to the reader to verify that the class of fintree frames defined above coincides with the class of finite-tree frames, up to isomorphism. In a Venn-diagram we can depict these classes of relational frames as follows:

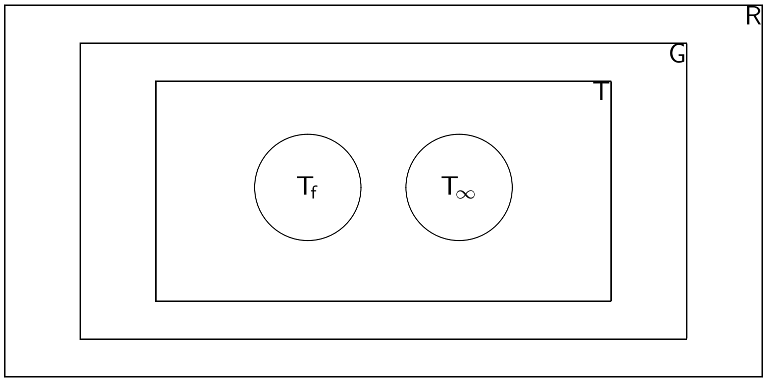

Figure 1.

Of course, these classes need not all have different theories — in fact, a consequence of the following result is that a sequent is valid in all groupoids iff it is valid in all relational frames:

Theorem 2 *NL is sound and complete with respect to* G*.*

The proof of this theorem is a straightforward adaptation of Theorem 2 in Buszkowski [2]), viz. that the (associative) Lambek Calculus is complete with respect to semigroup semantics. The basic idea in both proofs is to use an intermediate *labeled* natural deduction system (cf. also Kandulski [13]).

R and G form the only pair of frame classes in our list with identical categorial theories. In the next subsection we will give sequents separating the other classes. All these examples witness the incompleteness of *NL* with respect to (fin)tree semantics. Before moving to these negative results, let us mention a few more positive facts concerning *NL*.

Note that the existence of two notably different frame classes having identical theories is an indication of the weakness of a language. Dropping connectives from the language may leave us with a formalism of even less discriminating power. For instance, we have an interesting result for the language without product:

Proposition 5 *Let* $\mathfrak{F}, \mathfrak{F}'$ *be two groupoid frames such that* $\mathfrak{F}'$ *is a homomorphic image of* $\mathfrak{F}$*. If* $\circ$ *does not occur in the sequent* $X \longrightarrow A$*, then*

$$\mathfrak{F} \models X \longrightarrow A \textit{ implies } \mathfrak{F}' \models X \longrightarrow A.$$

Proof.
We reason by contraposition: assume that $X \longrightarrow A$ is *not* valid in $\mathfrak{F}'$. Then there are an interpretation V' and a world w' in $\mathfrak{F}'$ such that $\mathfrak{F}', V', w' \Vdash$

X and $\mathfrak{F}', V', w' \not\models A$. As a representative example, let X be of the form $(A_0, (A_{10}, A_{11}))$; then there are worlds w'_0, w'_1, w'_{10} and w'_{11} such that $w' = w'_0 \cdot w'_1$, $w'_1 = w'_{10} \cdot w'_{11}$ and $w'_i \Vdash A_i$. Now let w_0, w_{10} and w_{11} in $\mathfrak{F}$ be such that $fw_0 = w'_0$, etc. Define $w_1 := w_{10} \cdot w_{11}$ and $w := w_0 \cdot w_1$, then by the fact that f is a homomorphism, $fw_1 = w'_1$ and $fw = w'$.

Finally, define the following interpretation V on $\mathfrak{F}$:

$$V(p) := \{x \in W \mid fx \in V'(p)\}.$$

We now prove by induction on the complexity of $/, \backslash$-types that

$$(*) \qquad x \Vdash B \iff fx \Vdash B.$$

The base step of $(*)$ is immediate by definition of V.

For the induction step, we only consider the case where B is of the form C/D. First assume that $x \Vdash C/D$; to show that $fx \Vdash C/D$, let y' be such that $y' \Vdash D$. As f is surjective, $y' = fy$ for some y in $\mathfrak{F}$. The induction hypothesis gives that D is true at this y. Then $xy \Vdash C$, so by the induction hypothesis again, we find $f(xy) \Vdash C$. But $f(xy) = fx \cdot fy = fx \cdot y'$. As y' was arbitrary, this gives $fx \Vdash C/D$.

For the other direction, let x be such that $fx \Vdash C/D$. Take an arbitrary y in W with $y \Vdash D$; then $fy \Vdash D$ by the induction hypothesis. The truth definition of $/$ gives $f(xy) \Vdash C$, so by the induction hypothesis we get $xy \Vdash C$. This implies $x \Vdash C$.

To finish the proof, $(*)$ gives that $w \Vdash X$ while $w \nVdash A$. So $\mathfrak{F} \not\models X \longrightarrow A$. $\Box$

The above proposition allows us to give a new proof of the following theorem. It states that if we confine ourselves to the $/, \backslash$-language, *NL* is strong enough to capture the sequent logic of $\mathsf{T_f}$:

Theorem 3 (Kandulski) *NL*$(/, \backslash)$ *is sound and complete with respect to* $\mathsf{T_f}$.

Proof.
Although the original proof of this result (Theorem 1.1 in KANDULSKI [13]) is quite easy, we enjoy showing it to be an immediate corollary of Theorem 2 and Proposition 5. The key fact is that fintree frames precisely constitute the class of *free* groupoids, whence every groupoid frame is a homomorphic image of a (sufficiently large) fintree frame. $\Box$

2.2 Tree models

In this subsection we will go into some detail as to why the non-associative Lambek calculus is not complete with respect to tree models. The following example was given by Došen in [8]: define the sequent

$$(\Gamma_0) \qquad p, p\backslash(q \circ r) \longrightarrow p \circ r.$$

It will be clear that this sequent is not derivable in *NL*. However, it is valid in T, as a simple but instructive argument shows:

> Let $\mathfrak{M}$ be a model based on a tree frame, and assume that $t \in V(p, p\backslash(q \circ r))$. Then t has daughters t_0 and t_1 such that $t_0 \Vdash p$ and $t_1 \Vdash p\backslash(q \circ r)$. By the truth definition, $t \Vdash q \circ r$, so by the truth definition again, *and the fact that* t_0 *and* t_1 *are uniquely determined as the daughters of* t, we find that $t_0 \Vdash q$ and $t_1 \Vdash r$. But then we have $t \Vdash p \circ r$.

Clearly, the essential property used here is that of Unique Splittability (cf. Definition 3).

