Grammatical Composition: Modes, Models, Modalities

Logical and linguistic aspects of multimodal categorial grammars

OTS Dissertation Series

For further information about OTS publications, please contact:

Onderzoeksinstituut voor Taal en Spraak (OTS)
Research Institute for Language and Speech
Utrecht University
Trans 10, 3512 JK Utrecht
The Netherlands
phone: +31 30 2536006
fax: +31 30 2536000
e-mail: ots@let.ruu.nl

Grammatical Composition: Modes, Models, Modalities

Logical and linguistic aspects of
multimodal categorial grammars

Grammaticale Compositie: Modi, Modellen, Modaliteiten

Logische en taalkundige aspecten van multimodale
categoriale grammatica's

(Met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit Utrecht,
op gezag van de Rector Magnificus
Prof. Dr. J.A. van Ginkel
ingevolge het besluit van het College van Decanen
in het openbaar te verdedigen
op vrijdag 26 april 1996
des middags te 16.15 uur

door

Jacobus Antonius Gerardus Versmissen

Promotor: Prof. Dr. M.J. Moortgat
Onderzoeksinstituut voor Taal en Spraak
Universiteit Utrecht

CIP gegevens

Published by
LEd, Hooghiemstraplein 97, 3514 AX Utrecht, The Netherlands.
OTS, Trans 10, 3512 JK Utrecht, The Netherlands.

ISBN XX–XXXXXXX-X

Contents

Introduction

This thesis is concerned with multimodal categorial grammars. These are logics of grammatical inference designed on the principle that there are numerous ways in which linguistic resources may be structured and can interact. The text is divided into four parts.

Part I starts in Chapter 1 with an introduction along largely historical lines to the *Lambek calculus*, the grammar logic on which the systems that form the subject matter of the rest of the thesis are all based. Chapter 2 starts with a discussion of a number of systems that have been proposed since the mid 1980s as extensions of the Lambek calculus. The shortcomings of these systems where the semantics is concerned have led researchers to consider *multimodal* systems, which are introduced at the end of part I.

The second part is concerned with algebraic semantics for multimodal systems. It starts in Chapter 3 with a case study: the Lambek calculus to which a single domain modality is added. Semantic considerations there lead me to propose a change in the proof theory. The adapted version of the proof system is generalized in Chapter 4 to a whole range of multimodal logics. A uniform approach to syntax and semantics enables me to prove a general completeness theorem for this class of logics.

Part III is based on systems for which a different semantics is assumed, namely one in terms of *Kripke frames*. The different assumptions about the semantics lead to changes in the proof theory — the rules for modalities in particular look totally different in this setting. Chapter 5 introduces frame semantics, and then moves on to a discussion of the embedding theorems of Kurtonina and Moortgat (1995), alternatives to which are presented. This is followed in Chapter 6 by a discussion of the related issues of derivability invariants and characterizations of the conjoinability relation for multimodal calculi.

In the fourth part, a linguistic illustration of multimodal categorial grammars is presented. The fragment that will be accounted for concerns the so-called *verb-raising* construction in Dutch. In Chapter 7 the relevant data are presented, along with an overview of several accounts of verb-raising, both categorial and non-categorial ones. I then show in Chapter 8 how the ideas underlying one of these accounts, Reape's theory of *word order domains*, can be implemented in a multimodal categorial system. The resulting system is then worked out so that it can account for the basic verb-raising data.

Part I

Multimodal categorial grammars

In this part I give a largely historical introduction to the grammar systems that will be the subject of study in the remainder of the thesis. Chapter 1 starts in Section 1.1 with a discussion of three seminal papers on categorial grammar:

1. Kazimierz Ajdukiewicz, *Die syntaktische Konnexität*, 1935;
2. Yehoshua Bar-Hillel, *A quasi-arithmetical notation for syntactic description*, 1953;
3. Joachim Lambek, *The mathematics of sentence structure*, 1958.

Following this, in Section 1.2, I formally introduce the Lambek calculus, along with a number of relevant definitions and theorems.

Chapter 2 starts with a review of extensions to the Lambek calculus that have been proposed over the last decade or so:

- ▷ different sets of structural rules (Section 2.2);
- ▷ additional product operators (Section 2.3.1);
- ▷ modalities *à la* linear logic (Section 2.3.2).

Some of these proposals are problematic since no satisfactory semantics is available. Multimodal categorial grammars have recently surfaced as a possible way out of some of these problems. They are discussed in Section 2.4, and are also the subject of the rest of the thesis.

Chapter 1

Lambek categorial grammars

1.1 Introduction

1.1.1 Ajdukiewicz

Just like many other branches of contemporary logic, the roots of categorial grammar can be traced back, in this case via Leśniewski and Husserl, to Aristotle (see Casadio (1988)). Rather than doing so, I will follow most other writers by taking Ajdukiewicz (1935) as the founding paper of the field.

At the beginning of his paper, Ajdukiewicz states his main concern as follows:

> "[The problem] of syntactic connexion is of the greatest importance for logic. It is concerned with the specification of the conditions under which a word pattern, constituted of meaningful words, forms an expression which itself has a unified meaning [...]. A word pattern of this kind is *syntactically connected*."

The inspiration for his solution to this problem Ajdukiewicz takes from the theory of *semantic categories* developed by Leśniewski (1929). The concept of semantic category, originally due to Husserl (1913), is based on the property of *interreplaceability*. Basically, this property holds of two expressions if the grammaticality of expressions in which either occurs is not affected upon its replacement by the other. The semantic categories are then the classes of all expressions that are mutually interreplaceable.

Among semantic categories two kinds can be distinguished: *basic categories* and *functor categories*. The distinguishing property of functor categories is that their combinatory powers are most appropriately described by saying that the result of their combination with elements of some specific semantic class is invariably an element of some other specific semantic class. Ajdukiewicz takes as basic categories *propositions* and *entities*. From these, we can define functor

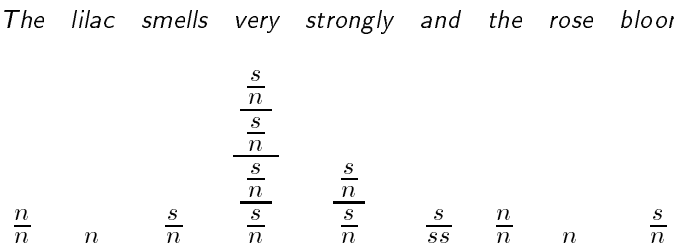
categories. For example, predicates can be described as those expressions which combine with entities to form propositions. Binary connectives such as conjunction and implication, when combined with two propositions, form another proposition.

Ajdukiewicz realised that similar remarks apply to combinations in syntax.[1] The semantic categories of propositions and entities roughly correspond to the syntactic categories of sentences and proper nouns, respectively. An example of a syntactic functor category are the verbs, which can be described as words combining with proper nouns to form sentences.

A further novelty of Ajdukiewicz's approach lies in his introduction of a notation which allows rigourous checking of whether or not an expression is syntactically connected. He defined this notation in the following way:[2]

> "We shall give single words an index according to their semantic category, viz. the simple index 's' to single words belonging to the sentence category; the simple index 'n' to words belonging to the category of names. To single words belonging not to a basic category but to functor categories we shall assign a fractional index, formed of a numerator and a denominator. In the numerator will be the index of the semantic category to which the whole expression composed of the functional sign plus its aguments belongs, while in the denominator appear, one after the other, the indices of the semantic categories of the arguments, which together with the functor combine into a significant whole."

As an example, the words occurring in the sentence *The lilac smells very strongly and the rose blooms* are assigned the following categories by Ajdukiewicz:



The procedure for checking the syntactic connection of an expression consists of two steps. The first step is concerned with dividing an expression into material constituting the functor and material constituting its argument or arguments. The process is then applied recursively to the expressions found this way, until the word level is reached. The second step consists of checking whether this analysis of the expression is consistent with the categories assigned to the words that constitute it. In order for this to be the case, each functor needs to have the right number of arguments, each of the right category.

[1] Cf. the distributional arguments used by American structuralists as a major tool in determining to which categories expressions belong.

[2] Note that Ajdukiewicz didn't distinguish between semantic and syntactic categories. The 'semantic categories' that he refers to are what I have just called syntactic categories.

Formally, the first step is carried out by recursively rearranging the list of categories assigned to an expression in such a way that the functor comes first, followed by all of its arguments in the right order.[3] For the example sentence, this leads to the following sequence of categories:

$$\frac{\frac{\frac{\frac{\frac{s}{n}}{\frac{s}{n}}}{\frac{s}{n}}}{\frac{s}{n}}}{\frac{s}{ss}} \quad \frac{\frac{s}{n}}{\frac{s}{n}} \quad \frac{\frac{s}{n}}{\frac{s}{n}} \quad \frac{s}{n} \quad \frac{n}{n} \quad n \quad \frac{s}{n} \quad \frac{n}{n} \quad n$$

The second step is made up of instances of singling out a functor and its arguments, removing them all and replacing them by the numerator of the functor category. An expression is syntactically connected if and only if this process results in a single type. The sequence of reductions below shows that the example sentence is indeed syntactically connected. I have added dots to indicate which functors and arguments combine in each step.

1. $\frac{s}{ss} \quad .\frac{\frac{\frac{\frac{s}{n}}{\frac{s}{n}}}{\frac{s}{n}}}{\frac{s}{n}} \quad \frac{\frac{s}{n}}{\frac{s}{n}}. \quad \frac{s}{n} \quad \frac{n}{n} \quad n \quad \frac{s}{n} \quad \frac{n}{n} \quad n$

2. $\frac{s}{ss} \quad .\frac{\frac{\frac{s}{n}}{\frac{s}{n}}}{\frac{s}{n}} \quad \frac{s}{n}. \quad \frac{n}{n} \quad n \quad \frac{s}{n} \quad \frac{n}{n} \quad n$

3. $\frac{s}{ss} \quad \frac{s}{n} \quad .\frac{n}{n} \quad n. \quad \frac{s}{n} \quad \frac{n}{n} \quad n$

4. $\frac{s}{ss} \quad .\frac{s}{n} \quad n. \quad \frac{s}{n} \quad \frac{n}{n} \quad n$

5. $\frac{s}{ss} \quad s \quad \frac{s}{n} \quad .\frac{n}{n} \quad n.$

6. $\frac{s}{ss} \quad s \quad .\frac{s}{n} \quad n.$

7. $.\frac{s}{ss} \quad s \quad s.$

8. s

1.1.2 Bar-Hillel

Ajdukiewicz' paper, published in German in a rather obscure Polish journal, remained largely unnoticed for almost two decades. Eventually, his ideas were

[3] This way of arranging funtors and their arguments is of course closely related to the well-known *Polish notation* for logical languages.

taken up and put to more serious linguistic use by Bar-Hillel (1953).

The first step in Ajdukiewicz' procedure for checking the syntactic connection of an expression may at first sight seem slightly odd. After all, how can we be sure of what the eventual funtor-argument structure of the expression will be without knowing what is going to happen in the second step? The reason why Ajdukiewicz set things up in this way was the fact that his main concern was not with *natural*, but rather with *logical* languages. The latter are generally designed with the express purpose of being unambiguous, which means that their functor-argument structure is uniquely determined. This is also exactly the reason why Ajdukiewicz didn't worry about the relative order of the words constituting an expression: in logical languages the structure of an expression determines its appearance.

Bar-Hillel, on the other hand, was interested in natural languages, so he did need to consider word order. This led him to propose an important innovation to Ajdukiewicz' formalism: the indication on the arguments in functor categories of where these arguments were expected to occur: on the right side or on the left side of the functor. He did this by enclosing arguments expected on the right in square brackets, and arguments expected on the left in ordinary brackets. The fact that in a natural language setting the functor-argument structure of an expression isn't predetermined also meant that Bar-Hillel needed to modify Ajdukiewicz' procedure for establishing syntactic connection. He did so by doing away with the first step, hence introducing more indeterminism into the second step, where it was now allowed to combine any functor expression with appropriate arguments occurring on the required side. In this formalism, called **AB**, a sentence such as *John knew that Paul was a poor man* would have the following derivation:

1. $n \quad \frac{s}{(n)[n]} \quad \frac{n}{[s]} \quad n \quad \frac{s}{(n)[n]} \quad \frac{n}{[n]} \quad .\frac{n}{[n]} \quad n.$

2. $n \quad \frac{s}{(n)[n]} \quad \frac{n}{[s]} \quad n \quad \frac{s}{(n)[n]} \quad .\frac{n}{[n]} \quad n.$

3. $n \quad \frac{s}{(n)[n]} \quad \frac{n}{[s]} \quad .n \quad \frac{s}{(n)[n]} \quad n.$

4. $n \quad \frac{s}{(n)[n]} \quad .\frac{n}{[s]} \quad s.$

5. $.n \quad \frac{s}{(n)[n]} \quad n.$

6. s

Bar-Hillel also introduced the distinction between *categories* and *types* into categorial grammar. From now on we will follow him in differentiating between sets of expressions (categories) and the formulas used to refer to these sets (types).

1.1.3 Lambek

Some five years after the publication of Bar-Hillel (1953), Lambek (1958) independently arrived at what can be seen as a further improved version of Bar-Hillel's calculus.

One arguable improvement introduced by Lambek is the different notation he uses. In his formalism, the direction in which a functor is looking for some argument is not indicated on the argument, but instead by using two different notations for fractions: the type y/x denotes a functor looking for an x to its right, whereas the type $x\backslash y$ denotes a functor looking for an x to its left.

In Lambek (1958), the author starts by arguing for a categorial treatment of grammar in a vein similar to Bar-Hillel. However, he then notes that sentences built up from simple transitive verbs raise an important point. Consider for instance a sentence such as *John likes Jane*. If, as usual, we assign the type n to names such as *John* and *Jane*, *semantic* considerations would lead us to assigning *likes* the type $(n\backslash s)/n$. However, from a *syntactic* point of view, the assignment of $n\backslash(s/n)$ would work just as well. In general, it seems as though we would like to have the following equivalence:

$$(x\backslash y)/z \longleftrightarrow x\backslash(y/z)$$

Another example adduced by Lambek concerns the pronouns *he* and *him*. In order to assure that *he* can only occur in subject position, and *him* only in object position, we could try to assign the type $s/(n\backslash s)$ to the former and $(s/n)\backslash s$ to the latter expression. This works fine for sentences such as *He likes Jane* and *Jane likes him*, but fails for *He likes him*. In order to be able to derive the sentencehood of this sentence, Lambek proposes to add the following rules:

$$(x/y)(y/z) \to (x/z), \quad (x\backslash y)(y\backslash z) \to (x\backslash z)$$

Finally, Lambek notes that the types assigned to *he* and *him* can be seen as special instances of the type n, which leads him to introduce the following two rules:

$$x \to y/(x\backslash y), \quad x \to (y/x)\backslash y$$

The main insight of Lambek is that all rules mentioned above can be seen as consequences of a very natural generalization of Bar-Hillel's calculus. This generalized calculus is introduced by Lambek in the form of the axiomatic system below:

$$x \to x$$

$$\frac{x \to y \quad y \to z}{x \to z}$$

$$(xy)z \to x(yz) \qquad x(yz) \to (xy)z$$

$$\frac{x \to y/z}{xy \to z} \qquad \frac{y \to x\backslash z}{xy \to z}$$

$$\frac{xy \to z}{x \to z/y} \qquad \frac{xy \to z}{y \to x\backslash z}$$

The first part of the above system just constitutes a formalisation of the calculus implicit in Bar-Hillel's paper. In particular, the formulas on the fourth line are Lambek's way of writing down Bar-Hillel's reduction rule.