Following a suggestion by ZIELONKA [32], we could try to capture the fact that Γ_0 should be derivable in the logic we are heading for, by adding the following proof rule to *NL*:

$$\frac{X \longrightarrow A \circ B \quad X \longrightarrow A' \circ B'}{X \longrightarrow A \circ B'}\ [S]$$

and indeed, it is easy to show that in the resulting calculus NL_S, Γ_0 is derivable. Zielonka raises the question, whether NL_S is complete with respect to tree models. Unfortunately, we have to answer this question in the negative, viz. the following sequent:

$$(\Gamma_1) \qquad p\backslash(q \circ r), p \longrightarrow r \circ p.$$

Γ_1 is not an NL_S-theorem:

> An easy proof shows that NL_S is *sound* with respect to the class of R-frames satisfying (US). However, Γ_1 is not valid in every US-frame, as the following counter example (W, R, V) witnesses: $W = \{a, b, c\}$, $R = \{(a, b, c)\}$ and $V(p) = \{c\}$, $V(q) = V(r) = \varnothing$. It is immediate that $b \Vdash p\backslash(q \circ r)$, so $a \in V(p\backslash(q \circ r), p)$, while $a \not\Vdash r \circ p$.

On the other hand, Γ_1 is valid in T:

> Let $\mathfrak{M}$ be a model based on a tree frame, and assume that $t \in V(p\backslash(q \circ r), p)$. Then t has daughters t_0 and t_1 such that $t_0 \Vdash p\backslash(q \circ r)$ and $t_1 \Vdash p$. Now the tree $(t_1 t_0)$ exists as well, and for this tree we have $(t_1 t_0) \in V(p, p\backslash(q \circ r))$. But we already showed that in this situation, ($t_1 \Vdash q$ and) $t_0 \Vdash r$. So we find that indeed $t \Vdash r \circ p$.

The essential property of tree models that we used in this argument (besides Unique Splittability) is the fact that the tree forming operation is a *total* function.

Heading for a calculus complete with respect to tree models, we might follow a naive idea and add the following proof rule to the calculus NL_S:

$$\frac{(Y, X) \longrightarrow B \circ A}{(X, Y) \longrightarrow A \circ B} \; [F_2]$$

It will be clear how to prove Γ_1 from the resulting calculus NL_{SF_2}:

$$\dfrac{\dfrac{\dfrac{p \longrightarrow p \quad p\backslash(q \circ r) \longrightarrow p\backslash(q \circ r)}{p, p\backslash(q \circ r) \longrightarrow p \circ (p\backslash(q \circ r))} [\circ R] \quad \dfrac{p \longrightarrow p \quad q \circ r \longrightarrow q \circ r}{p, p\backslash(q \circ r) \longrightarrow q \circ r} [\backslash L]}{p, p\backslash(q \circ r) \longrightarrow p \circ r} [S]}{p\backslash(q \circ r), p \longrightarrow r \circ p} [F_2]$$

However, NL_{SF_2} is not complete with respect to tree models. To see this, let us look with a bit more care at the proof as to why Γ_1 is valid in tree models. The crucial observation was that the two daughters t_0 and t_1 of the 'current' tree t, can be combined to form a new tree. However, the totality of the tree forming operation implies that *every* two elements of the frame can be combined to form a tree. This means that for instance, the sequent

$$(\Gamma_1') \qquad (p\backslash(q \circ r), s), p \longrightarrow (r \circ s) \circ p$$

is valid in tree frames too :

> To show why this is so, one now combines t's *granddaughter* t_{00} (for which $t_{00} \Vdash p\backslash(q \circ r)$) with t's daughter t_1 (where p is true) to a new tree $(t_1 t_{00})$, and proceeds with the earlier argument.

On the other hand, Γ_1' is not derivable in NL_{SF_2}:

> One proves this by first showing NL_{SF_2} to be sound with respect to the class of frames satisfying (US) and (FC_2): $\forall xy(\exists z Rzxy \leftrightarrow \exists z Rzyx)$. Then one inspects the following '(US)&(FC_2)-model': $W = \{a, b, c, (ab), (ba), (ab)c, (ba)c, c(ab), c(ba)\}$; R is defined in the obvious way and V is given by $V(p) = \{c\}$, $V(s) = \{b\}$ and $V(q) = V(r) = \varnothing$. Then we find $a \Vdash p\backslash(q \circ r)$, so $(ab) \in V((p\backslash(q \circ r), s))$, $(ab)c \in V((p\backslash(q \circ r), s), p)$ while $(ab)c \nVdash (r \circ s) \circ p$.

In the end, it seems that one would have to add *infinitely* many derivation rules to the system before even coming to think of completeness[1]. Putting it

[1] We state it as an open problem whether the sequents valid in tree frames are axiomatizable with a *finite* set of axioms and rules or not. We conjecture that the answer to this problem is negative. It may turn out to be difficult to prove this conjecture, since the usual methods to prove non-finite axiomatizability (like the use of ultra-products) do not seem to apply. We suspect that every finite derivation system only captures valid sequents up to a certain *depth* or *bracket complexity* (suitably defined), but we have not been able to formalize this idea.

differently, the essential difficulty seems to be that the following proposition holds for any tree t:

(∗) $t \Vdash p\backslash(q \circ r)$ implies $t \Vdash r$,
provided that *somewhere* in the frame there is a tree s with $s \Vdash p$.

Unfortunately, the sequent format of the calculus does not seem to be adequate to express this 'somewhere' concisely. There are several ways to try and solve this problem; for instance, one might think of *adding* an explicit 'somewhere' operator to the language, like in recent approaches to modal logic, cf. Goranko & Passy [10]. A disadvantage of this approach is that it does not fit nicely in the particular resource-sensitive paradigm of *NL*. It might be a better idea to go even further along the line of making information explicit that is already present in the sequent's antecedent. Note that if we evaluate a term (A, B) at a tree t, we know *exactly* where A and B have to hold: at the left resp. right daughter of t. So why not replace the antecedent (A, B) where this information is implicit, by a database $\{a_0 : A, a_1 : B\}$ where we have syntactic entities to refer explicitly to these daughters? In this way, (∗) can at least be *formulated* in the language, viz. as

$$\{a : p\backslash(q \circ r), b : p\} \longrightarrow a : r$$

This move takes us to the area of Labeled Deductive Systems (cf. Gabbay [9]), and will be worked out in detail in the following section.

Let us finish this section by giving some sequents discriminating the other frame classes of Fig. 1. To start with, the attentive reader may have noticed that we have been speaking about *tree* semantics in this section rather than about *finite-tree* frames. The reason for this is that even if we had found a calculus that is sound and complete with respect to T, this system would not do the job for $\mathsf{T_f}$. For, consider the following sequent

$$(\Gamma_2) \qquad p, p\backslash((p \circ \top) \circ \top) \longrightarrow \bot$$

which is valid in fintree frames, but not in every tree frame. (We use $\top$ and $\bot$ to indicate that *any* type may be substituted; so, the $\bot$ in the succedent says that the antecedent cannot be true in any tree.)

> To show that $\mathsf{T_f} \models \Gamma_2$, suppose that t is a tree in a fintree frame, such that the antecedent X of Γ_2 holds at t. Clearly $t_0 \Vdash p$ and $t_1 \Vdash p\backslash((p\circ\top)\circ\top)$, so by the truth definition, $t \Vdash (p\circ\top)\circ\top$. Unique Splittability gives $t_0 \Vdash p \circ \top$, whence t_0 has daughters t_{00} and t_{01}. Now we let the tree $t_{00}t_1$ take the place of t, observing that X is true at $t_{00}t_1$. We find that t_{00} has daughters too ... An inductive argument yields an infinite path $t_0, t_{00}, t_{000}, \ldots$, contradicting the fact that t should be *finite*.
>
> We leave it to the reader to give a counter example to the validity of Γ_2 in the class of arbitrary tree frames.

Note that in all the earlier examples, the types in the succedent appeared in the antecedent as well; however, the information used to prove that $\mathsf{T_f} \models \Gamma_2$ is far more implicit. Given this example, we fear that it may be hard to find a sequent axiomatization for $\mathsf{T_f}$, even in the labeled approach. Therefore, we decided to aim a bit lower, viz. a calculus for T instead of one for $\mathsf{T_f}$. Even this problem turned out to be harder than expected. The problem is that in a tree frame, leaves have a different behavior than trees with daughters. This is well illustrated by the following sequent

$$(\Gamma_3) \qquad q\backslash(q \circ ((p/q) \circ (q\backslash p)) \longrightarrow (p/q) \circ (q\backslash p)$$

which distinguishes inftree frames from tree frames.

> To see why Γ_3 is valid in an arbitrary inftree frame, assume that for an inftree t, $t \Vdash q\backslash(q \circ A)$, where A abbreviates the type $(p/q) \circ (q\backslash p)$. Now make a case distinction as to whether $V(q)$ is empty or not. If $V(q) = \varnothing$, then *any* tree s in the frame satisfies $s \Vdash p/q$ and $s \Vdash q\backslash p$. In particular, the daughters of t do. So $t \Vdash (p/q) \circ (q\backslash p)$. If on the other hand $V(q)$ has an element s, then by a now familiar argument, $t \Vdash q\backslash(q \circ A)$ implies $t \Vdash A$. Again, it is left to the reader to give a counter example to the validity of Γ_3 in the class of arbitrary tree frames.

So, inftree frames seem to be simpler to axiomatize because they are more homogeneous. Therefore, we will first give a sound and complete labeled calculus for the class T_∞.

3 Labeled categorial grammar

In this section we will define two extensions LC_∞ and LC_t of the non-associative Lambek calculus with a labeling discipline and show that these derivation systems are sound and complete for respectively the classes of inftree frames and tree frames. For LC_∞ we also give a cut elimination result.

3.1 LC_∞: a complete labeled calculus for inftree frames

The basic idea of a labeled categorial calculus is that the *structure* of the 'database' of assumptions $A_1, \ldots, A_n$ in a consequence relation

$$A_1, \ldots, A_n \longrightarrow B$$

is made *explicit* by labeling the types:

$$x_1 : A_1, \ldots, x_n A_n \longrightarrow y : B.$$

The basic idea behind our labeling algebra is that somehow labels will refer to trees in the model — this is our instantiation of Gabbay's slogan 'bringing semantics into the syntax'. So let us start with defining the label algebra:

Definition 6 *Assume that we have been given a set M of elements that we will call* **markers**. *Let S be the set of strings over the alphabet $\{0,1\}$ (inductively defined: S is the smallest set containing the empty string λ which contains the strings $s0$ and $s1$ whenever it contains the string s). Elements of the set $M \times S$ are called* **atomic labels**; *the atomic label (a,s) is denoted as a_s, a_λ as a. A* **label over** *M is either an atomic label over M or it has the form (xy) where x and y are labels over M. The set of labels over M is denoted as $\mathtt{Lab}(M)$.*
As abbreviations[2] *we will use the functions l and r over $\mathtt{Lab}(M)$ defined as follows:*

$$\begin{array}{ll} l(a_s) = a_{s0} & l(xy) = x \\ r(a_s) = a_{s1} & r(xy) = y. \end{array}$$

Intuitively, the labels of the form a_s denote the *descendants* of the tree denoted by a; labels of the form (xy) will denote the tree arising by *adjoining* the trees referred to by x and by y. Now we can give a definition of our labeled language:

Definition 7 *Let Pr be a set of primitive types, and M a set of markers. Elements of the Cartesian product $\mathtt{Lab}(M) \times Tp(Pr)$ are called* **formulas (in M and Pr)** *and denoted as $x : A$ where $x \in \mathtt{Lab}(M)$ and $A \in Tp(Pr)$. A* **sequent (in M and Pr)** *is a pair $X \longrightarrow \phi$ where ϕ is a formula and X a finite set of formulas (in M and Pr).*

Turning to the calculus, one of the first questions that we have to answer is whether we want sequents of the form

$$x : A \longrightarrow y : A$$

to be theorems if x and y refer to the same trees in the model. Note that we have to be careful here: what about the theoremhood of $a : p \longrightarrow a_0 a_1 : p$? Although we have not introduced a semantics for labeled sequents yet, it will be clear that the answer to these questions depends on whether we want the tree referred to by a to have daughters or not. As we have an inftree semantics in mind for LC_∞, we will accept such sequents as theorems. For the precise formulation of the rule that takes care of these 'label shifts', we need some terminology:

Definition 8 *Define the relation $\rightarrow_l$ on $\mathtt{Lab}(M)$ by: $a_{s0}a_{s1} \rightarrow_l a_s$. Let $\equiv$ be the congruence relation generated by $\rightarrow_l$, i.e. $\equiv$ is the smallest equivalence relation R containing $\rightarrow_l$ such that $((xz),(yz)) \in R$ and $((zx),(zy)) \in R$ whenever $(x,y) \in R$. We denote the congruence class of x by $\bar{x}$.*

[2]Note that l and r are *not* part of the label algebra; their introduction has the sole purpose of providing a uniform way of referring to the left resp. right daughter of a label x (whether x is atomic or not), thus enabling a concise formulation of for instance the left rule for $\circ$.

Note that it is easy to show that $\equiv$ is decidable.

As for the structural rules of the system, note that our databases are *sets*; therefore, the rules of associativity, permutation and contraction are implicit. The only rule that we need to add explicitly is Weakening (Monotonicity). The operational rules will be discussed after the definition.

Definition 9 *LC_∞ is defined as follows. Its* **logical axioms** *are sequents of the form*

$$x : A \longrightarrow x : A.$$

Its **logical rule** *is the $[Cut]$-rule given by*

$$\frac{X \longrightarrow x : A \quad Y, x : A \longrightarrow \phi}{X, Y \longrightarrow \phi}\ [Cut]$$

Its **label rule** *has the following form:*

$$\frac{X \longrightarrow \phi}{(X \longrightarrow \phi)[x \leftarrow x']}\ [\equiv]\dagger$$

where $(X \longrightarrow \phi)[x \leftarrow x']$ is the sequent $X \longrightarrow \phi$ with one occurrence of x replaced by x'. The rule $[\equiv]$ is only licensed if the side condition ($\dagger$) is met that $x \equiv x'$.