The crucial addition made by Lambek consists of the formulas on the bottom line, which are the exact reverse of Bar-Hillel's reduction schema. What these formulas mark are the transition from a *rule-based* grammar formalism to a *grammar logic*. Although elements of the logical approach are already apparent in the writings of Ajdukiewicz and Bar-Hillel, the basic assumption of these authors was still that they were just *describing* linguistic phenomena, not *explaining* them. This is made very clear by Bar-Hillel when at the beginning of his paper he states that:

> "We are not interested here in developing a method which a linguist might use to ARRIVE at an analysis of a linguistic corpus, but only in a new way in which he could PRESENT the results of his investigations."

The Lambek calculus on the other hand, although by no means alleviating the linguist altogether from the task of arriving at an analysis, helps him or her by providing a logic for reasoning about the behavior of linguistic expressions.

1.2 The Lambek calculus

In this section the Gentzen formulation of the Lambek calculus **L**, which will be used throughout this thesis, is introduced. In order to understand the difference between this system and the axiomatic formulation of the last section, it is important to realize that in Lambek's presentation there is a third operator besides $\backslash$ and $/$. This is the product with respect to which the two slashes can be seen as divisions, i.e. it is the operator representing the concatenation of strings. Lambek only denotes this operator implicitly, i.e. by means of simple juxtaposition. In Gentzen systems it is more convenient to use an explicit notation for it. I will employ the usual symbol '$\bullet$' for this.

The difference between the Gentzen formulation and the axiomatic formulation of the last section lies in the form in which the left hand sides of statements $x \to y$ are cast. In the axiomatic formulation x is always a type, just like y. In the Gentzen formulation, the left hand side is rather a *sequence* of types. Such a sequence of types can be seen as expressing the behaviour of the product in *structural* terms.

I start in Section 1.2.1 with a formal definition of the language used in the Gentzen setting. The actual Gentzen system is given in Section 1.2.2, where I also discuss two important properties of this system: Cut-elimination and decidability. In Section 1.2.3, I introduce free semigroup semantics for **L**, and give proofs of soundness and completeness. Finally the link between the axiomatic and Gentzen systems is made formal in Section 1.2.4.

1.2.1 Language

The Lambek calculus refers to syntactic categories by means of an inductively defined set of *types* $\mathcal{T}$. Occasionally I will refer to types as *formulas* instead. The elements of $\mathcal{T}$ are built up inductively from some finite set $\mathcal{B} = \{\mathsf{p}, \mathsf{q}, \mathsf{r}, \ldots\}$ of *basic types*[4] and the set $\mathbf{C} = \{\bullet, \backslash, /\}$. The operators in this latter set are hence *type constructors*, but in view of the strongly logical orientation of the Lambek calculus they are usually referred to as *connectives*. The construction of $\mathcal{T}$ can be expressed in EBNF notation as follows:

$$\mathcal{T} \quad ::= \quad \mathcal{B} \mid \mathcal{T} \bullet \mathcal{T} \mid \mathcal{T} \backslash \mathcal{T} \mid \mathcal{T} / \mathcal{T}$$

In a type $\mathsf{A} \star \mathsf{B}$, where $\star$ is any of the three connectives, this occurrence of $\star$ is called the *main connective* of the type. Let $\mathbf{S} = \{, , \Delta, \Theta, \ldots\}$ be the set of non-empty sequences of types:

$$\mathbf{S} \quad ::= \quad \mathcal{T} \mid \mathcal{T}, \mathbf{S}$$

Object-level statements are expressed by *sequents*, which can formally be defined as pairs $(, , \mathsf{A})$, where $, \in \mathbf{S}$ and $\mathsf{A} \in \mathcal{T}$. Instead of $(, , \mathsf{A})$ we will write $, \vdash \mathsf{A}$.[5] In such a sequent, $,$ is called the *antecedent* and A the *succedent*.

The notion of *subtype* has the usual recursive definition: every type is a subtype of itself, and the subtypes of A and B are also subtypes of $\mathsf{A} \star \mathsf{B}$ ($\star \in \mathbf{C}$). The *complexity* $c(\mathsf{A})$ of a type A equals the number of occurrences of connectives in it. In other words, $c(\mathsf{p}) = 0$ for basic types p, while $c(\mathsf{B} \star \mathsf{C}) = c(\mathsf{B}) + c(\mathsf{C}) + 1$. The complexity of a sequence of types is the sum of the complexities of the types comprising it.

1.2.2 Gentzen presentation

The Gentzen-style presentation of the Lambek calculus $\mathbf{L}$ is defined as follows:

$$\mathsf{A} \vdash \mathsf{A}$$

$$\frac{, , \mathsf{A}, \mathsf{B}, \Delta \vdash \mathsf{C}}{, , \mathsf{A} \bullet \mathsf{B}, \Delta \vdash \mathsf{C}}\ [\bullet\mathrm{L}] \qquad \frac{, \vdash \mathsf{A} \quad \Delta \vdash \mathsf{B}}{, , \Delta \vdash \mathsf{A} \bullet \mathsf{B}}\ [\bullet\mathrm{R}]$$

$$\frac{, \vdash \mathsf{A} \quad \Delta, \mathsf{B}, \Delta' \vdash \mathsf{C}}{\Delta, \mathsf{B}/\mathsf{A}, , , \Delta' \vdash \mathsf{C}}\ [/\mathrm{L}] \qquad \frac{, , \mathsf{A} \vdash \mathsf{B}}{, \vdash \mathsf{B}/\mathsf{A}}\ [/\mathrm{R}]$$

$$\frac{, \vdash \mathsf{A} \quad \Delta, \mathsf{B}, \Delta' \vdash \mathsf{C}}{\Delta, , , \mathsf{A}\backslash\mathsf{B}, \Delta' \vdash \mathsf{C}}\ [\backslash\mathrm{L}] \qquad \frac{\mathsf{A}, , \vdash \mathsf{B}}{, \vdash \mathsf{A}\backslash\mathsf{B}}\ [\backslash\mathrm{R}]$$

$$\frac{, \vdash \mathsf{A} \quad \Delta, \mathsf{A}, \Delta' \vdash \mathsf{B}}{\Delta, , , \Delta' \vdash \mathsf{B}}\ [\mathrm{Cut}]$$

[4] Other names for basic types that can be found in the literature include *atomic type*, *primitive type* and *propositional variable*.

[5] I.e. the turnstile $\vdash$ replaces the $\rightarrow$ that was used for the axiomatic system.

In any inference step, the sequent or sequents above the line are called the *premises*. The sequent below the line is called the *conclusion* of that inference step. Note that each of the [L] and [R]-rules introduces exactly one occurrence of a connective in the derivation. This connective is called the *active connective* of the inference step. The type A in the [Cut]-rule is called the *Cut formula*.

The most important property of this calculus is the fact that the Cut-rule, which expresses the transitivity of derivability that one would like any well-behaved logic to enjoy, doesn't add to the expressivity of the system. In other words, we have the following theorem:

Theorem 1.1 [Cut] *is an admissible rule of* **L**. *I.e. if a sequent is derivable in* $\mathbf{L} \cup$ [Cut], *then it is derivable in* **L**.

Proof I will give an outline of the original proof of Lambek (1958). This proof is by induction on the complexity of an application of Cut. Consider such an application:

$$\frac{, \vdash \mathsf{A} \quad \Delta, \mathsf{A}, \Delta' \vdash \mathsf{B}}{\Delta, , , \Delta' \vdash \mathsf{B}} \text{ [Cut]}$$

Its *complexity* is defined as

$$c(,) + c(\Delta) + c(\Delta') + c(\mathsf{A}) + c(\mathsf{B})$$

The proof considers 'top-level' occurrences of Cut. These are applications of Cut whose premises have been derived without any use of the Cut-rule. Essentially, the proof consists of showing that any such occurrence of Cut can either be removed, or can be replaced by one or two applications of Cut with a smaller complexity. Since complexities are never negative, this implies that all occurrences of Cut can be removed from a derivation, starting from the top.

Applications of Cut can be divided into three broad (and partially overlapping) types:

1. *trivial* Cuts: at least one of both premises is an axiom sequent;
2. *non-principal* Cuts: any Cuts that aren't principal;
3. *principal* Cuts: the last steps in the derivations of both premises introduce the main connective of the Cut formula A.

The first case is simple: if one premise of an application of Cut is an axiom, then the other premise coincides with the conclusion — hence the term trivial Cut. Obviously, trivial Cuts can be removed from a derivation without affecting its validity

In the case of a non-principal Cut, we know that for at least one of the premises it holds that the last step of its derivation does not introduce the main connective of the Cut formula A. There are two subcases:

1. The last step in the proof of the first premise $, \vdash \mathsf{A}$ does not introduce the main connective of A. Then $, \vdash \mathsf{A}$ must in fact have been inferred by [/L], [\L] or [•L] from one or two sequents, one of which has the form

$, ' \vdash \mathsf{A}$, with $c(, ') < c(,)$. The original application of Cut can therefore be replaced by the following one:

$$\dfrac{, ' \vdash \mathsf{A} \quad \Delta, \mathsf{A}, \Delta' \vdash \mathsf{B}}{\Delta, , ', \Delta' \vdash \mathsf{B}} \text{ [Cut]}$$

It is now straightforward to verify that in all three cases the rule which led from $, ' \vdash \mathsf{A}$ to $, \vdash \mathsf{A}$ will also lead from $\Delta, , ', \Delta' \vdash \mathsf{B}$ to $\Delta, , , \Delta' \vdash \mathsf{B}$.

2. The last step in the proof of the second premise $\Delta, \mathsf{A}, \Delta' \vdash \mathsf{B}$ does not introduce the main connective of A. Then $\Delta, \mathsf{A}, \Delta' \vdash \mathsf{B}$ must have been inferred from one or two sequents, one of which has the form $\Theta, \mathsf{A}, \Theta' \vdash \mathsf{B}'$. Since the inference introduces an occurrence of a connective, $c(\Theta) + c(\Theta') + c(\mathsf{B}') < c(\Delta) + c(\Delta') + c(\mathsf{B})$. Therefore, the following Cut has lower complextiy than the original one:

$$\dfrac{, \vdash \mathsf{A} \quad \Theta, \mathsf{A}, \Theta' \vdash \mathsf{B}'}{\Theta, , , \Theta' \vdash \mathsf{B}'} \text{ [Cut]}$$

Again, it is easy to verify that the same rule which led from $\Theta, \mathsf{A}, \Theta' \vdash \mathsf{B}'$ to $\Delta, \mathsf{A}, \Delta' \vdash \mathsf{B}$ will also lead from $\Theta, , , \Theta' \vdash \mathsf{B}'$ to $\Delta, , , \Delta' \vdash \mathsf{B}$.

Finally, principal Cuts need to be considered. In these cases, the original Cut can be replaced by two Cuts, both of which have lower complexity than the original one. I illustrate this for $/$ (the case of $\backslash$ being analogous) and $\bullet$:

$$\dfrac{\dfrac{, , \mathsf{A} \vdash \mathsf{B}}{, \vdash \mathsf{B}/\mathsf{A}} \text{ [/R]} \quad \dfrac{\Theta \vdash \mathsf{A} \quad \Delta, \mathsf{B}, \Delta' \vdash \mathsf{C}}{\Delta, \mathsf{B}/\mathsf{A}, \Theta, \Delta' \vdash \mathsf{C}} \text{ [/L]}}{\Delta, , , \Theta, \Delta' \vdash \mathsf{C}} \text{ [Cut]}$$

$$\rightsquigarrow$$

$$\dfrac{\Theta \vdash \mathsf{A} \quad \dfrac{, , \mathsf{A} \vdash \mathsf{B} \quad \Delta, \mathsf{B}, \Delta' \vdash \mathsf{C}}{\Delta, , , \mathsf{A}, \Delta' \vdash \mathsf{C}} \text{ [Cut]}}{\Delta, , , \Theta, \Delta' \vdash \mathsf{C}} \text{ [Cut]}$$

$$\dfrac{\dfrac{, ' \vdash \mathsf{A}' \quad , '' \vdash \mathsf{A}''}{, ', , '' \vdash \mathsf{A}' \bullet \mathsf{A}''} \text{ [$\bullet$R]} \quad \dfrac{\Delta, \mathsf{A}', \mathsf{A}'', \Theta \vdash \mathsf{B}}{\Delta, \mathsf{A}' \bullet \mathsf{A}'', \Theta \vdash \mathsf{B}} \text{ [$\bullet$L]}}{\Delta, , ', , '', \Theta \vdash \mathsf{B}} \text{ [Cut]}$$

$$\rightsquigarrow$$

$$\dfrac{, '' \vdash \mathsf{A}'' \quad \dfrac{, ' \vdash \mathsf{A}' \quad \Delta, \mathsf{A}', \mathsf{A}'', \Theta \vdash \mathsf{B}}{\Delta, , ', \mathsf{A}'', \Theta \vdash \mathsf{B}} \text{ [Cut]}}{\Delta, , ', , '', \Theta \vdash \mathsf{B}} \text{ [Cut]}$$

□

The importance of the Cut-elimination theorem lies in the fact that it tends to be a lot easier to prove metalogical properties of Gentzen systems if there's no need to worry about applications of Cut. In the particular case of the Lambek calculus, an almost immediate consequence of Cut-elimination is decidability:

Theorem 1.2 (Decidability) *The question of determining whether or not a sequent has a derivation in the Lambek calculus is decidable.*

Proof From Cut-elimination it follows that we only need to look for Cut-free derivations of the sequent. Note now that every step in such a derivation introduces a connective. In other words, the complexity of the conclusion is strictly larger than the complexity of the premise (or the combined complexity of both premises). Moreover, there are only finitely many possible derivation steps leading to any sequent. These two properties together guarantee that a backward-chaining search algorithm will have a finite search space for any sequent. □

1.2.3 Semantics

Since types are considered to represent sets of strings, *free semigroups* provide a natural semantics for **L** from a linguistic point of view. A free semigroup can be defined as the set of all finite sequences over some set of generators, with sequence concatenation as the product. This associative binary operation will be denoted by the dot $\cdot$ or by simple juxtaposition. The attractiveness of free semigroup semantics lies in the fact that sets of strings with concatenation as the operation are instances of free semigroups. This semantics can already be found between the lines in Lambek (1958).