For every connective LC_∞ has two **operational rules***:*

$$\frac{X, l(x) : A, r(x) : B \longrightarrow \phi}{X, x : A \circ B \longrightarrow \phi}\ [\circ L] \qquad \frac{X \longrightarrow l(x) : A \quad Y \longrightarrow r(x) : B}{X, Y \longrightarrow x : A \circ B}\ [\circ R]$$

$$\frac{X, yx : B \longrightarrow \phi \quad Y \longrightarrow y : A}{X, x : A\backslash B, Y \longrightarrow \phi}\ [\backslash L] \qquad \frac{X, a : A \longrightarrow ax : B}{X \longrightarrow x : A\backslash B}\ [\backslash R]*$$

$$\frac{X, xy : B \longrightarrow \phi \quad Y \longrightarrow y : A}{X, x : B/A, Y \longrightarrow \phi}\ [/L] \qquad \frac{X, a : A \longrightarrow xa : B}{X \longrightarrow x : B/A}\ [/R]*$$

In the rules marked with $*$, there is a side condition on the rule stating that the marker a should not occur in x or X.

Finally, LC_∞ has the **structural rule** *of Weakening:*

$$\frac{X \longrightarrow \phi}{X, Y \longrightarrow \phi}\ [W]$$

Most of these rules seem to be pretty obvious[3]. For instance, the side condition in $[\backslash R]$ and $[/R]$ (ensuring a totally hypothetical introduction of $a : A$)

[3]Note that it is a fairly easy exercise to turn the calculus into a classical one by adding boolean type constructors and replacing the intuitionistic one-formula succedents with finite sets of formulas. It seems that the three major results (soundness, cut elimination and completeness) that we are about to prove for LC_∞, will still hold for this extended calculus.

is quite familiar. However, there are some subtleties in the rule $[\circ L]$, as will be discussed in the proof of soundness, and in the right rules for / and \, as we will see in the next subsection. Let us first define what it means for an NL-sequent (i.e. without labels) to be derivable in LC_∞:

Definition 10 *Let X be a term of NL (i.e. a tree over types); the* **formula representation of** X*, notation: $X^\bullet$, is given by the usual inductive definition: $A^\bullet = A$, $(X,Y)^\bullet = (X^\bullet) \circ (Y^\bullet)$.*

Now let $X \longrightarrow A$ be an NL-sequent. We say that $X \longrightarrow A$ is LC_∞-derivable, notation: $LC_\infty \vdash X \longrightarrow A$, if there is (for an arbitrary marker a) an LC_∞-derivation for the LC_∞-sequent $a : X^\bullet \longrightarrow a : A$.

As an example, we show how the sequent Γ_1 (discussed in the previous section) can be derived:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{a_0 : r \longrightarrow a_0 : r}{a_1 : q, a_0 : r \longrightarrow a_0 : r}\,[W]}{a_1 a_0 : q \circ r \longrightarrow a_0 : r}\,[\circ L] \qquad a_1 : p \longrightarrow a_1 : p}{a_0 : p\backslash(q \circ r), a_1 : p \longrightarrow a_0 : r, a_1 : p}\,[\backslash L] \qquad a_1 : p \longrightarrow a_1 \longrightarrow a_1}{a_0 : p\backslash(q \circ r) \longrightarrow a : r \circ p}\,[\circ R]}{a : (p\backslash(q \circ r)) \circ p \longrightarrow a : r \circ p}\,[\circ L]
$$

Now we turn to the semantics of LC_∞:

Theorem 4 *LC_∞ is sound and complete with respect to inftree semantics, i.e. for any NL-sequent*

$$LC_\infty \vdash X \longrightarrow A \iff \mathsf{T}_\infty \models X \longrightarrow A.$$

Proof.
Let us first consider soundness. Here we arrive at the subtlety involved in the left rule for product. The point is that it allows us to define a sound interpretation for arbitrary *labeled* sequents. Let $\mathfrak{M} = (\mathfrak{G}, V)$ be an inftree model. An *assignment* to $\mathfrak{G}$ is a map $f : \mathtt{Lab}(M) \mapsto G$ satisfying $f(x) = f(lx) \cdot f(rx)$. We leave it to the reader to verify that this implies that for any assignment f, $x \equiv y$ implies $fx = fy$.

Now we say that a labeled sequent $X \longrightarrow y : b$ *holds in* $\mathfrak{M}$ *under* f, notation: $\mathfrak{M}, f \models X \longrightarrow y : B$, if

$$(\forall x : A \in X \ \ fx \Vdash A) \ \Rightarrow \ fy \Vdash B.$$

Clearly then, for an NL-sequent $X \longrightarrow A$ we have

$$\mathfrak{M} \models X \longrightarrow A \qquad \text{iff} \qquad \mathfrak{M}, f \models a : X^\bullet \longrightarrow a : A \text{ for all assignments } f.$$

so to prove soundness it suffices to show that $LC_\infty \vdash X \longrightarrow y : B$ implies that for any inftree model $\mathfrak{M}$ and any assignment f, we have $\mathfrak{M}, f \models X \longrightarrow y : B$.

We do this by a straightforward induction on LC_∞-proofs. As an example, we treat the induction step for $[\circ L]$.

Assume as an inductive hypothesis, that $X, l(x) : A_0, r(x) : A_1 \longrightarrow y : B$ holds at every inftree model, and let f be an assignment to an inftree model $\mathfrak{M}$ such that for all $z : C$ in X, $fz \Vdash C$, and $fx \Vdash A_0 \circ A_1$. The latter fact implies that fx has daughters u_0 and u_1 such that $u_i \Vdash A_i$. Now the crucial point is that $fl(x) \cdot fr(x) = fx$, so by Unique Splittability we find $f(lx) = u_0$ and $f(rx) = u_1$. But then the induction hypothesis gives $fy \Vdash B$.

The completeness direction is relatively easy, after we have introduced some terminology: a *description* is a triple $D = (M, P, N)$, where M is a set of markers, and P and N are sets of formulas. Elements of P resp. N are called *positive* resp. *negative requirements*. A description is called *consistent* if for no finite $\Pi \subseteq P$ and $\phi \in N$ we have a derivation $\vdash \Pi \longrightarrow \phi$, *complete* if $P \cup N$ is the set of all formulas (in a given set M of markers and a given set Pr of primitive types). A formula $x : C$ is a $\circ$-*defect* of a description D if C is of the form $A \circ B$ and $x : C$ is in P, but we do not find both $l(X) : A$ and $r(x) : B$ in P. A formula $x : C$ is a $/$-*defect* of a description D if C is of the form A/B and $x : C \in N$, but we do not have a $y \in \mathtt{Lab}(M)$ with both $y : A \in P$ and $(yx) : B \in N$; $\backslash$-defects are defined analogously. A description is called *saturated* if it does not have any defects, *perfect* if it is consistent, complete and saturated.

Let D, D' be two descriptions; D' is an *extension* of D, notation: $D \subseteq D'$, if $M \subseteq M'$, $P \subseteq P'$ and $N \subseteq N'$.