A *model* in this semantics consists of a semigroup L together with a valuation $[\![.]\!] : \mathcal{B} \to \mathcal{P}(L)$, where $\mathcal{P}(L)$ denotes the set of subsets of L. The valuation is extended to $\mathcal{T}$ by the following recursive definition:

$$\begin{aligned} [\![\mathsf{A} \bullet \mathsf{B}]\!] &= \{a \cdot b \mid a \in [\![\mathsf{A}]\!], b \in [\![\mathsf{B}]\!]\} \\ [\![\mathsf{C}/\mathsf{B}]\!] &= \{a \in L \mid \forall b \in [\![\mathsf{B}]\!] : a \cdot b \in [\![\mathsf{C}]\!]\} \\ [\![\mathsf{A}\backslash\mathsf{C}]\!] &= \{b \in L \mid \forall a \in [\![\mathsf{A}]\!] : a \cdot b \in [\![\mathsf{C}]\!]\} \end{aligned}$$

For subsets $L_1, L_2, \ldots, L_n$ of L, let $L_1 \cdot L_2 \cdots L_n = \{l_1 \cdot l_2 \cdots l_n \mid l_i \in L_i \ (1 \leq i \leq n)\}$. For a sequence $, = \mathsf{A}_1, \mathsf{A}_2, \ldots, \mathsf{A}_n$ I will abbreviate $[\![\mathsf{A}_1]\!] \cdot [\![\mathsf{A}_2]\!] \cdots [\![\mathsf{A}_n]\!]$ as $[\![,]\!]$.

A sequent $, \vdash \mathsf{A}$ is said to be *valid* with respect to a given model, if in that model $[\![,]\!] \subseteq [\![\mathsf{A}]\!]$ holds. A sequent is *generally valid* iff it is valid in all models. The proof system is said to be *sound* with respect to the semantics if all derivable sequents are generally valid. It is *complete* if the converse holds, i.e. if all generally valid sequents can be derived in it. If a proof system is sound, this means that only statements that are actually true in the semantics can be derived. If a proof system is complete, all such statements are derivable.

We now have the following two basic results:

Theorem 1.3 (Soundness for L) *If a sequent is derivable, then it is generally valid.*

Proof This is proved by induction on the length of a derivation.

- ▷ Axiom sequents are obviously generally valid, so in case the derivation consists of a single axiom, the theorem holds.

▷ Now assume the last step in the derivation was a slash right rule, say [/R]:

$$\frac{,\ ,\mathsf{A} \vdash \mathsf{B}}{,\ \vdash \mathsf{B/A}}\ [/\mathrm{R}]$$

Let $t \in [\![,]\!]$. By the induction hypothesis, $t \cdot a \in [\![\mathsf{B}]\!]$ for any $a \in [\![\mathsf{A}]\!]$. That is, $t \in [\![\mathsf{B/A}]\!]$.

▷ The last step in the derivation may also have been a slash left rule, say [/L]:

$$\frac{,\ \vdash \mathsf{A} \quad \Delta,\mathsf{B},\Delta' \vdash \mathsf{C}}{\Delta,\mathsf{B/A},\ ,\Delta' \vdash \mathsf{C}}\ [/\mathrm{L}]$$

Let $t \in [\![,]\!]$, $u \in [\![\Delta]\!]$, $v \in [\![\Delta']\!]$ and $c \in [\![\mathsf{B/A}]\!]$. By the induction hypothesis, $t \in [\![\mathsf{A}]\!]$. Then, by definition of $[\![.]\!]$, $c \cdot t \in [\![\mathsf{B}]\!]$. Now, again by the induction hypothesis, $u \cdot c \cdot t \cdot v \in [\![\mathsf{C}]\!]$, and we're done.

▷ The simple cases where the last step in the dervation was either of the rules for the product $\bullet$ are left to the reader.

□

Theorem 1.4 (Completeness for L) *If a sequent is generally valid, then it is derivable.*

Proof I will only give Buszkowski's (1982) completeness proof for the product-free fragment of **L**. This will be sufficient preparation for the completeness proofs in Part II, since the calculi considered there are product-free as well. The completeness proof for the full Lambek calculus with respect to free semigroup semantics can be found in Pentus (1994b).

Consider the *canonical model* $\mathcal{M}$ given by the free semigroup $\mathcal{T}^+$ of sequences of types, with the product denoted by the comma, and the following interpretation for basic types:

$$[\![\mathsf{p}]\!] = \{,\ \mid ,\ \vdash \mathsf{p}\}$$

The crucial part of the completeness proof is the following *canonical lemma*:

Lemma 1.5 $[\![\mathsf{A}]\!] = \{,\ \mid ,\ \vdash \mathsf{A}\}$ *for all* $\mathsf{A} \in \mathcal{T}$.

Proof We prove this by induction on the complexity of the type A.

▷ If A is a basic type, it is true by the definition of $[\![.]\!]$.

▷ Now let A be a complex type, say $\mathsf{A} = \mathsf{C/B}$.

– First, suppose $, \in [\![\mathsf{A}]\!] = [\![\mathsf{C/B}]\!]$. Then for any $\Delta \in [\![\mathsf{B}]\!]$ we have that $,\ ,\Delta \in [\![\mathsf{C}]\!]$. By the induction hypothesis we deduce from this that $,\ ,\Delta \vdash \mathsf{C}$. In particular, since $\mathsf{B} \in [\![\mathsf{B}]\!]$, we have that $,\ ,\mathsf{B} \vdash \mathsf{C}$, from which it follows by [/R] that $,\ \vdash \mathsf{C/B}$.

– Conversely, suppose that $\Gamma \vdash C/B$, and let $\Delta \in [\![B]\!]$. Then, by the induction hypothesis, $\Delta \vdash B$. We now have the following derivation:

$$\dfrac{\Delta \vdash B \qquad \dfrac{\Gamma \vdash C/B \qquad \dfrac{B \vdash B \quad C \vdash C}{C/B, B \vdash C}\,[/\mathrm{L}]}{\Gamma, B \vdash C}\,[\mathrm{Cut}]}{\Gamma, \Delta \vdash C}\,[\mathrm{Cut}]$$

From this we conclude by the induction hypothesis that $\Gamma, \Delta \in [\![C]\!]$ for all $\Delta \in [\![B]\!]$. That is, $\Gamma \in [\![C/B]\!]$, and we're done.

□

Now suppose the sequent $\Gamma \vdash A$ is not derivable. Then in $\mathcal{M}$ we have, by the lemma we just proved, that $\Gamma \notin [\![A]\!]$. Since $\Gamma \in [\![\Gamma]\!]$, this implies that $[\![\Gamma]\!] \not\subseteq [\![A]\!]$. That is, $\Gamma \vdash A$ is not valid in $\mathcal{M}$, and hence is not generally valid. □

1.2.4 Linking the axiomatic and Gentzen systems

In this section I will show that the axiomatic and Gentzen systems presented earlier are indeed equivalent. The proof that the axiomatic system implies the Gentzen system can already be found in Lambek (1958). The proof that the Gentzen system implies the axiomatic system is due to Došen (1988). As a preliminary to the proof, and to several other proofs in the sequel, I give the following basic lemma. The notation $\Gamma[A]$ denotes a sequence Γ with a distinguished element A.

Lemma 1.6 (Inversion Lemma)

$$\begin{aligned} \Gamma[A, B] \vdash C \quad &\textit{iff} \quad \Gamma[A \bullet B] \vdash C \\ \Gamma, A \vdash B \quad &\textit{iff} \quad \Gamma \vdash B/A \\ A, \Gamma \vdash B \quad &\textit{iff} \quad \Gamma \vdash A\backslash B \end{aligned}$$

Proof Each of these equivalences can be established through an elementary induction on the length of Cut-free derivations. □

With the help of the Inversion Lemma, it is easy to show that all inference steps in the axiomatic system are also valid in the Gentzen set up. The axiom schema is present in both systems, and the transitivity rule in the axiomatic system is an instance of Cut. The associativity of the product follows immediately from the Inversion Lemma and the associativity of the comma. This leaves the following inference rules:

$$\dfrac{A \bullet B \to C}{A \to C/B} \qquad \dfrac{A \bullet B \to C}{B \to A\backslash C}$$

I will prove the first of these two symmetric pairs of equivalences. Suppose first that $A \bullet B \vdash C$. Then by the Inversion Lemma $A, B \vdash C$. From this the [/R]-rules yields $A \vdash C/B$. Conversely, if $A \vdash C/B$ is true, then so is $A, B \vdash C$ by the Inversion Lemma. From this sequent the [•L]-rule yields $A \bullet B \vdash C$.

To show that the Gentzen rules follow from the axiomatic ones, we must first be clear about what exactly this means, since Gentzen sequents are richer in structure than axiomatic ones. Due to the associativity rule, the following equivalences hold:

$$\begin{array}{rcl} A \bullet (B \bullet C) & \textit{iff} & (A \bullet B) \bullet C \\ A\backslash(C/B) & \textit{iff} & (A\backslash C)/B \end{array}$$

This makes it possible to write the above types as simply $A \bullet B \bullet C$ and $A\backslash C/B$, which I will do on occasion. In particular, given a sequence of types , , the type , $^\bullet$ obtained by replacing all occurrences of the comma by a product is well-defined. An immediate consequence of the Inversion Lemma is the following:

$$, \vdash A \quad \textit{iff} \quad ,^\bullet \vdash A$$

This means that in order to establish that the Gentzen rules follow from the axiomatic ones, it is reasonable to show that when , $\vdash A$ is a Gentzen derivable sequent, , $^\bullet \to A$ is derivable in the axiomatic system. In order to facilitate this proof, I will first show the equivalence of the axiomatic presentation below, which was first proposed by Buszkowski (1986), to the one given before:

$$A \to A$$

$$(AB)C \to A(BC) \qquad A(BC) \to (AB)C$$

$$\frac{A \to B}{A \bullet C \to B \bullet C} \qquad \frac{A \to B}{C \bullet A \to C \bullet B}$$

$$\frac{A \to B}{A/C \to B/C} \qquad \frac{A \to B}{C\backslash A \to C\backslash B}$$

$$A \bullet (A\backslash B) \to B \qquad (B/A) \bullet A \to B$$

$$B \to A\backslash(A \bullet B) \qquad B \to (B \bullet A)/A$$

$$\frac{A \to B \quad B \to C}{A \to C}$$

Since the axiom, transitivity and associativity schemata are present in both systems, it is sufficient to show that given these, the two sets of additional rules imply each other.

The easiest direction is to show that the original presentation follows from the one above. Indeed, the equivalence of $A \bullet B \to C$ and $A \to C/B$ in the second system follows from the derivations below:

$$\frac{A \to (A \bullet B)/B \quad \dfrac{A \bullet B \to C}{(A \bullet B)/B \to C/B}}{A \to C/B}$$

$$\dfrac{\dfrac{A \to C/B}{A \bullet B \to (C/B) \bullet B} \quad (C/B) \bullet B \to C}{A \bullet B \to C}$$

The other equivalence is proved in a similar manner.

To show that the new rules follow from the old ones, first observe that this is true for the schemata that were added:

$$\dfrac{A\backslash B \to A\backslash B}{A \bullet (A\backslash B) \to B} \qquad \dfrac{B/A \to B/A}{(B/A) \bullet A \to B}$$

$$\dfrac{A \bullet B \to A \bullet B}{B \to A\backslash(A \bullet B)} \qquad \dfrac{B \bullet A \to B \bullet A}{B \to (B \bullet A)/A}$$

Once this has been established, the validity of the monotonicity inferences follows from the derivations below:

$$\dfrac{\dfrac{A \to B \quad B \to (B \bullet C)/C}{A \to (B \bullet C)/C}}{A \bullet C \to B \bullet C} \qquad \dfrac{\dfrac{A \to B \quad B \to C\backslash(C \bullet B)}{A \to C\backslash(C \bullet B)}}{C \bullet A \to C \bullet A}$$

$$\dfrac{\dfrac{(A/C) \bullet C \to A \quad A \to B}{(A/C) \bullet C \to B}}{A/C \to B/C} \qquad \dfrac{\dfrac{C \bullet (C\backslash A) \to A \quad A \to B}{C \bullet (C\backslash C) \to B}}{C\backslash A \to C\backslash B}$$

The proof that the Gentzen rules follow from the axiomatic rules is by induction on the length of a Cut-free derivation. The cases where the last rule in the derivation was an instance of the axiom schema, [/R], [\R] or [•L] are evident. Suppose the last rule applied was:

[/L] That is, we have the following subderivation:

$$\dfrac{, \vdash A \quad \Delta, B, \Delta' \vdash C}{\Delta, B/A, , , \Delta' \vdash C}$$

From the induction hypothesis it follows that $, ^\bullet \vdash A$ and $\Delta^\bullet \bullet B \bullet (\Delta')^\bullet \vdash C$. The derivation below (in which the product is denoted by juxtaposition in order to save space) shows that $\Delta^\bullet \bullet (B/A) \bullet , ^\bullet \bullet (\Delta')^\bullet \vdash \Delta^\bullet \bullet B \bullet (\Delta')^\bullet$ is derivable. From that sequent and $\Delta^\bullet \bullet B \bullet (\Delta')^\bullet \vdash C$, transitivity then allows us to conclude that $\Delta^\bullet \bullet (B/A) \bullet , ^\bullet \bullet (\Delta')^\bullet \vdash C$.

$$\dfrac{\dfrac{\dfrac{, ^\bullet \vdash A}{, ^\bullet(\Delta')^\bullet \vdash A(\Delta')^\bullet} \qquad \dfrac{(B/A)A \vdash B}{\Delta^\bullet(B/A)A \vdash \Delta^\bullet B}}{\Delta^\bullet(B/A), ^\bullet(\Delta')^\bullet \vdash \Delta^\bullet(B/A)A(\Delta')^\bullet \qquad \Delta^\bullet(B/A)A(\Delta')^\bullet \vdash \Delta^\bullet B(\Delta')^\bullet}}{\Delta^\bullet(B/A), ^\bullet(\Delta')^\bullet \vdash \Delta^\bullet B(\Delta')^\bullet}$$

[\L] Analogous to [/L].

[•R] That is, we have the following subderivation:

$$\frac{, \vdash A \quad \Delta \vdash B}{, , \Delta \vdash A \bullet B}$$

From the induction hypothesis it follows that $, ^{\bullet} \vdash A$ and $\Delta^{\bullet} \vdash B$. The derivation below now shows that $, ^{\bullet} \bullet \Delta^{\bullet} \vdash A \bullet B$, as required:

$$\frac{\dfrac{\Delta^{\bullet} \vdash B}{, ^{\bullet} \bullet \Delta^{\bullet} \vdash , ^{\bullet} \bullet B} \quad \dfrac{, ^{\bullet} \vdash A}{, ^{\bullet} \bullet B \vdash A \bullet B}}{, ^{\bullet} \bullet \Delta^{\bullet} \vdash A \bullet B}$$

Chapter 2

Extensions to the basic Lambek systems

2.1 Introduction

Lambek's 1958 paper didn't spark off a lively research interest in the grammar formalism proposed in it. Quite to the contrary, the article went largely unnoticed for about a quarter century. During that time, interest in the study of natural language along categorial lines was almost exclusively limited to philosophers, who were all concerned with *semantic* issues.[1]

The rediscovery, so to speak, of categorial grammar as a means of *syntactic* description should probably be ascribed to Montague (see Thomason (1974)), who used a modified version of **AB** for the syntax of his grammar formalism. However, Montague's main preoccupation was semantics as well, so that the fact that syntax could be dealt with categorially was of merely tangential interest to him. Still, it is interesting to note that Montague's additions to the descriptive mechanism of the calculus **AB** are largely motivated by *discontinuity* phenomena. Very broadly, these can be described as cases where words or phrases for which there is good reason to assume that they together form a constituent, occur discontinuously in a larger constituent. It is exactly such phenomena that classical categorial grammars such as **AB** and **L** are unable to handle, because of their formulation in terms of *adjacency*: an expression of type B/A is looking for an expression of type A to its *immediate* right, not just *somewhere* to its right.