Perfect descriptions give rise to inftree models in a natural way: the trees of the model will be the equivalence classes of $\mathtt{Lab}(M)$ under $\equiv$; note that if $x \equiv y$, the rule $[\equiv]$ ensures that $x : A \in P$ iff $y : A \in P$ and likewise for N. Therefore, the following definition is correct: for a perfect description D, the groupoid model *generated by* D, notation: $\mathfrak{M}^D$, is given as $(\mathfrak{G}, V)$ where $\mathfrak{G}$ is the quotient algebra of the labeling algebra over $\equiv$, and V is given by

$$V(p) = \{\bar{x} \in \mathtt{Lab}(M) \mid x : p \in P\}.$$

Now it is easy to prove by induction on the complexity of types, that for any perfect description D and any formula $x : C$, we have

$$\mathfrak{G}^D, \bar{x} \models C \iff x : C \in \Pi. \tag{1}$$

After these preliminaries, we can start to prove the completeness result. We will show that any sequent which is not derivable in LC_∞ can be falsisied in an inftree model. Let $X \longrightarrow \phi$ be such a sequent. By definition then, $D_0 = (\{a\}, \{a : X^\bullet\}, \{a : \phi\})$ is a consistent description. Suppose that we can extend D_0 to a perfect extension D, then it is easy to show by (1) that $\mathfrak{M}^D \not\models X \longrightarrow A$, as $\overline{a} \in V(X) - V(A)$. It is also immediate that $\mathfrak{M}^D$ is an inftree model.

So the only thing left is to prove the following crucial extension lemma:

$$\text{any consistent description can be extended to a perfect description.} \tag{2}$$

To prove (2), one shows by a straightforward procedure that

1. If D is a consistent description with a defect δ, then D has a consistent extension D' of which δ is not a defect.
2. If D is a consistent description, and $x \in \mathtt{Lab}(M)$, then at least one of $(M, P, N \cup \{x : A\})$ or $(M, P \cup \{x : A\}, N)$ is consistent.

Then by a standard step-by-step method, one can extend any consistent description to a perfect one. This proves (2) and thus the theorem. □

Finally, we show that the cut rule is not really needed in derivations of LC_∞:

Theorem 5 *Let $X \longrightarrow \phi$ be a theorem of LC_∞. Then there is a cut-free derivation of $X \longrightarrow \phi$.*

Proof.
We will need the usual notions like *derivations* or *proof trees*, the *depth* of a proof, the *main formula* in the application of a rule, and the *cut formula* in the application of the $[Cut]$-rule. The statement that $\mathcal{D}$ is a derivation of the sequent $X \longrightarrow \phi$ is denoted by: $X \stackrel{\mathcal{D}}{\longrightarrow} \phi$.

Now as usual, the essential idea in the proof of the theorem is to concentrate first on derivations in which the $[Cut]$-rule is applied only once. To be more precise, we will prove the following claim:

$$\begin{array}{l} \text{If } X \stackrel{\mathcal{D}}{\longrightarrow} \phi, \text{ where } \mathcal{D} \text{ has only one cut,} \\ \text{then there is a cut-free derivation } \mathcal{D}' \text{ with } X \stackrel{\mathcal{D}'}{\longrightarrow} \phi. \end{array} \quad (3)$$

After establishing this claim, we can prove the theorem by an easy induction on the number of cuts in the proof of $X \longrightarrow \phi$.

So let us set out to prove (3). Note that it is sufficient to confine ourselves to treating derivations $\mathcal{D}$ which *end* in an application of $[Cut]$. Assume that the daughters of $\mathcal{D}$ are $\mathcal{D}_0$ and $\mathcal{D}_1$, and that $x : A$ is the cut formula, i.e. $\mathcal{D}$ has the following form:

$$\frac{\dfrac{\mathcal{D}_0}{X \longrightarrow x : A} \quad \dfrac{\mathcal{D}_1}{Y, x : A \longrightarrow y : B}}{X, Y \longrightarrow y : B}\,[Cut]$$

Then the *degree* of the cut is defined as the pair consisting of the number of connectives occurring in (the type of) the cut formula and the sum of the depths of $\mathcal{D}_0$ and $\mathcal{D}_1$; assume that we impose a lexicographical ordering on cut-degrees. Now (3) is proved by induction on the cut-degree of $\mathcal{D}$. For the inductive step, we make the following case distinction:

I First, assume that the cut-formula is main in both $\mathcal{D}_0$ and $\mathcal{D}_1$. We distinguish cases as to whether a connective was introduced in the main formula, or a new label:

$\boxed{[\circ]}$ In this case the derivation looks like

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_{00}}{X_0 \longrightarrow l(x) : A_0} \quad \dfrac{\mathcal{D}_{01}}{X_1 \longrightarrow r(x) : A_1}}{X_0, X_1 \longrightarrow x : A_0 \circ A_1}[\circ R] \quad \dfrac{\dfrac{\mathcal{D}_1}{Y, l(x) : A_0, r(x) : A_1 \longrightarrow y : B}}{Y, x : A_0 \circ A_1 \longrightarrow y : B}[\circ L]}{X_0, X_1, Y \longrightarrow y : B}[Cut]$$

and can be replaced by $\mathcal{D}''$ of the form

$$\dfrac{\dfrac{\mathcal{D}_{00}}{X_0 \longrightarrow l(x) : A_0} \quad \dfrac{\dfrac{\mathcal{D}_{01}}{X_1 \longrightarrow r(x) : A_1} \quad \dfrac{\mathcal{D}_1}{Y, l(x) : A_0, r(x) : A_1 \longrightarrow y : B}}{Y, l(x) : A_0, X_1 \longrightarrow y : B}[Cut]}{X_0, X_1, Y \longrightarrow y : B}[Cut]$$

Note that both cut formulas in $\mathcal{D}''$ have a smaller complexity than the original one, so by the induction hypothesis the two cuts can be removed (one by one).

$\boxed{[/]}$ Here $\mathcal{D}$ has the following form:

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_0}{X, a : A_1 \longrightarrow xa : A_0}}{X \longrightarrow x : A_0/A_1}[/R] \quad \dfrac{\dfrac{\mathcal{D}_{10}}{Y_0, xy : A_1 \longrightarrow y : B} \quad \dfrac{\mathcal{D}_{11}}{Y_1 \longrightarrow y : A_1}}{Y_0, x : A_0/A_1, Y_1 \longrightarrow y : B}[/L]}{X, Y_0, Y_1 \longrightarrow y : B}[Cut]$$

We leave it to the reader to verify that there is a cut free derivation $\mathcal{D}_0'$ of $X, y : A_1 \longrightarrow xy : A_0$. (Here one needs the side condition on $[/R]$ that a does not occur in X.)

So, if we replace $\mathcal{D}$ by

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_{11}}{Y_1 \longrightarrow y : A_1} \quad \dfrac{\mathcal{D}_0'}{X, y : A_1 \longrightarrow xy : A_0}}{X, Y_1, \longrightarrow xy : A_0}[Cut] \quad \dfrac{\mathcal{D}_{10}}{Y_0, xy : A_1 \longrightarrow y : B}}{X, Y_0, Y_1 \longrightarrow y : B}[Cut]$$

we are dealing with a proof tree to which we can apply the inductive hypothesis (twice, just like in the case above).

$\boxed{[\equiv]}$ Here we may replace

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_0}{X \longrightarrow x' : A}}{X \longrightarrow x : A}\,[\equiv] \qquad \dfrac{\dfrac{\mathcal{D}_1}{Y, x'' : A \longrightarrow y : B}}{Y, x : A \longrightarrow y : B}\,[\equiv]}{Y, X \longrightarrow y : B}\,[Cut]$$

by

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_0}{X \longrightarrow x' : A}}{X \longrightarrow x'' : A}\,[\equiv] \qquad \dfrac{\mathcal{D}_1}{Y, x'' : A \longrightarrow y : B}}{Y, X \longrightarrow y : B}\,[Cut]$$

where the application of $[\equiv]$ is justified by the transitivity of $\equiv$. The proof depth of the cut has decreased, so the induction hypothesis applies.

II Now, assume that the cut formula is side formula in the last step of $\mathcal{D}_0$ or $\mathcal{D}_1$. In this case, we will permute the $[Cut]$-rule upwards; again, the particular action we take will depend on the last applied rule of the subtree in which the cut formula was not main. As most of these cases are standard, we only give a few examples:

- Suppose that the cut formula is side formula in the last step of $\mathcal{D}_0$, where we applied the rule $[W]$. Then the derivation $\mathcal{D}$

  $$\dfrac{\dfrac{\dfrac{\mathcal{D}_0'}{X_0 \longrightarrow x : A}}{X_0, X_1 \longrightarrow x : A}\,[W] \qquad \dfrac{\mathcal{D}_1}{Y, x : A \longrightarrow y : B}}{X_0, X_1, Y \longrightarrow y : B}\,[Cut]$$

  is replaced by

  $$\dfrac{\dfrac{\dfrac{\mathcal{D}_0'}{X_0 \longrightarrow x : A} \qquad \dfrac{\mathcal{D}_1}{Y, x : A \longrightarrow y : B}}{X_0, Y \longrightarrow y : B}\,[Cut]}{X_0, X_1, Y \longrightarrow y : B}\,[W]$$

  This proof has a smaller cut-degree than the original one (as the depth of the left subtree has decreased), and can thus be replaced by a cut-free derivation.

- If the last applied rule of $\mathcal{D}_1$ was $[/R]$, and the cut formula was not the main formula of this rule, $\mathcal{D}$ looks like

  $$\dfrac{\dfrac{\mathcal{D}_0}{X \longrightarrow x : A} \qquad \dfrac{\dfrac{\mathcal{D}_1'}{Y, x : A, a : B_1 \longrightarrow ya : B_0}}{Y, x : A \longrightarrow y : B_0/B_1}\,[/R]}{X, Y \longrightarrow y : B_0/B_1}\,[Cut]$$

Note that we may replace the marker a by a marker b that does not appear in X, Y, x, y, obtaining a derivation $\mathcal{D}_1''$ for $Y, x : A, b : B_1 \longrightarrow yb : B_0$ of the same depth as $\mathcal{D}_1'$. We can then show that the following derivation $\mathcal{D}'$ may replace $\mathcal{D}$:

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_0}{X \longrightarrow x : A} \quad \dfrac{\mathcal{D}_1''}{Y, x : A, b : B_1 \longrightarrow yb : B_0}}{X, Y, b : B_1 \longrightarrow yb : B_0}\,[Cut]}{X, Y \longrightarrow y : B_0/B_1}\,[/R]$$

Again we have found a derivation to which the inductive hypothesis applies. □

It follows from the cut elimination theorem that any theorem $X \longrightarrow \phi$ has a proof using only sub*types* of the types occurring in X and ϕ. Note however that this does *not* imply decidability of the calculus, as the rules $[\equiv]$, $[\backslash L]$ and $[/L]$ presume an *infinite* search space (for the left rules of the slashes, observe that the label y of the premisse does not occur in the conclusion).

3.2 LC_t: a complete labeled calculus for tree frames

In this subsection we will transform the system LC_∞ into a calculus that works for arbitrary tree frames. Obviously, it is crucial to avoid as theorems sequents like

$$a : p \longrightarrow a_0 a_1 : p$$

as their soundness would imply that every tree is branching. Clearly, we have to change the rule $[\equiv]$:

$$\frac{X \longrightarrow \phi}{(X \longrightarrow \phi)[x \leftarrow x']}\,[\equiv]$$

Let us concentrate on the case where we replace the label of the *succedent*, i.e. ϕ is of the form $x : A$. We need to install a side condition permitting the rule only when the atomic labels in x' are 'presupposed by' the ones in X. To formalize this condition, we set

Definition 11 *Let a_s and b_r be two atomic labels; we call a_s* **presupposed by** *b_r if $a = b$ and s is an initial segment of r. For a set X of formulas, the label y is* presupposed by *X, notation: $X \rhd y$, if every atomic label in y is presupposed by some atomic label in one of the labels of X.*

We can now formulate a *right* rule $[\equiv R]$ as

$$\frac{X \longrightarrow x : A}{X \longrightarrow x' : A}\,[\equiv R]\$$$

with the side condition ($) that $X \rhd x'$. We have not been able to formulate an appropriate left label rule, in the sense that the arising calculus allows a cut elimination theorem. Therefore we confine ourselves to this one right rule.

Note however, that this change is not sufficient: here we arrive at one of the subtleties mentioned in the previous subsection. The sequent Γ_3, true only in inftree frames (cf. section 2), would still be derivable, witness the derivation below:

$$\dfrac{\dfrac{\dfrac{b:q \longrightarrow b:q \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{b:q \longrightarrow b:q \quad a_0b:p \longrightarrow a_0b:p}{b:q, a_0:p/q \longrightarrow a_0b:p}\,[/L]}{b:q, a_0:p/q, a_1:q\backslash p \longrightarrow a_0b:p}\,[W]}{b:q, a:A \longrightarrow a_0b:p}\,[\circ L]}{ba:q\circ A \longrightarrow a_0b:p}\,[\circ L]}{}}{a:q\backslash(q\circ A), b:q \longrightarrow a_0b:p}\,[\backslash L]}{a:q\backslash(q\circ A) \longrightarrow a_0:p/q}\,[/R] \qquad \dfrac{\text{likewise}}{a:q\backslash(q\circ A) \longrightarrow a_1:q\backslash p}\,[\backslash R]}{a:q\backslash(q\circ A) \longrightarrow a:(p/q)\circ(q\backslash p)}\,[\circ R]$$

A close inspection of this derivation shows that the problem lies in the right rule for $/$: where from $\{a : q\backslash(q\circ A)\}$ one cannot conclude *semantically* that the tree referred to by a has daughters, this conclusion is justified from the database $\{a : q\backslash(q \circ A), b : q\}$. A solution is to replace $[/R]$ and $[\backslash R]$ by

$$\dfrac{X, a:A \longrightarrow xa:B}{X \longrightarrow x:B/A}\,[/R']\ddagger \qquad \text{resp.} \qquad \dfrac{X, a:A \longrightarrow ax:B}{X \longrightarrow x:A\backslash B}\,[\backslash R']\ddagger$$

where we impose the side condition (‡) that x is presupposed by X.