It is not surprising, then, that when categorial syntax began to be studied in its own right again in the late 1970s, the predominant question was that of how to deal with discontinuity. The initiator of this revival was Emmon

[1] Incidentally, many proposed rules for semantic type inference stemming from this strand of research are in fact derivable in the semantic counterpart of **L**, the Lambek-van Benthem calculus **LP** (see Section 2.2). These rules include function composition (Geach (1971)), type raising (Thomason (1974)), and division (Partee and Rooth (1983)), cf. also Van Benthem (1991).

Bach, see e.g. his (1981), (1983), (1984). Another major contributor was Mark Steedman, see e.g. Ades and Steedman (1982) and Steedman's (1985), (1987), (1990).

The discovery of Lambek's papers was a logical consequence of this renewed interest in categorial syntax. Three names are intimately connected with the early literature on Lambek style cateogrial grammars during the mid and late 1980s. Johan van Benthem concentrated, like Montague before him, on the semantic aspects of such systems, see his (1986), (1991). Wojciech Buszkowski conducted an in-depth investigation of their logical and formal aspects, see e.g. his (1982), (1985), (1986). And finally, Michael Moortgat looked at the use of Lambek grammars for linguistic description, witness his (1988).

Not surprisingly perhaps, research during these first years was focused on the actual Lambek calculus **L** and some of its closest relatives. However, as we already noticed above, **L** has limitations similar to those of **AB** with respect to discontinuity phenomena. Therefore, attempts at linguistic description on the basis of **L** unavoidably ran into problems of descriptive adequacy. In the rest of this chapter I will discuss a number of proposals made to overcome these problems. Leading up to this, in Section 2.2, I discuss the so-called *structural hierarchy* of categorial logics. Section 2.3.1 is concerned with approaches that increase the inventory of binary connectives. The addition of *modal* operators in the spirit of the '!' and '?' of Girard's (1987) linear logic is the subject of Section 2.3.2. Finally, in Section 2.4 it is shown how *multimodal* can overcome some of the problems faced by these proposals.

2.2 A structural hierarchy of logics

The system below is a sequent presentation of a simple, implicational fragment of intuitionistic logic:[2]

$$A \vdash A$$

$$\frac{, \vdash A \quad \Delta, B, \Delta' \vdash C}{\Delta, , , A \Rightarrow B, \Delta' \vdash C}\ [\Rightarrow\text{L}] \qquad \frac{A, , \vdash B}{, \vdash A \Rightarrow B}\ [\Rightarrow\text{R}]$$

$$\frac{, , , ' \vdash C}{, , A, , ' \vdash C}\ [\text{W}] \qquad \frac{, , A, A, , ' \vdash C}{, , A, , ' \vdash C}\ [\text{C}] \qquad \frac{, , A, B, , ' \vdash C}{, , B, A, , ' \vdash C}\ [\text{P}]$$

$$\frac{, \vdash A \quad \Delta, A, \Delta' \vdash B}{\Delta, , , \Delta' \vdash B}\ [\text{Cut}]$$

Apart from the axiom schema and Cut, the rules in the above system can be divided into two groups of rules with quite different characteristics. The

[2] The [Cut]-rule could also be stated in a simpler form, where there is no Δ' (cf. the calculus **LP** given at the end of this section. However, this would cause problems in the absence of the structural rule of permutation.

first group consists of the so-called *logical* rules. These are the rules that give us the conditions for the introduction of the logical connectives on both sides of $\vdash$, such as [$\Rightarrow$L]. The other group contains the *structural* rules. These rules, such as [W], tell us how we can manipulate the set of assumptions without jeopardizing the soundness of sequents.

The advent of *linear logic* (Girard (1987)) drew renewed attention to the structural rules. After all, it was clear that the implicational fragment of intuitionistic linear logic was similar to that of ordinary intuitionistic logic. The major difference was that linear logic had neither of the structural rules of contraction and weakening. The implicational fragment of *relevant logic* (Routley and Meyer (1973)) is in between the ones just mentioned, since it has contraction but not weakening. And finally, the Lambek calculus can be seen as the implicational fragment of linear logic lacking the structural rule of permutation as well.

In fact, a fourth structural rules is hiding behind the structural comma used in the presentation of these logics. This comma arranges elements in sequent antecedents into sequences, which are *associative*: both $((\mathsf{A},\mathsf{B}),\mathsf{C})$ and $(\mathsf{A},(\mathsf{B},\mathsf{C}))$ denote the same sequence, usually written more perspicuously as just $\mathsf{A},\mathsf{B},\mathsf{C}$.

This means that what should rather be considered as the basic implicational substructural logic is the variant of the Lambek calculus from which furthermore the structural rule of associativity has been dropped. This logic, the non-associative Lambek calculus **NL**, was introduced in Lambek (1961). Its Gentzen presentation consists of the following proof system:

$$\mathsf{A} \vdash \mathsf{A}$$

$$\frac{,\,[(\mathsf{A},\mathsf{B})] \vdash \mathsf{C}}{,\,[\mathsf{A} \bullet \mathsf{B}] \vdash \mathsf{C}}\ [\bullet\mathrm{L}] \qquad \frac{,\ \vdash \mathsf{A} \quad \Delta \vdash \mathsf{B}}{(,\,,\Delta) \vdash \mathsf{A} \bullet \mathsf{B}}\ [\bullet\mathrm{R}]$$

$$\frac{,\ \vdash \mathsf{A} \quad \Delta[\mathsf{B}] \vdash \mathsf{C}}{\Delta[(\mathsf{B}/\mathsf{A},\,)] \vdash \mathsf{C}}\ [/\mathrm{L}] \qquad \frac{(,\,,\mathsf{A}) \vdash \mathsf{B}}{,\ \vdash \mathsf{B}/\mathsf{A}}\ [/\mathrm{R}]$$

$$\frac{,\ \vdash \mathsf{A} \quad \Delta[\mathsf{B}] \vdash \mathsf{C}}{\Delta[(,\,,\mathsf{A}\backslash\mathsf{B})] \vdash \mathsf{C}}\ [\backslash\mathrm{L}] \qquad \frac{(\mathsf{A},\,,\,) \vdash \mathsf{B}}{,\ \vdash \mathsf{A}\backslash\mathsf{B}}\ [\backslash\mathrm{R}]$$

$$\frac{,\ \vdash \mathsf{A} \quad \Delta[\mathsf{A}] \vdash \mathsf{B}}{\Delta[,\,] \vdash \mathsf{B}}\ [\mathrm{Cut}]$$

The picture that emerges is that of a hierarchy of implicational logics which all share the same set of logical rules. The differences between these logics are entirely due to differences in the sets of structural rules specified for them. There are of course many such hierarchies, depending on the exact nature of the calculus employed and the structural rules considered. This is not the place for a detailed discussion of such structural hierarchies. The interested reader is instead referred to the relevant discussions in the literature, such as Ono (1988), Moortgat (1988), Van Benthem (1991) and Morrill (1994). A concrete example can be found in Section 5.2.1.

For future reference, let me single out the Lambek-van Benthem calculus **LP**, which can be seen either as the Lambek calculus with the rule of permutation added, or as a simple, multiplicative fragment of linear logic. The product in **LP** is denoted as $\otimes$. Both Lambek slashes are equivalent in **LP**, and the resulting single implication connective is denoted as $\rightarrow$. The structural rules of associativity and commutativity are left implicit in the notation, i.e. sequent antecedents are to be read as *multisets*. The proof system then looks as follows:

$$\mathsf{A} \vdash \mathsf{A}$$

$$\frac{\mathsf{,\ ,A,B} \vdash \mathsf{C}}{\mathsf{,\ ,A} \otimes \mathsf{B} \vdash \mathsf{C}}\ [\otimes\mathrm{L}] \qquad \frac{\mathsf{,} \vdash \mathsf{A} \quad \Delta \vdash \mathsf{B}}{\mathsf{,\ ,}\Delta \vdash \mathsf{A} \otimes \mathsf{B}}\ [\otimes\mathrm{R}]$$

$$\frac{\mathsf{,} \vdash \mathsf{A} \quad \Delta\mathsf{,B,}\Delta' \vdash \mathsf{C}}{\Delta\mathsf{,\ ,\ ,A} \rightarrow \mathsf{B,}\Delta' \vdash \mathsf{C}}\ [\rightarrow\mathrm{L}] \qquad \frac{\mathsf{A,\ ,} \vdash \mathsf{B}}{\mathsf{,} \vdash \mathsf{A} \rightarrow \mathsf{B}}\ [\rightarrow\mathrm{R}]$$

$$\frac{\mathsf{,} \vdash \mathsf{A} \quad \Delta\mathsf{,A} \vdash \mathsf{B}}{\Delta\mathsf{,\ ,} \vdash \mathsf{B}}\ [\mathrm{Cut}]$$

2.3 Adding connectives

2.3.1 Other binary operators

The first proposal for additions to the stock of binary operators in order to overcome the problems **L** has in dealing with discontinuity phenomena are due to Moortgat (1988). The proposal there is to consider the connectives $\uparrow$ and $\downarrow$, whose intuitive behaviour can be described as follows: an expression of type $\mathsf{B} \uparrow \mathsf{A}$ wraps itself around its type A argument in order to form an expression of type B; conversely, an expression of type $\mathsf{B} \downarrow \mathsf{A}$ infixes itself into its A argument in order to form an expression of type B.

The Lambek connectives $/$ and $\backslash$ are deterministic in the sense that there's a fixed point where the argument is to be 'inserted', viz. at the right periphery of the functor for $/$, and at the left periphery of the functor for $\backslash$. With the connectives $\uparrow$ and $\downarrow$ a certain amount of indeterminacy is introduced. Indeed, the insertion point of the argument (in the case of $\uparrow$) or the functor (in the case of $\downarrow$) can in principle be anywhere in the other string. The two most obvious ways of defining $\uparrow$ and $\downarrow$ are those using either existential or universal quantification over the possible insertion points. Differentiating between the two kinds of connectives thus obtained by means of a subscript $\exists$ or $\forall$, we get the following definitions:

$$\begin{aligned}
[\![\mathsf{B} \uparrow_\forall \mathsf{A}]\!] &= \{w \mid \forall w', w'' : (w'w'' = w \ \rightarrow\ \forall a \in [\![\mathsf{A}]\!] : w'aw'' \in [\![\mathsf{B}]\!])\} \\
[\![\mathsf{B} \uparrow_\exists \mathsf{A}]\!] &= \{w \mid \exists w', w'' : (w'w'' = w \ \&\ \forall a \in [\![\mathsf{A}]\!] : w'aw'' \in [\![\mathsf{B}]\!])\} \\
[\![\mathsf{B} \downarrow_\forall \mathsf{A}]\!] &= \{w \mid \forall a \in [\![\mathsf{A}]\!] : \forall a', a'' : (a'a'' = a \ \rightarrow\ a'wa'' \in [\![\mathsf{B}]\!])\} \\
[\![\mathsf{B} \downarrow_\exists \mathsf{A}]\!] &= \{w \mid \forall a \in [\![\mathsf{A}]\!] : \exists a', a'' : (a'a'' = a \ \&\ a'wa'' \in [\![\mathsf{B}]\!])\}
\end{aligned}$$

If one tries to give sequent rules for these connectives, it turns out that this can only be done in a sound way for $[\uparrow_\exists \mathrm{R}]$ and $[\downarrow_\forall \mathrm{L}]$. These rules look as follows:

$$\frac{\Gamma, \Delta, \mathsf{A}, \Delta' \vdash \mathsf{B}}{\Gamma, \Delta, \Delta' \vdash \mathsf{B} \uparrow_\exists \mathsf{A}}\ [\uparrow_\exists \mathrm{R}] \qquad \frac{\Gamma, \Delta, \Delta' \vdash \mathsf{A} \quad \Sigma, \mathsf{B}, \Sigma' \vdash \mathsf{C}}{\Sigma, \Delta, \Gamma, \mathsf{B} \downarrow_\forall \mathsf{A}, \Delta', \Sigma' \vdash \mathsf{C}}\ [\downarrow_\forall \mathrm{L}]$$

In Versmissen (1991), I have shown Cut-elimination and decidability for **L** with this pair of rules added, and soundness with respect to the semantics given above. Since the system only constitutes a *partial* logic for the connectives $\uparrow_\exists$ and $\downarrow_\forall$, a general completeness result obviously doesn't hold.

In Moortgat (1992), it is argued that the connectives $\uparrow$ and $\downarrow$ can be seen as the implications that come with the *substring product* $\odot$. However, the proof rules for $\odot$, $\uparrow$ and $\downarrow$ are expressed with the help of equational constraints on string labels of types, whose exact formal status remains unclear; not surprisingly, no soundness or completeness proofs are given.

Morrill and Solias (1993) attempt to give a more exact formulation of the proof theory and semantics of these connectives.[3] The models they propose to use consist of semigroups on which a *pairing* operation $\langle .,. \rangle$ and its *projections* π_1 and π_2 are defined. The semantics of the three connectives can then be defined in the following way:

$$\begin{aligned} [\![\mathsf{B} \uparrow \mathsf{A}]\!] &= \{s \mid \forall s' \in [\![\mathsf{A}]\!] : \pi_1 s \cdot s' \cdot \pi_2 s \in [\![\mathsf{B}]\!] \\ [\![\mathsf{B} \downarrow \mathsf{A}]\!] &= \{s \mid \forall s' \in [\![\mathsf{A}]\!] : \pi_1 s' \cdot s \cdot \pi_2 s' \in [\![\mathsf{B}]\!] \\ [\![\mathsf{A} \odot \mathsf{B}]\!] &= \{\pi_1 s \cdot s' \cdot \pi_2 s \mid s \in [\![\mathsf{A}]\!], s' \in [\![\mathsf{B}]\!]\} \end{aligned}$$

The accompanying proof system is again phrased in terms of labelled deduction. It is well enough defined that a definition of the interpretation of sequents and a soundness proof seem feasible (although none are given). A completeness proof on the other hand might still be a hard nut to crack.

For a more extensive discussion of the issues touched upon in this section, see e.g. Morrill (1994).

2.3.2 Modal extensions

Quite a different way of tackling the problem emanated from an active group of Ph.D. students working on Lambek categorial grammar at the Centre for Cognitive Science in Edinburgh. Its most important members were Glyn Morrill, Mark Hepple, Guy Barry, Neil Leslie and Martin Emms. The solution they proposed was the addition of so-called *structural modalities*.

In Morrill *et al.* (1990), the authors review the classification of categorial and related logics into a structural hierarchy that I discussed in the last section. They then remark that in linguistic practice, different phenomena may require description in terms of different sets of structural rules, and ask themselves how the hybrid logics needed for this can be obtained.

They take their inspiration from linear logic. In that theory, the full expressivity of classical propositional logic can be regained through the *modalities*

[3] Further illustrations of their use for linguistic description can be found in Hendriks (1995).

'!' and '?'. These modalities bring back the required structural rules in a restricted way. What Morrill *et al.* propose is to make a similar move, starting from the Lambek calculus, for the structural rules of permutation, weakening and contraction,

They give linguistic examples motivating each of these modalities. I will here discuss the one using the permutation modality.

In a sequent format, the essential rules for this modality look as follows:

$$\frac{\Gamma, [A, \Box_P B] \vdash C}{\Gamma, [\Box_P B, A] \vdash C}\ [\Box_P] \qquad \frac{\Gamma, [\Box_P B, A] \vdash C}{\Gamma, [A, \Box_P B] \vdash C}\ [\Box_P]$$

These rules can be construed as two instances of a single rule which is valid in two directions. From now on, I will use the following notation for such rules:

$$\frac{\Gamma, [A, \Box_P B] \vdash C}{\Gamma, [\Box_P B, A] \vdash C}\ [\Box_P]$$

The above set of rules is actually only one of two possible ways of defining structural modalities. The other possibility consists of requiring that *all* types handled by a structural rule are modalized appropriately, rather than just one of them. In case of a permutation modality this can be expressed by the following sequent rule:

$$\frac{\Gamma, [\Box_P B, \Box_P A] \vdash C}{\Gamma, [\Box_P A, \Box_P B] \vdash C}\ [\Box_P]$$

Consider now the type that should be assigned to relative pronouns such as *who*. This type depends on the position of the gap[4] in the relative clause. If there is an object gap, then the appropriate type for *who* is $(N\backslash N)/(S/NP)$. With this type assignment, which basically states that *who* combines with a sentence containing a right-peripheral NP gap to form a noun modifier, we find that a constituent such as *the man who Mary likes* is an NP. Similarly, in the case of a subject gap *who* needs to be assigned the type $(N\backslash N)/(NP\backslash S)$, accounting for NPs such as *the man who Mary met*.

So far, so good. However, we run into problems when we also consider modifiers such as *today*. After all, for an NP such as *the man who Mary met today*, neither of the two type assignments given above works. The reason for this is that the gap doesn't occur on the periphery of the relative clause, and the Lambek notation only allows us to refer to material missing there. The solution proposed by Morrill *et al.* (1990) is to assign *who* the single type $(N\backslash N)/(S/\Box_P NP)$. This modalizes the gapped NP, in effect allowing it to move around in the relative clause. This means that we can require it to occur anywhere in the relative clause — in particular on the right periphery.[5]

Actually, one more thing is needed to make the story presented in the previous paragraph fly. After all, the $\Box_P$NP-gap has a different type from the NP that *met* is looking for. How do we get from one to the other? What this boils down to is the question of what are the appropriate *logical* rules for $\Box_P$.

[4] The usual disclaimer applies.

[5] Or on the left periphery, of course: assigning the type $(N\backslash N)/(\Box_P NP\backslash S)$ to *who* works just as well.

The introduction of modal operators into categorial logics is due to Morrill (1990). The grammar formalism used in that paper uses Montague's IL as the basis for its semantics. Morrill introduces a modality $\Box$ as the syntactic counterpart of the semantic operators $\check{}$ and $\hat{}$. The logical rules for $\Box$ follow from the behaviour of $\check{}$ and $\hat{}$, and look as follows:

$$\frac{\ ,\ ,\mathsf{A},\Delta \vdash \mathsf{B}}{\ ,\ ,\Box\mathsf{A},\Delta \vdash \mathsf{B}}\ [\Box\mathrm{L}] \qquad \frac{\Box\mathsf{A}_1,\ldots,\Box\mathsf{A}_n \vdash \mathsf{B}}{\Box\mathsf{A}_1,\ldots,\Box\mathsf{A}_n \vdash \Box\mathsf{B}}\ [\Box\mathrm{R}]$$

Morrill (1992) proposed to use the same set of logical rules for the structural modalities, and it is this proposal that we will consider in the sequel. Note that these rules allow us to derive $\Box\mathsf{A}$ from A, which was what we needed to make the linguistics application work. The resulting derivation corresponding to the noun phrase *the man who mary met today* is given below:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\textit{Mary met}\ \mathsf{NP}\ \textit{today} \vdash \mathsf{S}}\ [\Box_P\mathrm{L}]}{\textit{Mary met}\ \Box_P\mathsf{NP}\ \textit{today} \vdash \mathsf{S}}\ [\Box_P]}{\textit{Mary met today}\ \Box_P\mathsf{NP} \vdash \mathsf{S}}\ [/\mathrm{R}]}{\textit{Mary met today} \vdash \mathsf{S}/\Box_P\mathsf{NP}}\ [/\mathrm{L}] \quad \dfrac{\mathsf{N} \vdash \mathsf{N} \quad \mathsf{N} \vdash \mathsf{N}}{\mathsf{N},\mathsf{N}\backslash\mathsf{N} \vdash \mathsf{N}}\ [\backslash\mathrm{L}]}{\mathsf{N}\ (\mathsf{N}\backslash\mathsf{N})/(\mathsf{S}/\Box_P\mathsf{NP})\ \textit{Mary met today} \vdash \mathsf{N}}\ [\mathrm{Lex}]}{\textit{man who Mary met today} \vdash \mathsf{N}}\ [/\mathrm{L}] \qquad \mathsf{NP} \vdash \mathsf{NP}}{\mathsf{NP}/\mathsf{N}\ \textit{man who Mary met today} \vdash \mathsf{NP}}\ [\mathrm{Lex}]}{\textit{the man who Mary met today} \vdash \mathsf{NP}}$$

2.4 Multimodal systems

In the last section, I discussed two kinds of proposed extensions to the Lambek calculus. The first of these was the addition of new product connectives in order to describe non-standard behavior of linguistic resources leading to discontinuity phenomena. The second extension concerned the addition of modalities in order to enable controlled access of entities to certain rules.

Although they may have their merits from a linguistic point of view, logically these proposals are quite problematic. This is true in particular for the new product connectives. Often it is hard enough to define an appropriate semantics for them — let alone to give a completeness proof.

This is an undesirable state of affairs. After all, one of the central features of the Lambek calculus is that it is a system for reasoning about grammatical resources. The rules of the calculus express truths about the structuring and interaction of these resources, rather than being mere manipulations of meaningless symbols. In other words, the semantics is just as important as the proof theory. It is this property of the Lambek calculus that has motivated a great deal of contemporary research in categorial grammar, and it would be a great loss if it could not be kept for extended systems.

A solution to this problem is suggested by the intuitions underlying the use of modalities. Recall that these were introduced so that resources could

be granted limited access to certain structural rules, reflecting the fact that not all material behaves the same way all the time. The key insight is now that this latter statement is not only true in the context of modalities, but more generally. Linguistic resources can be structured in different ways, and depending on their structural position can have acess to different structural rules. The problem with the systems presented so far is that there may be more product connectives, but that these are invariably linked to a single operation on linguistic resources, namely concatenation. What is called for, then, is a move to systems built on the recognition that there may be several different modes for structuring linguistic resources, with possibly different properties.

In a Gentzen setting, this can be reflected by replacing the structural comma (which represents concatenation) by a set of structural connectives $\{\circ_i\}_{i\in\mathcal{I}}$, where $\mathcal{I}$ is the set of indexes which denote the different resource management modes. On the type level, there will be the usual set of connectives $\{\bullet_i, \backslash_i, /_i\}$ for each mode $\circ_i$.

Not only can different modes have different structural properties — one mode may be commutative while another one isn't, but there may also be *interaction* between them, cf. the linking of wrapping, tupling and concatenation in the proposal of Morrill and Solias.

The definition of modalities can remain largely unchanged, although it must now be specified for each one which mode (or modes) it interacts with. Since several interpretaions turn out to be available for multimodal systems set up along these lines, the exact proof rules in each case depend on the intended semantics. An example system can be found in Section 4.6.

The availability in principle of several interpretations for multimodal categorial grammars is due to the principled way in which these systems are set up. In particular, there is a separation between *logical rules* expressing the basic relationship between the product and its accompanying slashes on the one hand, and additional *structural* rules and interaction postulates on the other hand. The nature of this relationship is expressed most clearly in the axiomatic presentation of **L** given in Section 1.1.3:

$$\frac{x \to y/z}{xy \to z} \qquad \frac{y \to x\backslash z}{xy \to z}$$

$$\frac{xy \to z}{x \to z/y} \qquad \frac{xy \to z}{y \to x\backslash z}$$

This relationship can alternatively be stated in the form of the following pair of equivalences, commonly known as *residuation* equations:

$$\mathsf{A} \vdash \mathsf{C/B} \quad \textit{iff} \quad \mathsf{A} \bullet \mathsf{B} \vdash \mathsf{C} \quad \textit{iff} \quad \mathsf{B} \vdash \mathsf{A\backslash C}$$

As we saw in Chapter 1, it is easy to define operations on subsets of free semigroups that satisfy these equations. The intuitively appealing free semigroup semantics can to a certain extend be adapted for multimodal systems. Given a set of equations describing the structural behaviour of the different modes of composition, interpretation can take place in the *initial algebra* for these equations. This approach has at least two limitations, however:

1. It only allows us to state structural behavior in terms of *equations*. This means that it can't deal with such rules as weakening and contraction, which essentially involve *inequalities*.

2. It can't capture the behavior of structural modalities if these are to be interpreted in terms of subalgebras exhibiting certain structural properties (which is the path we will follow in the next two chapters), since in general the initial algebra doesn't have any such subalgebras

These considerations will lead me to define the algebraic semantics for multimodal systems in more general terms, namely in terms of arbitrary algebras. After all, it is not at all necessary to start from a free semigroup. Indeed, given any *groupoid* $\langle G, \cdot \rangle$, which is a set endowed with an arbitrary binary operation, the following operations on subsets of G satisfy the residuation equations:

$$\begin{aligned} A \bullet B &= \{a \cdot b \mid a \in A, b \in B\} \\ C/B &= \{a \mid \forall b \in B : a \cdot b \in C\} \\ A\backslash C &= \{b \mid \forall a \in A : a \cdot b \in C\} \end{aligned}$$

The generalized algebraic semantics, which was introduced by Buszkowski (1986), is able to deal with the second problem mentioned above, but not with the first one. However, further generalizations are possible. In Section 4.10, I will discuss the proposal of Venema (1994), who follows Došen (1985) in introducing an *information ordering* $\leq$ on the interpretation algebras. The operations on subsets are then defined as follows:

$$\begin{aligned} A \bullet B &= \{c \mid \exists a \in A, b \in B : a \cdot b \leq c\} \\ C/B &= \{a \mid \forall b \in B : a \cdot b \in C\} \\ A\backslash C &= \{b \mid \forall a \in A : a \cdot b \in C\} \end{aligned}$$

These sets do not satisfy the residuation equations, but this can be remedied by requiring that the interpretation function is upward closed, i.e. for any set S if $x \leq y$ and $x \in S$, then $y \in S$ as well. Under this condition, the semantics is sound and complete. This more complicated semantics has the advantage over the simpler one that it is able to deal with certain inequalities, such as weakening and contraction.

It is even possible to do away with explicit products in the semantics altogether. In this case products are modeled implicitly through ternary relations. The fact that c can be the product of a and b is expressed by adding the tuple $\langle c; a, b \rangle$ to the appropriate relation. The operations on subsets in this case have the following appearance (where R is the modeling relation):

$$\begin{aligned} A \bullet B &= \{c \mid \exists a \in A, b \in B : \mathrm{R}cab\} \\ C/B &= \{a \mid \forall c : \forall b \in B : (\mathrm{R}cab \rightarrow c \in C)\} \\ A\backslash C &= \{b \mid \forall c : \forall a \in A : (\mathrm{R}cab \rightarrow c \in C)\} \end{aligned}$$

This semantics turns out to enable still more general soundness and completeness theorems, as I will discuss in Chapter 5.

For more background on multimodal categorial grammars, see Moortgat and Morrill (1991), Moortgat and Oehrle (1993b), Morrill (1994), Moortgat (to appear).

Part II

Algebraic semantics

In this part I investigate some issues in the algebraic semantics of multimodal categorial grammars as they were described in Section 2.4. Chapter 3 starts with a case study: the Lambek calculus with a single domain modality added. The insights obtained there are then used in Chapter 4 to define a broad class of multimodal systems that come with a semantics with respect to which they are sound and complete.

Chapter 3

A case study

In the next chapter I will conduct an investigation into some of the formal properties of a fairly broad class of multimodal categorial grammars from an algebraic perspective. Leading up to this, in the present chapter I try to give a gentle introduction to the issues involved by investigating a simple example: the Lambek calculus **L** with a domain modality added.

I start by examinining in Section 3.1 whether the proof rules and semantics proposed in the literature are a perfect match. It turns out that such is not the case: the proof system is sound, but not complete. This is remedied in Section 3.2 by a modification of the [□R] rule, leading to a sound and complete system.

3.1 Adding a domain modality to L

As we saw in Section 2.3.2, the system **L**□, introducing a domain modality □ into the Lambek calculus, consists of **L** with the following two rules added:

$$\frac{\Gamma, \Delta, \mathsf{B}, \Delta', \Gamma' \vdash \mathsf{A}}{\Gamma, \Delta, \Box\mathsf{B}, \Delta', \Gamma' \vdash \mathsf{A}}\ [\Box\mathrm{L}] \qquad \frac{\Box\mathsf{B}_1, \ldots, \Box\mathsf{B}_n \vdash \mathsf{A}}{\Box\mathsf{B}_1, \ldots, \Box\mathsf{B}_n \vdash \Box\mathsf{A}}\ [\Box\mathrm{R}]$$

I will show in Section 3.1.1 that this system has the Cut-elimination property. After that, I turn to the semantics of the system, recalling in Section 3.1.2 the interpretation of □ proposed by Hepple (1990). It turns out that with respect to this semantics **L**□ is sound (Section 3.1.3), but not complete (Section 3.1.5).

3.1.1 Cut-elimination

Like **L**, the calculus **L**□ enjoys Cut-elimination, c.f. LIncoln *et al.* (1990). That is, we have the following theorem:

Theorem 3.1 [Cut] *is an admissible rule of* **L**□.