Definition 12 *The calculus LC_t is defined just like LC_∞, with the rules $[/R]$ and $[\backslash R]$ replaced by $[/R']$ and $\backslash R']$, and $[\equiv]$ by $[\equiv R]$.*

We can now give the desired soundness and completeness results:

Theorem 6 *LC_t is sound and complete with respect to tree frames, i.e. for all NL-sequents*

$$LC_t \vdash X \longrightarrow A \iff \mathsf{T} \models X \longrightarrow A.$$

Proof.
To prove soundness, we again introduce an interpretation for *arbitrary* sequents. Here we have to be more careful in our formulation however.

Let X be a set of formulas. An X-assignment f is a *partial* map from labels to elements of a tree model such that (i) $f(x)$ is defined (notation: $f(x)\downarrow$) for all atomic labels x occurring in X, and (ii) for all labels x: if $f(lx)\downarrow$ and $f(rx)\downarrow$, then $f(x)\downarrow$ and $f(x) = f(lx)f(rx)$. Now we say that a labeled sequent $X \longrightarrow x : A$ *holds* in $\mathfrak{M}$ under f, if f can be extended to an X-assignment

g satisfying $g(x) \Vdash A$ whenever $f(u) \Vdash B$ for every $u : B$ in X. With this adaptation we can follow the strategy of the old proof and show soundness of LC_t for tree frames. We give one crucial example: the right rule for $/$.

Assume that $X, x : A \longrightarrow xa : B$ holds in every tree model (i.e. under every $X, x : A$-assignment), and that f is an X-assignment into a tree model $\mathfrak{M}$ such that $\mathfrak{M}, f(u) \Vdash C$ for all $u : C$ in X. The key observation is that $x \in Dom(f)$, as by our new side condition on $[/R']$, X presupposes x . Now distinguish two cases: if $V(A) = \varnothing$, then B/A is true at *every* tree in $\mathfrak{M}$, so in particular $f(x) \Vdash A/B$. But then indeed $\mathfrak{M}, f \models X \longrightarrow x : B/A$. If on the other hand $V(A) \neq \varnothing$, take an arbitrary element $s \in V(A)$. It is left to the reader to verify that there is an extension g_s of f such that $g_s(u) = f(u)$ if $X \rhd u$, and $g_s(a) = s$ (here we crucially use the fact that a is *fresh*). But then g_s is an $X, a : A$-assignment, so by the inductive hypothesis, $g_s(xa) \Vdash B$. Observing that $g_s(xa) = g_s(x)g_s(a) = f(x)s$, we find $f(x) \Vdash A/B$, as s was arbitrary.

For the completeness direction we copy the proof of the previous section, only indicating the places where changes have to be made. The main adaptation is in the definition of a description; a description will here be a triple (L, P, N) with L an upward closed set of labels, and P and N sets of formulas with labels in L. (A set I of labels is *upward closed* if $y \in I$ whenever there is an x such that $x \in I$ and $x \rhd y$.)

Furthermore, we impose the 'finite-presupposition' constraint that every label in L is presupposed by a finite subset of P. In the definition of a complete description, we now take labels in L into account only.

The universe of the groupoid model generated by a perfect description D will now consist of equivalence classes over L. To show that the definition of the interpretation map V is correct, it suffices to prove that for all $x, y \in L$ such that $x \equiv y$, $x : A \in P$ implies $y : A \in P$. Here we need the new constraint that there is a finite subset P_0 of P such that $P_0 \rhd y$. For, the following derivation ensures that $x : A \in P \;\Rightarrow\; y : A \in P$:

$$\cfrac{\cfrac{x : A \longrightarrow x : A}{P_0, x : A \longrightarrow x : A}\,[W]}{P_0, x : A \longrightarrow y : A}\,[\equiv R]$$

In the remainder of the proof, we only have to take care to drag the finite-presupposition condition with us along the construction of the desired perfect description; this is relatively straightforward. $\Box$

4 Evaluation: labels in categorial grammar

Compared to other formalisms studied in the literature (cf. the references given in the introduction), the labeled categorial calculi presented here are of a very basic nature. We believe that it may be useful to put the problems that we encountered here in a wider context, since these problems will by no means vanish in more involved systems. Putting it bluntly, we have the feeling that the introduction of labels in a categorial logic causes as many problems as it solves[4]. One reason for this seems to be the following. Usually, the *motivation* for converting an ordinary calculus into a labeled one, is a mismatch between syntax and semantics. For instance, in the introduction we mentioned the problem concerning Moortgat's infixation and extraction operators, that allow a clear-cut definition of a semantic interpretation, but for which operational derivation rules cannot be expressed. Now, implementing Gabbay's slogan 'bringing semantics into the syntax', one runs the risk of importing this mismatch too...

Let us try to be a bit more precise. When labeling a categorial sequent calculus, one finds a number of areas where radical changes are brought about:

language The most obvious change is that the new 'declarative unit' (to use the LDS terminology) is no longer the pure type, but a type-with-a-label-attached-to-it. There are two points to be made here. First, let us assume that the labels will refer to elements of the intended interpretation. The problem is that the intuitions rising from this intended interpretation may also be quite confusing. For instance, suppose that the interpretation is some kind of *free* algebra, like the fintree models in our example, or the language models for the associative Lambek Calculus. Now, should the language have different kinds of labels referring to generators (leaves), complex terms (trees with daughters) and arbitrary elements (trees), respectively? Note that an affirmative answer may lead to a very complicated syntax, while a negative answer may cause problems for the soundness proof.

Second, in general the label set will have structure. It is not a trivial matter as to how to represent this structure formally, and it may even become necessary to add more kinds of 'declarative units' to the language; for instance in a (substructural) *modal* calculus one may need *symbols* referring to the accessibility relation.

calculus It is by no means a trivial task to arrive at a perspicuous *formal* definition of a labeled calculus, even if (or perhaps precisely because) one is guided by sound semantical considerations. The main issue of course is how to receive the new bookkeeping forced by the introduction of the labels.

A relatively simple matter is formed by the *structural rules* of the calculus. Where the old set of structural rules is more or less determined by the implicit structuring of the database (antecedent), switching to an explicit representation in general will take one to a different level in, or even outside the substructural hierarchy.

[4]One should not read this statement as a denunciation of labeled categorial grammars. A labeled approach *can* be a solution to problems, witness MOORTGAT [19], or KURTONINA [14]. And besides this, some of the problems introduced by labels are quite interesting, and solving them might lead to a better understanding of the issues involved.