Proof I prove this by extending Lambek's Cut-elimination proof for **L**, which was discussed in Section 1.2.2, to **L**$\Box$. The necessary adaptations to this proof are the following ones:

- ▷ The case of trivial Cuts remains the same.
- ▷ In the case of a non-principal Cut, there are several additional possibilities:
  - The premise $\Gamma, \Delta' \vdash \mathsf{B}$ may have been derived by the $[\Box\mathrm{L}]$-rule:

$$\dfrac{\dfrac{\Gamma, \Delta', \mathsf{C}, \Delta'' \vdash \mathsf{A}}{\Gamma, \Delta', \Box\mathsf{C}, \Delta'' \vdash \mathsf{A}}\,[\Box\mathrm{L}] \qquad \Delta, \mathsf{A}, \Delta''' \vdash \mathsf{B}}{\Delta, \Gamma, \Delta', \Box\mathsf{C}, \Delta'', \Delta''' \vdash \mathsf{B}}\,[\mathrm{Cut}]$$

    This derivation can be replaced by the following one, in which a Cut of smaller complexity occurs:

$$\dfrac{\dfrac{\Gamma, \Delta', \mathsf{C}, \Delta'' \vdash \mathsf{A} \quad \Delta, \mathsf{A}, \Delta''' \vdash \mathsf{B}}{\Delta, \Gamma, \Delta', \mathsf{C}, \Delta'', \Delta''' \vdash \mathsf{B}}\,[\mathrm{Cut}]}{\Delta, \Gamma, \Delta', \Box\mathsf{C}, \Delta'', \Delta''' \vdash \mathsf{B}}\,[\Box\mathrm{L}]$$

  - The premise $\Delta, \mathsf{A}, \Delta' \vdash \mathsf{B}$ may have been derived with a $[\Box\mathrm{L}]$ or $[\Box\mathrm{R}]$ rule that doesn't introduce the main connective of A.
    1. $\Delta, \mathsf{A}, \Delta' \vdash \mathsf{B}$ was derived with the $[\Box\mathrm{L}]$ rule:

$$\dfrac{\Gamma \vdash \mathsf{A} \qquad \dfrac{\Delta, \mathsf{A}, \Delta'', \mathsf{C}, \Delta''' \vdash \mathsf{B}}{\Delta, \mathsf{A}, \Delta'', \Box\mathsf{C}, \Delta''' \vdash \mathsf{B}}\,[\Box\mathrm{L}]}{\Delta, \Gamma, \Delta'', \Box\mathsf{C}, \Delta''' \vdash \mathsf{B}}\,[\mathrm{Cut}]$$

       The following is a derivation of the same sequent with an application of [Cut] of smaller complexity:

$$\dfrac{\dfrac{\Gamma \vdash \mathsf{A} \quad \Delta, \mathsf{A}, \Delta'', \mathsf{C}, \Delta''' \vdash \mathsf{B}}{\Delta, \Gamma, \Delta'', \mathsf{C}, \Delta''' \vdash \mathsf{B}}\,[\mathrm{Cut}]}{\Delta, \Gamma, \Delta'', \Box\mathsf{C}, \Delta''' \vdash \mathsf{B}}\,[\Box\mathrm{L}]$$

    2. $\Delta, \mathsf{A}, \Delta' \vdash \mathsf{B}$ was derived with the $[\Box\mathrm{R}]$ rule:

$$\dfrac{\Gamma \vdash \mathsf{A} \qquad \dfrac{\Delta, \mathsf{A}, \Delta' \vdash \mathsf{B}}{\Delta, \mathsf{A}, \Delta' \vdash \Box\mathsf{B}}\,[\Box\mathrm{R}]}{\Delta, \Gamma, \Delta' \vdash \Box\mathsf{B}}\,[\mathrm{Cut}]$$

       Note that it follows that $\Delta, \mathsf{A}, \Delta'$ must be modal. In particular, A is of the form $\Box\mathsf{C}$. If $\Gamma \vdash \Box\mathsf{C}$ was derived by means of a left rule, we are dealing with the first instance of a non-principal Cut discussed above. Otherwise, $\Gamma \vdash \Box\mathsf{C}$ was derived by an application of $[\Box\mathrm{R}]$. But then Γ is modal, and hence

$\Delta, , , \Delta'$ is so as well, which means that the following derivation of $\Delta, , , \Delta' \vdash \Box B$, with a Cut of smaller complexity, is valid:

$$\dfrac{\dfrac{, \vdash A \quad \Delta, A, \Delta' \vdash B}{\Delta, , , \Delta' \vdash B}\,[\mathrm{Cut}]}{\Delta, , , \Delta' \vdash \Box B}\,[\Box \mathrm{R}]$$

▷ This only leaves the case of the principal Cut, where the last steps in the derivations of the premises both introduce the main connective of $A = \Box C$:

$$\dfrac{\dfrac{, \vdash C}{, \vdash \Box C}\,[\Box \mathrm{R}] \quad \dfrac{\Delta, C, \Delta' \vdash B}{\Delta, \Box C, \Delta' \vdash B}\,[\Box \mathrm{L}]}{\Delta, , , \Delta' \vdash B}\,[\mathrm{Cut}]$$

This derivation can be replaced by the following one, with a Cut of smaller complexity:

$$\dfrac{, \vdash C \quad \Delta, C, \Delta' \vdash B}{\Delta, , , \Delta' \vdash B}\,[\mathrm{Cut}]$$

□

Since Cut-free derivations in $\mathbf{L}\Box$ clearly enjoy the subformula property, the Cut-elimination theorem has decidability as an easy corollary (cf. Section 1.2.2).

3.1.2 Semantics

The first author to propose a semantics for the rules for $\Box$ introduced above was Hepple (1990). He remarks that an expression of type $\Box A$ can be thought of as an expression of type A that is in some sense special, and proposes[1] to express this in the semantics by singling out a subset $L_\Box$ of the domain L which consists of exactly those elements that are considered to be special. The clause for the interpretation of $\Box$ looks as follows:

$$[\![\Box A]\!] = L_\Box \cap [\![A]\!]$$

Under this interpretation, the $[\Box \mathrm{R}]$-rule says that a product of elements of $L_\Box$ is again an element of that set. In other words, we need to require that $L_\Box$ actually be a *subsemigroup* of L in order to ensure the validity of the $[\Box \mathrm{R}]$-rule. In order to check the compatibility of the proposed proof rules and their semantics, I will now try to extend the soundness and completeness proofs of Section 1.2.3 to the new system.

3.1.3 Soundness

For soundness, there are two extra cases to check.

[1] According to Morrill (1994) both this idea and its modification to structural modalities discussed in the next chapter are originally due to Guy Barry.

▷ First suppose that the last rule used in the derivation was [□L]:

$$\frac{\Gamma, \Delta, \mathsf{B}, \Delta', \Gamma' \vdash \mathsf{A}}{\Gamma, \Delta, \Box\mathsf{B}, \Delta', \Gamma' \vdash \mathsf{A}}\ [\Box\mathrm{L}]$$

Let $t \in [\![\Gamma, \Delta]\!]$, $b \in [\![\Box\mathsf{B}]\!]$ and $t' \in [\![\Delta', \Gamma']\!]$. Then, since $[\![\Box\mathsf{B}]\!] = [\![\mathsf{B}]\!] \cap L_\Box \subseteq [\![\mathsf{B}]\!]$, $b \in [\![\mathsf{B}]\!]$. Then, by the induction hypothesis, $t \cdot b \cdot t' \in [\![\mathsf{A}]\!]$.

▷ Next suppose that the last rule used was [□R]:

$$\frac{\Box\mathsf{B}_1, \ldots, \Box\mathsf{B}_n \vdash \mathsf{A}}{\Box\mathsf{B}_1, \ldots, \Box\mathsf{B}_n \vdash \Box\mathsf{A}}\ [\Box\mathrm{R}]$$

Let $b_i \in [\![\Box\mathsf{B}_i]\!]$ $(1 \leq i \leq n)$, and $a = b_1 \cdots b_n$. By the induction hypothesis, $a \in [\![\mathsf{A}]\!]$. Also, since $b_i \in [\![\Box\mathsf{B}_i]\!] = [\![\mathsf{B}_i]\!] \cap L_\Box \subseteq L_\Box$, and $L_\Box$ is a subsemigroup, the product $a = b_1 \cdots b_n$ is in $L_\Box$ as well. Hence $a \in [\![\mathsf{A}]\!] \cap L_\Box = [\![\Box\mathsf{A}]\!]$.

3.1.4 An attempt at proving completeness

The first thing that needs to be done in order to adapt the completeness proof for **L** to **L**□, is to extend the canonical model $\mathcal{M}$ by defining a subsemigroup $\mathcal{M}_\Box$ of $\mathcal{M}$ that will be used for the interpretation of □. If one examines the completeness proof for **L** it becomes clear that $\mathcal{M}_\Box$ will have to contain the set $\{\Gamma, \Delta \mid \exists \mathsf{A} : \Gamma, \Delta \vdash \Box\mathsf{A}\}$. The obvious choice for $\mathcal{M}_\Box$ is then the subsemigroup generated by this set.

In order to extend the completeness proof of Section 1.2.3 to the present case, we have to prove the induction hypothesis for types having □ as their main connective.

▷ First then, let $\Gamma, \Delta \vdash \Box\mathsf{A}$. Then $\Gamma, \Delta \vdash \mathsf{A}$ by the following derivation:

$$\dfrac{\Gamma, \Delta \vdash \Box\mathsf{A} \qquad \dfrac{\mathsf{A} \vdash \mathsf{A}}{\Box\mathsf{A} \vdash \mathsf{A}}\ [\Box\mathrm{L}]}{\Gamma, \Delta \vdash \mathsf{A}}\ [\mathrm{Cut}]$$

The induction hypothesis now allows us to conclude that $\Gamma, \Delta \in [\![\mathsf{A}]\!]$. Also $\Gamma, \Delta \in \mathcal{M}_\Box$, by definition of $\mathcal{M}_\Box$. Hence $\Gamma, \Delta \in [\![\mathsf{A}]\!] \cap \mathcal{M}_\Box = [\![\Box\mathsf{A}]\!]$.

▷ Conversely, let $\Gamma, \Delta \in [\![\Box\mathsf{A}]\!] = [\![\mathsf{A}]\!] \cap \mathcal{M}_\Box$. From $\Gamma, \Delta \in [\![\mathsf{A}]\!]$ and the induction hypothesis, we conclude that $\Gamma, \Delta \vdash \mathsf{A}$. Also, since $\Gamma, \Delta \in \mathcal{M}_\Box$, there are Γ_i, Δ_i, B_i such that $\Gamma, \Delta = \Gamma_1, \Delta_1, \ldots, \Gamma_n, \Delta_n$ and $\Gamma_i, \Delta_i \vdash \Box\mathsf{B}_i$ $(1 \leq i \leq n)$. It is at this point that our attempt fails, since from these data the proof system does not allow us to conclude that $\Gamma, \Delta \vdash \Box\mathsf{A}$.

3.1.5 Incompleteness

Indeed, the proof system is in fact incomplete, as the following example[2] shows.

Example Consider the following two sequents:

1. $\Box B/A, A \vdash C/(A\backslash((\Box B/A)\backslash C))$
2. $\Box B/A, A \vdash \Box(C/(A\backslash((\Box B/A)\backslash C)))$

The first of these is a theorem of $\mathbf{L}\Box$, as is shown by the following derivation:

$$\dfrac{\dfrac{A \vdash A \quad \dfrac{\dfrac{\vdots}{\Box B/A \vdash \Box B/A} \quad C \vdash C}{\Box B/A, (\Box B/A)\backslash C \vdash C}\,[\backslash L]}{\Box B/A, A, A\backslash((\Box B/A)\backslash C) \vdash C}\,[\backslash L]}{\Box B/A, A \vdash C/(A\backslash((\Box B/A)\backslash C))}\,[/R]$$

On the other hand, the second sequent is not derivable, as the following attempt at proving it indicates:

$$\dfrac{\dfrac{A \vdash A \quad \dfrac{\dfrac{\dfrac{\dfrac{\text{fail}}{B \vdash A}}{\Box B \vdash A}\,[\Box L] \quad \dfrac{\text{fail}}{(\Box B/A)\backslash C \vdash C}}{\Box B, (A\backslash((\Box B/A)\backslash C)) \vdash C}\,[\backslash L]}{\Box B \vdash C/(A\backslash((\Box B/A)\backslash C))}\,[/R]}{\Box B \vdash \Box(C/(A\backslash((\Box B/A)\backslash C)))}\,[\Box R]}{\Box B/A, A \vdash \Box(C/(A\backslash((\Box B/A)\backslash C)))}\,[/L]$$

The reader may check that other proof attempts also lead to failure. The underivable second sequent is generally valid. To see this, assume that $t \in [\![\Box B/A]\!]$ and $a \in [\![A]\!]$. Because the first sequent is derivable, and the proof system is sound, we find that $t \cdot a \in [\![C/(A\backslash((\Box B/A)\backslash C))]\!]$. Also, by the definition of $[\![.]\!]$, we know that $t \cdot a \in [\![\Box B]\!] = [\![B]\!] \cap L_\Box \subseteq L_\Box$. But then $t \cdot a \in [\![C/(A\backslash((\Box B/A)\backslash C))]\!] \cap L_\Box = [\![\Box(C/(A\backslash((\Box B/A)\backslash C)))]\!]$, and since $t \in [\![B/A]\!]$ and $a \in [\![A]\!]$ were arbitrary, this proves general validity of the sequent.

3.2 A sound and complete calculus

Although the attempted completeness proof of Section 3.1.4 was bound to fail, it still has its merits. I show in Section 3.2.1 that it suggests a modification of the $[\Box R]$-rule which results in a sound and complete calculus. This calculus also enjoys Cut-elimination (Section 3.2.2), and is decidable (Section 3.2.3). This is not to say that it is an impeccable logic — see the end of Section 4.10 for discussion.

[2] Due to Marco Hollenberg.

3.2.1 Soundness and completeness

Our attempted completeness proof suggests the following generalization of the [□R]-rule:

$$\frac{,_1 \vdash \Box B_1 \quad \ldots \quad ,_n \vdash \Box B_n \quad ,_1,\ldots,,_n \vdash A}{,_1,\ldots,,_n \vdash \Box A}\ [\Box R']$$

In other words, the condition that all types in , have □ as their main connective is relaxed to the requirement that , can be split up in subsequences, from each of which a boxed type is derivable.

Let's call $\mathbf{L}\Box'$ the proof system that arises from adding [□L] and [□R′] to **L**. The proof attempt of Section 3.1.4 becomes an actual completeness proof when it is applied to the calculus $\mathbf{L}\Box'$. In order to prove soundness for $\mathbf{L}\Box'$, we need only to show that the new rule, [□R′], is sound. The straightforward proof of this fact is left to the reader.

Thus we've established the following:

Theorem 3.2 (Soundness and Completeness for $\mathbf{L}\Box'$) *A sequent is generally valid if and only if it is derivable.*

3.2.2 Cut-elimination for $\mathbf{L}\Box'$

$\mathbf{L}\Box'$ also enjoys the Cut-elimination property:

Theorem 3.3 [Cut] *is an admissible rule of* $\mathbf{L}\Box'$.

Proof For this proof, and similar ones in the next chapter, the notion of 'complexity of an application of [Cut]' needs to be refined. I will not give a formal definition here, but just note that basically all rewriting steps in the proof of Cut-elimination replace an application of [Cut] by one or more that are either of lower complexity or are higher up in the derivation. For an exact definition of a well-founded ordering on the basis of this, the reader can for instance turn to Venema (1994).

For the larger part, the present proof exactly mimics the one for $\mathbf{L}\Box$. Only two complications arise.

The first complication arises in the case of a non-principal Cut where the last step in the derivation of the second premise was an instance of the [□R′]-rule. Writing $\Delta = \Delta_1,\ldots,\Delta_k$ and $\Delta' = \Delta'_k,\Delta_{k+1},\ldots,\Delta_n$, this case looks as follows:

$$\dfrac{,\ \vdash A \qquad \dfrac{\Delta_k, A, \Delta'_k \vdash \Box C_k \quad \Delta_i \vdash \Box C_{i\ (i\neq k)} \quad \Delta, A, \Delta' \vdash B'}{\Delta, A, \Delta' \vdash \Box B'}\ [\Box R']}{\Delta,,,\Delta' \vdash \Box B'}\ [\mathrm{Cut}]$$

This can be replaced by the following derivation, containing two cuts that both have a smaller degree than the original one:

$$\dfrac{\dfrac{,\ \vdash A \quad \Delta_k, A, \Delta'_k \vdash \Box C_k}{\Delta_k,,,\Delta'_k \vdash \Box C_k}\ [\mathrm{Cut}] \qquad \Delta_i \vdash \Box C_{i\ (i\neq k)} \qquad \dfrac{,\ \vdash A \quad \Delta, A, \Delta' \vdash B'}{\Delta,,,\Delta' \vdash B}\ [\mathrm{Cut}]}{\Delta,,,\Delta' \vdash \Box B'}\ [\Box R']$$

The second complication is the different shape of the principal Cut on a Cut-formula $\mathsf{A} = \Box\mathsf{C}$, which in this case looks as follows:

$$\dfrac{\dfrac{, _1 \vdash \Box\mathsf{B}_1 \quad \ldots \quad , _n \vdash \Box\mathsf{B}_n \quad , _1, \ldots, , _n \vdash \mathsf{C}}{, _1, \ldots, , _n \vdash \Box\mathsf{C}}\,[\Box\mathrm{R}] \qquad \dfrac{\Delta, \mathsf{C}, \Delta' \vdash \mathsf{B}}{\Delta, \Box\mathsf{C}, \Delta' \vdash \mathsf{B}}\,[\Box\mathrm{L}]}{\Delta, , _1, \ldots, , _n, \Delta' \vdash \mathsf{B}}\,[\mathrm{Cut}]$$

This application of Cut can be replace by the following simpler one:

$$\dfrac{, _1, \ldots, , _n \vdash \mathsf{C} \quad \Delta, \mathsf{C}, \Delta' \vdash \mathsf{B}}{\Delta, , _1, \ldots, , _n, \Delta' \vdash \mathsf{B}}\,[\mathrm{Cut}]$$

□

3.2.3 Subformula property and decidability

For **L** and **L**$\Box$, decidability followed from the subformula property for Cut-free derivations, which was evident. For **L**$\Box'$ things aren't so easy: there is no a priori restriction in the $[\Box\mathrm{R}']$ rule on the types B_i in the premises. Erik Aarts (p.c.) has shown that a kind of subformula property holds for **L**$\Box'$ as well, implying the decidability of the calculus. Below I present a slightly modified version of his argument.

The first observation to make is that Cut-free derivations in **L**$\Box'$ have a normal form in which none of the premises $, _i \vdash \Box\mathsf{B}_i$ of an application of the $[\Box\mathrm{R}']$ rule is the conclusion of another application of that rule. Indeed, suppose we have such a situation, say:

$$\dfrac{\dfrac{, _i \vdash \Box\mathsf{B}_i \ (1 \leq i \leq k) \quad , _1, \ldots, , _k \vdash \mathsf{B}}{, _1, \ldots, , _k \vdash \Box\mathsf{B}}\,[\Box\mathrm{R}'] \qquad , _j \vdash \Box\mathsf{B}_j \ (k+1 \leq j \leq n) \quad , \vdash \mathsf{A}}{, \vdash \Box\mathsf{A}}\,[\Box\mathrm{R}']$$

This can be replaced by the following single application of $[\Box\mathrm{R}']$:

$$\dfrac{, _1 \vdash \Box\mathsf{B}_1 \quad \ldots \quad , _k \vdash \Box\mathsf{B}_k \quad , _{k+1} \vdash \Box\mathsf{B}_{k+1} \quad \ldots \quad , _n \vdash \Box\mathsf{B}_n \quad , \vdash \mathsf{A}}{, \vdash \Box\mathsf{A}}\,[\Box\mathrm{R}']$$

Now consider a premise $, _k \vdash \Box\mathsf{B}_k$ of an application of $[\Box\mathrm{R}']$ in such a normalized Cut-free derivation. The succedent type $\Box\mathsf{B}_k$ must have been introduced by means of an axiom $\Box\mathsf{B}_k \vdash \Box\mathsf{B}_k$, since only such axioms and the $[\Box\mathrm{R}']$ rule can create a succedent type that has $\Box$ as its main connective. In between this axiom and the application of $[\Box\mathrm{R}']$ only left rules have been applied, which do not remove subtypes from the sequent antecedent. Therefore, $\Box\mathsf{B}_k$ must be a subtype of $, _k$. Decidability now follows immediately from this subformula property for normal form proofs.

Chapter 4

Gautam logics

In this chapter the insights from the previous one will be used to obtain soundness and completeness result for a broad class of multimodal categorial grammars. These grammars are allowed to have structural rules, both for their product connectives and for their modalities. The question then becomes how to link the semantics with the proof system. I start in Section 4.1 with an investigation of what this link looks like in some simple cases. These suggest a generalization of which I show in Section 4.2 that it is intimately connected with Gautam's (1957) theorem. This is the reason why the logics that I'm going to define will be termed *Gautam logics*.

The language of these logics is described in Section 4.3. Before giving the proof system, it will be necessary to review some notions from the theory of equational specifications, which I do in Section 4.4. With this out of the way it is finally possible to exactly formulate what parameters determine a Gautam logic (Section 4.5) and give the proof system (Section 4.6), of which I show that it enjoys Cut-elimination in Section 4.7.

The semantics is formulated precisely in Section 4.8, after which I prove soundness and completeness in Section 4.9. Finally, in Section 4.10, I contrast my approach with that of Venema (1994).

4.1 Linking semantics and proof theory

Morrill (1992) proposes to set up the semantics for structural modalities along the same lines as the semantics for domain modalities which I discussed in Section 3.1.2. The idea is that the interpretation of, say, a permutation modality should be a special kind of subsemigroup, namely one that is *commutative*. Similarly, just like **L** has a natural interpretation in terms of semigroups, its commutative cousin **LP** has a natural interpretation in terms of *commutative* semigroups.

Natural though this connection between syntax and semantics may seem, there is nonetheless a hidden assumption behind it. After all, in the semantics

the structural equalities obtain on the object level, whereas in the syntax they hold between types, which denote *sets* of objects.

In the case of permutation this is a valid assumption to make. However, if one moves to a more general setting it may no longer be true. Consider for example the structural rule $x = x + x$. This would translate to the following pair of inference rules:

$$\frac{, [\Delta] \vdash \mathsf{A}}{, [\Delta, \Delta] \vdash \mathsf{A}}$$

Suppose we try to interpret the Lambek calculus with this structural rule added in semigroups whose product satisfies $x = x \cdot x$. It then turns out that the new rule isn't even sound with respect to this semantics. The downward direction is sound, since if $a \in [\![\Delta]\!]$, then also $a \cdot a \in [\![\Delta, \Delta]\!]$ from which soundness follows. In the other direction, however, one obtains a term $c \in [\![\mathsf{A}]\!]$ with a subterm $a \cdot b \in [\![\Delta, \Delta]\!]$. This doesn't in general allow one to conclude that c with $a \cdot b$ replaced by any element of $[\![\Delta]\!]$ is also an element of $[\![\mathsf{A}]\!]$, which would be the normal way of proving soundness.

Since a sequent $, \vdash \mathsf{A}$ is valid iff $[\![,]\!] \subseteq [\![\mathsf{A}]\!]$, it is not a priori absolutely necessary for both antecedents occurring in inference rules like the ones above to have exactly the same interpretation. However, in case the product $\bullet$ is present in the language, all structures correspond to types, and then it is easy to show that such an equality must hold. If, which is usually the case, the calculus with product is a conservative extension of the one without product, this in turn implies that also for the latter system equality is required. The question that needs to be asked, then, is exactly when equations that obtain among objects of the model also hold of sets of such objects. In the next section an answer to this question will be provided.

4.2 Gautam's theorem

As we have seen, given an algebra, the object level multiplication $\cdot$ can be lifted to the set level by means of the following definition:

$$S \cdot T = \{s \cdot t \mid s \in S, t \in T\}$$

This definition is easily generalized to algebras of an arbitrary signature by stipulating for every n-ary operator f that:

$$f(S_1, \ldots, S_n) = \{f(s_1, \ldots, s_n) \mid s_i \in S_i \, (1 \leq i \leq n)\}$$

Let us call an equation over an arbitrary signature *c-valid* if it carries over from the object level to the set level under the above definitions. We then have the following theorem:

Theorem 4.1 (Gautam) *The necessary and sufficient condition that an equation $s = t$ be c-valid is that the individual variables that occur in $s = t$ occur only once both in s and t or that s and t be identical.*

Proof See Gautam (1957). □

4.3 Language

The languages of the logics that will be considered in this chapter are specified by the following parameters:

- Three finite, disjoint index sets $\mathcal{I}$, $\mathcal{J}$ and $\mathcal{K}$;
- A finite set $\mathcal{B}$ of basic types.

Given these, we define the following sets of expressions (cf. Section 1.2.1):

- The set of binary connectives $\mathbf{C} = \{/_i, \backslash_i\}_{i\in\mathcal{I}}$;
- The set of unary connectives $\mathbf{M} = \{\triangle_j\}_{j\in\mathcal{J}} \cup \{\triangledown_k\}_{k\in\mathcal{K}}$;
- The set of types $\mathcal{T}$, being the inductive closure of $\mathcal{B}$ under $\mathbf{C} \cup \mathbf{M}$;
- The set of structural connectives $\mathbf{SC} = \{\circ_i\}_{i\in\mathcal{I}}$;
- The set of structures $\mathbf{S}$, being the inductive closure of $\mathcal{T}$ under $\mathbf{SC}$;
- The set of sequents $\{\, \vdash \mathsf{A} \mid \, \in \mathbf{S}, \mathsf{A} \in \mathcal{T}\}$.

The distinction of two kinds of unary connectives reflects the alternatives mentioned in Section 2.3.2. Modalities $\triangle_j$ are those whose structural rules only apply when all types involved are prefixed with them, while only a single type prefixed with $\triangledown_k$ needs to be involved in order for the accompanying structural rules to be applicable.

4.4 Equational specifications

I will use equational specifications to describe the structural behaviour of connectives and modalities, as well as the algebraic structures in which these are interpreted. To start with, I recall a number of definitions.

A *signature* Σ is a collection of function symbols, each of which has a fixed arity. Let $\mathcal{V}$ be a countably infinite set of variables. The *term algebra* $\mathcal{T}(\Sigma, \mathcal{V})$ is defined as the inductive closure of $\mathcal{V}$ under Σ. An *equational specification* is a pair $(\Sigma, \mathcal{E})$ where Σ is a signature and $\mathcal{E}$ is a set of equations $s = t$ between terms $s, t \in \mathcal{T}(\Sigma, \mathcal{V})$. A Σ-algebra $\mathcal{A}$ is a *model* for a set of equations $\mathcal{E}$ over $\mathcal{T}(\Sigma, \mathcal{V})$, written as $\mathcal{A} \models \mathcal{E}$, if every equation of $\mathcal{E}$ holds in $\mathcal{A}$ (i.e. is valid under arbitrary assignments of elements of the carrier set of $\mathcal{A}$ to the elements of $\mathcal{V}$). A $(\Sigma, \mathcal{E})$-algebra is a Σ-algebra that is a model for $\mathcal{E}$.

Let $\mathcal{E}$ be an equational specification. Then we define $\mathcal{E}_{\triangle_j}$ to be the equational specification obtained from $\mathcal{E}$ by prefixing each variable occurrence with $\triangle_j$. The definition of the equational specification $\mathcal{E}_{\triangledown_k}$ is given below, where $\mathcal{V}(s = t)$ denotes the set of variables occurring in the equation, and $s_{x:=t}$ denotes the term obtained from s by substituting t for x.

$$(s = t)_{x:=\triangledown_k x} \quad =_{\mathsf{def}} \quad s_{x:=\triangledown_k x} = t_{x:=\triangledown_k x}$$

$$(s = t)_{\triangledown_k} \quad =_{\mathsf{def}} \quad \bigcup_{x\in\mathcal{V}(s=t)} (s = t)_{x:=\triangledown_k x}$$

$$\mathcal{E}_{\bigtriangledown_k} \quad =_{\mathsf{def}} \quad \bigcup_{E \in \mathcal{E}} E_{\bigtriangledown_k}$$

To give a concrete example of these definitions, let $\mathcal{E}$ consist of the following two equations:

$$\begin{aligned} x + y &= y + x \\ x + (y + z) &= (x + y) + z \end{aligned}$$

Then $\mathcal{E}_{\bigtriangleup_j}$ contains the following pair of equations:

$$\begin{aligned} \bigtriangleup_j x + \bigtriangleup_j y &= \bigtriangleup_j y + \bigtriangleup_j x \\ \bigtriangleup_j x + (\bigtriangleup_j y + \bigtriangleup_j z) &= (\bigtriangleup_j x + \bigtriangleup_j y) + \bigtriangleup_j z \end{aligned}$$

whereas $\mathcal{E}_{\bigtriangledown_k}$ is comprised of five equations in all:

$$\begin{aligned} \bigtriangledown_k x + y &= y + \bigtriangledown_k x \\ x + \bigtriangledown_k y &= \bigtriangledown_k y + x \\ \bigtriangledown_k x + (y + z) &= (\bigtriangledown_k x + y) + z \\ x + (\bigtriangledown_k y + z) &= (x + \bigtriangledown_k y) + z \\ x + (y + \bigtriangledown_k z) &= (x + y) + \bigtriangledown_k z \end{aligned}$$

We will say that a term equation has the *Gautam property* if it has the shape mentioned in Theorem 4.1. An equational specification has the Gautam property iff all of its member equations do.