Less clear is the problem how to *adapt* the *existing* axioms and derivation rules to the new formalization — there may be venomenous subtleties here. Obviously, this issue is crucially dependent on the system's logical properties that one wants to establish. In particular, it is a quite non-trivial matter how to find and state proper side conditions for the operational rules. First of all, in order to have a decidable set of rules, one should stay away from formulations involving undecidable problems like the quasi-equational theory of semigroups (the word problem). The main problem however seems to be to avoid undesirable side effects of a too naive implementation of semantical intuitions (cf. our discussion preceding Definition 12).

Finally, one has to make clear whether the calculus needs *label rules*, i.e. rules that only involve a re-labeling of the types (as an example, we mention our $\equiv$-rules). Note how tricky the matter may become here: small changes in the labeling rules may have tremendous effects on the properties of the calculus (like cut-elimination). Note too that the decidability problem pops up again.

logical properties Obviously, the motivation to introduce labels in a calculus stems from the desire to obtain a system with some nice properties. Concerning the logical properties, *soundness* seems to be the minimal constraint for a system. Unfortunately, in some cases, it is quite difficult to give an *interpretation* of a labeled sequent in the intended semantics. (In the case of LC_∞ we were in some sense 'lucky' with our semantics, cf. the soundness proof in Theorem 4.) Note that even the *notion* of soundness may have various interpretations: in our examples LC_∞ and LC_t, one might demand validity either for all *labeled* theorems, or for (indirectly) derivable NL-sequents only.

At the moment, *completeness* seems to be out of reach for most systems, and as for *cut elimination*, the symmetry of the old calculus may be disturbed by the side conditions on the operational rules or the label rules.

Finally, *decidability* is no longer an easy consequence of a cut elimination result: the complications involved with the label management may blow up the size of the search space — one is likely to get mixed up with some non-trivial unification problems.

In short, the logical surroundings of labeled categorial grammars differ in almost all fundamental aspects from the substructural landscape that one has some familiarity with. Although we are not implying that labels form a Trojan horse for categorial grammars, it seems to us that the logical foundations of the area of labeled categorial grammars are not established as yet. As to further consequences of our findings for the *linguistic* side of categorial grammar, let us reconsider Problem 1 of the Introduction. Of course, the fact that *we* were not able to find a positive solution to it does not imply that no such solution exist. Nevertheless, we conjecture that a sound and complete calculus (labeled or not) for the class of fintree frames will be quite involved. This means that, at least for the moment, one has to live with calculi that are *incomplete*, like

LC_t, or work in the categorial language *without the product*, for which the simple non-associative Lambek Calculus is complete. Concerning the example of discontinuous constituency, it is interesting to mention the approach in MORRILL [23], in which the problem of finding a unique inserting point in a string or tree is circumvented by the introduction of a *primitive* wrapping operator.

References

[1] van Benthem, J., *Language in Action*, North-Holland, Amsterdam, 1991.

[2] Buszkowki, W., "Completeness results for Lambek syntactic calculus", *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, **32** (1986) pp. 13–28.

[3] Buszkowski, W., Marciszewski, W., & van Benthem, J., (eds.), *Categorial Grammar*, John Benjamin Publ. Co., Amsterdam/Philadelphia, 1988.

[4] Chau, H.F., "A proof search system for a modal substructural logic based on labeled Deductive Systems", manuscript, Department of Computing, Imperial College, London, 1993.

[5] Dekker, P., & Stokhof, M., *Proceedings of the 8th Amsterdam Colloquium*, Institute of Logic, Language and Computation, University of Amsterdam, Amsterdam, 1992.

[6] Dekker, P., & Stokhof, M., *Proceedings of the 9th Amsterdam Colloquium*, Institute of Logic, Language and Computation, University of Amsterdam, Amsterdam, 1994.

[7] Došen, K. & Schröder-Heister (eds.), *Substructural Logics*, Oxford University Press, Oxford, 1993.

[8] Došen, K., "A brief survey of frames for the Lambek calculus", *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, **38** (1992) 179–187.

[9] Gabbay, D.M., *Labeled Deductive Systems – Part I*, Oxford University Press, to appear.

[10] Goranko, V. & Passy, S., "Using the universal modality: gains and questions", *Journal of Logic and Computation*, **2** (1992) 5–30.

[11] Hepple, M., *Labeled deduction and discontinuous constituency*, manuscript, 41 pp., Institute for Research in Cognitive Science, University of Pennsylvania, 1993.

[12] Jakobson, J. (ed.), *Structure of Language and Its Mathematical Aspects*, Providence, 1961.

[13] Kandulski, M., "The non-associative Lambek calculus", in [3], 141–151.

[14] Kurtonina, N., "The Lambek calculus: relational semantics and the method of labeling", *Studia Logica*, to appear.

[15] Lambek, J., 'The mathematics of sentence structure", *American Mathematical Monthly* **65**, 154–484.

[16] Lambek, J., 'On the calculus of syntactic types", in: [12], pp. 166–178.

[17] Moortgat, M., *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus*, Floris, Dordrecht, 1988.

[18] Moortgat, M., "Generalized quantifiers and discontinuous type constructors", *OTS Working Papers* OTS–WP–CL–92–001, Research Institute for Language and Speech, Utrecht University, 1992, to appear in: [29].

[19] Moortgat, M., "Labeled deductive systems for categorial theorem proving", in: [5], 403–424.

[20] Moortgat, M., & Morrill, G., "Heads and Phrases. Type calculus for dependency and constituent structure", manuscript, OTS, Utrecht, 1992, to appear.

[21] Moortgat, M., & Oehrle, R.T., "Adjacency and Dependency", in: [6], to appear.

[22] Morrill, G. & Solias, T., "Tuples, discontinuity, and gapping in categorial grammar", *Proceedings of the European Association for Computational Linguistics 1993*, 287–296.

[23] Morrill, G., *Type Logical Grammar*, Kluwer Academic Publishers, Dordrecht, 1994.

[24] Oehrle, R.T., "Multidimensional compositional functions as a basis for grammatical analysis", in: [25], 349–390.

[25] Oehrle, R.T., Bach, E., & Wheeler, D., (eds.), *Categorial Grammars and Natural Language Structures*, Reidel, Dordrecht, 1988.

[26] Pentus, M., *Lambek calculus is L-complete*, ILLC-Prepublication Series LP–93–14, University of Amsterdam, 1993.

[27] Roorda, D., "Lambek Calculus and Boolean Connectives: on the road", *OTS Working Papers* OTS–WP–CL–92–004, Research Institute for Language and Speech, Utrecht University, 1992.

[28] Roorda, D., *Reasoning with* $\bullet, /, \backslash, \wedge, \vee$ *in semigroups: sequent systems*, manuscript, 21 pp., Vakgroep Informatica, University of Groningen, 1993.

[29] Sijtsma, W., & van Horck, A., (eds.), *Discontinuous Constituency*, Mouton de Gruyter, Berlin, to appear.

[30] Versmissen, K., *Discontinuous Type Constructors in Categorial Grammar*, MSc Thesis, Department of Computing Science, Utrecht University, 1991.

[31] Zielonka, W., "Interdefinability of Lambekian functors", *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, **38** (1992) 501–507.

[32] Zielonka, W., personal communication.