4.5 Gautam logics

A *Gautam logic* is determined by the following:

- ▷ Instantiation of the language parameters $\mathcal{B}$, $\mathcal{I}$, $\mathcal{J}$ and $\mathcal{K}$;
- ▷ An equational specification $\mathcal{E}_{\mathcal{I}}$ over the signature $\{+_i\}_{i \in \mathcal{I}}$;
- ▷ A set of indices $\{i_\ell\}_{\ell \in \mathcal{J} \cup \mathcal{K}} \subseteq \mathcal{I}$;
- ▷ A set of equational specifications $\{\mathcal{E}_\ell\}_{\ell \in \mathcal{J} \cup \mathcal{K}}$, where $\mathcal{E}_\ell$ is specified over the signature $\{+_{i_\ell}\}$.

Of course, all equational specifications occurring in the above list are required to have the Gautam property. The operator $+_i$ is intended as a generic one, which is to be replaced by a specific connective or operator on each separate occasion. I will write $\mathcal{E}^\star$ for the equational specification obtained by substituting $\star_i$ for $+_i$ in $\mathcal{E}$ for all $i \in \mathcal{I}$, but will drop this superscript when it is clear from the context. $(\mathcal{E}_j)_{\bigtriangleup_j}$ will be abbreviated as $\mathcal{E}_{\bigtriangleup_j}$, and $(\mathcal{E}_k)_{\bigtriangledown_k}$ as $\mathcal{E}_{\bigtriangledown_k}$. From now on, I assume that we are dealing with a fixed Gautam logic $\mathcal{G}$. I will write $\mathcal{E}_{\mathcal{G}}$ for $\mathcal{E}_{\mathcal{I}} \cup \bigcup_{j \in \mathcal{J}} \mathcal{E}_{\bigtriangleup_j} \cup \bigcup_{k \in \mathcal{K}} \mathcal{E}_{\bigtriangledown_k}$.

4.6 Proof system

For $\mathcal{G}$ we have the following rules of inference:

$$\mathsf{A} \vdash \mathsf{A}$$

$$\frac{, \vdash \mathsf{A} \quad \Delta[\mathsf{B}] \vdash \mathsf{C}}{\Delta[(\mathsf{B}/_i\mathsf{A}) \circ_i ,] \vdash \mathsf{C}}\ [/_i\mathrm{L}] \qquad \frac{, \circ_i \mathsf{A} \vdash \mathsf{B}}{, \vdash \mathsf{B}/_i\mathsf{A}}\ [/_i\mathrm{R}]$$

$$\frac{, \vdash \mathsf{A} \quad \Delta[\mathsf{B}] \vdash \mathsf{C}}{\Delta[, \circ_i (\mathsf{A}\backslash_i\mathsf{B})] \vdash \mathsf{C}}\ [\backslash_i\mathrm{L}] \qquad \frac{\mathsf{A} \circ_i , \vdash \mathsf{B}}{, \vdash \mathsf{A}\backslash_i\mathsf{B}}\ [\backslash_i\mathrm{R}]$$

$$\frac{, [\mathsf{A}] \vdash \mathsf{B}}{, [\triangle_j\mathsf{A}] \vdash \mathsf{B}}\ [\triangle_j\mathrm{L}] \qquad \frac{, _1 \vdash \triangle_j\mathsf{B}_1 \quad \ldots \quad , _n \vdash \triangle_j\mathsf{B}_n \quad , \vdash \mathsf{A}}{, \vdash \triangle_j\mathsf{A}}\ [\triangle_j\mathrm{R}]$$

$$\frac{, [\mathsf{A}] \vdash \mathsf{B}}{, [\bigtriangledown_k\mathsf{A}] \vdash \mathsf{B}}\ [\bigtriangledown_k\mathrm{L}] \qquad \frac{, _1 \vdash \bigtriangledown_k\mathsf{B}_1 \quad \ldots \quad , _n \vdash \bigtriangledown_k\mathsf{B}_n \quad , \vdash \mathsf{A}}{, \vdash \bigtriangledown_k\mathsf{A}}\ [\bigtriangledown_k\mathrm{R}]$$

$$\frac{, \vdash \mathsf{A}}{\Delta \vdash \mathsf{A}}\ [\mathcal{E}_\mathcal{G}]$$

$$\frac{, \vdash \mathsf{A} \quad \Delta[\mathsf{A}] \vdash \mathsf{B}}{\Delta[,] \vdash \mathsf{B}}\ [\mathrm{Cut}]$$

In these rules i, j and k range over $\mathcal{I}$, $\mathcal{J}$ and $\mathcal{K}$, respectively. In the $[\triangle_j\mathrm{R}]$ and $[\bigtriangledown_k\mathrm{R}]$ rules, , must be a configuration built up from , $_1$ through , $_n$ (each occurring exactly once) by means of $\circ_{i_j}$. The $[\mathcal{E}_\mathcal{G}]$ rule schema is subject to the condition that , $=_{\mathcal{E}_\mathcal{G}^\circ}$ Δ, where the variables are assigned structures in case of an application of $\mathcal{E}_\mathcal{I}$, and types in case of an application of $\mathcal{E}_{\mathcal{J}\cup\mathcal{K}}$. This schema could alternatively have been stated thus:

$$\frac{\Theta[,] \vdash \mathsf{A}}{\Theta[\Delta] \vdash \mathsf{A}}\ [\mathcal{E}_\mathcal{G}]$$

It is however not necessary to explicitly add the context in the rule, since this is already part of the definition of equality under the equational specification.

4.7 Cut-elimination

Theorem 4.2 (Cut-elimination for $\mathcal{G}$) [Cut] *is an admissible rule of* $\mathcal{G}$.

Proof For the most part, this Cut-elimination result can be obtained by means of straightforward adaptations to the proof for $\mathbf{L}\Box'$ given in Section 3.1.1. The only essentially new cases are those non-principal Cuts where the rule $[\mathcal{E}_\mathcal{G}]$ is involved.

▷ Suppose the last step in the derivation of $, \vdash \mathsf{A}$ was an application of $[\mathcal{E}_\mathcal{G}]$. That is, we have the situation below:

$$\dfrac{\dfrac{, ' \vdash \mathsf{A}}{, \vdash \mathsf{A}}\,[\mathcal{E}_\mathcal{G}] \quad \Delta[\mathsf{A}] \vdash \mathsf{B}}{\Delta[,] \vdash \mathsf{B}}\,[\mathrm{Cut}]$$

This can be replaced by the following subderivation:

$$\dfrac{, ' \vdash \mathsf{A} \quad \Delta[,] \vdash \mathsf{B}}{\dfrac{\Delta[, '] \vdash \mathsf{B}}{\Delta[,] \vdash \mathsf{B}}\,[\mathcal{E}_\mathcal{G}]}\,[\mathrm{Cut}]$$

▷ Suppose the last step in the derivation of $\Delta[,] \vdash \mathsf{B}$ was an application of $[\mathcal{E}_\mathcal{G}]$. Because $\mathcal{E}_\mathcal{G}$ satisfies the Gautam property, we know that A is also a substructure of the premise. This means that we have the following situation:

$$\dfrac{, \vdash \mathsf{A} \quad \dfrac{\Delta'[\mathsf{A}] \vdash \mathsf{B}}{\Delta[\mathsf{A}] \vdash \mathsf{B}}\,[\mathcal{E}_\mathcal{G}]}{\Delta[,] \vdash \mathsf{B}}\,[\mathrm{Cut}]$$

This subderivation can be replaced by the one below:

$$\dfrac{, \vdash \mathsf{A} \quad \Delta'[\mathsf{A}] \vdash \mathsf{B}}{\dfrac{\Delta'[,] \vdash \mathsf{B}}{\Delta[,] \vdash \mathsf{B}}\,[\mathcal{E}_\mathcal{G}]}\,[\mathrm{Cut}]$$

□

4.8 Semantics

The basis for any model of $\mathcal{G}$ is a $(\Sigma, \mathcal{E})$-algebra $\mathcal{A}$, where $\Sigma = \{+_i\}_{i \in \mathcal{I}}$ and the product operation interpreting $\circ_i$ is denoted as $\cdot_i$. We say that $\mathcal{S} \subseteq \mathcal{A}$ is an $\mathcal{E}_j$-*subalgebra* of $\mathcal{A}$ if it is closed under $\cdot_{i_j}$, and $s^\sigma = t^\sigma$ whenever $s = t \in \mathcal{E}_j$ and $\sigma : \mathcal{V} \to \mathcal{S}$. An *easy* $\mathcal{E}_k$-*subalgebra* of $\mathcal{A}$ is a subset of $\mathcal{A}$ that is closed under $\cdot_{i_k}$, and such that $s^\sigma = t^\sigma$ whenever $s = t \in \mathcal{E}_k$ and $\sigma : \mathcal{V} \to \mathcal{A}$ assigns an element of $\mathcal{S}$ to at least one of the variables occurring in the equation. A *model* for $\mathcal{G}$ is a 4-tuple $\langle \mathcal{A}, \{\mathcal{A}_j\}_{j \in \mathcal{J}}, \{\mathcal{A}_k\}_{k \in \mathcal{K}}, [\![.]\!] \rangle$ such that:

- ▷ $\mathcal{A}$ is a $(\Sigma, \mathcal{E})$-algebra;
- ▷ $\mathcal{A}_j$ is an $\mathcal{E}_j$-subalgebra of $\mathcal{A}$ $(j \in \mathcal{J})$;
- ▷ $\mathcal{A}_k$ is an easy $\mathcal{E}_k$-subalgebra of $\mathcal{A}$ $(k \in \mathcal{K})$;
- ▷ $[\![.]\!]$ is a function $\mathcal{B} \to \mathcal{P}(\mathcal{A})$.

Here, $\mathcal{P}(\mathcal{A})$ denotes the set of all subsets of $\mathcal{A}$. The interpretation function $[\![.]\!]$ is extended to arbitrary types and structures as follows:

- $[\![\mathsf{B}/_i\mathsf{A}]\!] = \{c \in \mathcal{A} \mid \forall a \in [\![\mathsf{A}]\!] : c \cdot_i a \in [\![\mathsf{B}]\!]\}$
- $[\![\mathsf{A}\backslash_i\mathsf{B}]\!] = \{c \in \mathcal{A} \mid \forall a \in [\![\mathsf{A}]\!] : a \cdot_i c \in [\![\mathsf{B}]\!]\}$
- $[\![\triangle_j\mathsf{A}]\!] = [\![\mathsf{A}]\!] \cap \mathcal{A}_j$
- $[\![\triangledown_k\mathsf{A}]\!] = [\![\mathsf{A}]\!] \cap \mathcal{A}_k$
- $[\![\mathsf{A} \circ_i \mathsf{B}]\!] = \{c \in \mathcal{A} \mid \exists a \in [\![\mathsf{A}]\!], b \in [\![\mathsf{B}]\!] : c = a \cdot_i b\}$

4.9 Soundness and completeness

4.9.1 Soundness

The bulk of the soundness proof consists of a straightforward modification of that for $\mathbf{L}\Box'$ given in Section 3.2.1. The only essential addition in the present setup consists of the $[\mathcal{E}_\mathcal{G}]$-rule. To prove its soundness, we can of course limit ourselves to single applications of an equation in $\mathcal{E}_\mathcal{G}$. If this equation comes from $\mathcal{E}_\mathcal{I}$, soundness is an immediate consequence of the Gautam property for this set of equations. I will illustrate the cases where the equation comes from some $\mathcal{E}_{\triangle_j}$ or $\mathcal{E}_{\triangledown_k}$ by means of an example.

Suppose we apply the equation $\triangle_j x +_{i_j} \triangle_j y = \triangle_j y +_{i_j} \triangle_j x$ to a sequent antecedent. This means that we have an inference step that looks as follows:

$$\frac{, [\triangle_j\mathsf{A} \circ_{i_j} \triangle_j\mathsf{B}] \vdash \mathsf{C}}{, [\triangle_j\mathsf{B} \circ_{i_j} \triangle_j\mathsf{A}] \vdash \mathsf{C}}\ [\mathcal{E}_{\triangle_j}]$$

Now suppose that we are given an element c of $[\![, [\triangle_j\mathsf{B} \circ_{i_j} \triangle_j\mathsf{A}]]\!]$. In this term c occurs a subterm $b \cdot_{i_j} a$, with $b \in [\![\mathsf{B}]\!] \cap \mathcal{A}_j$ and $a \in [\![\mathsf{A}]\!] \cap \mathcal{A}_j$. In particular, both a and b are in $\mathcal{A}_j$, and therefore $b \cdot_{i_j} a = a \cdot_{i_j} b$. This implies that c is also in $[\![, [\triangle_j\mathsf{A} \circ_{i_j} \triangle_j\mathsf{B}]]\!]$, whence $c \in [\![\mathsf{C}]\!]$. Note that in the general case it is exactly the fact that all $\mathcal{E}_{\triangle_j}$ and $\mathcal{E}_{\triangledown_k}$ have the Gautam property which ensures that this proof goes through. The proof for modalities $\triangledown_k$ is analogous.

4.9.2 The calculus $\mathcal{G}\bullet$

In the completeness proof I will make use of the fact proved in this Section that the calculus $\mathcal{G}\bullet$, which consists of $\mathcal{G}$ with a set of product connectives $\{\bullet_i\}_{i\in\mathcal{I}}$, is a conservative extension of $\mathcal{G}$. The proof rules for the products are of course the following ones:

$$\frac{, [\mathsf{A} \circ_i \mathsf{B}] \vdash \mathsf{C}}{, [\mathsf{A} \bullet_i \mathsf{B}] \vdash \mathsf{C}}\ [\bullet\mathrm{L}] \qquad \frac{, \vdash \mathsf{A} \quad \Delta \vdash \mathsf{B}}{, \circ_i \Delta \vdash \mathsf{A} \bullet_i \mathsf{B}}\ [\bullet\mathrm{R}]$$

As expected, [Cut] is eliminable from $\mathcal{G}\bullet$:

Theorem 4.3 (Cut-elimination for $\mathcal{G}\bullet$) [Cut] *is an admissible rule of* $\mathcal{G}\bullet$.

Proof This follows almost immediately from the Cut-elimination proofs for $\mathbf{L}\bullet$ and for $\mathcal{G}$. □

Notice moreover that a proof along the lines of Section 3.2.3 shows that for both $\mathcal{G}$ and $\mathcal{G}\bullet$ we have a normalization result for Cut-free derivations, which now says that for $j \in \mathcal{J}$ we can always make sure that a premise $, _i \vdash \triangle_j \mathsf{B}_i$ of an application of $[\triangle_j \mathrm{R}]$ is not the conclusion of another application of that same rule. As before, we find that Cut-free normalized derivations satisfy the subformula property.

Now consider a $\mathcal{G}$ sequent that is derivable in $\mathcal{G}\bullet$. Then in particular it has a Cut-free normalized derivation. Since this derivation has the subformula property, the connectives $\bullet_i$ $(i \in \mathcal{I})$ do not occur in it. In other words, it is also a $\mathcal{G}$ derivation. This shows that $\mathcal{G}\bullet$ is a conservative extension of $\mathcal{G}$.

4.9.3 Completeness

For completeness, I start with defining the *canonical model* $\mathcal{M}$. Its carrier is the set $\mathbf{S}/\equiv$, where $\equiv$ is the equivalence relation defined by:

$$, \equiv \Delta \qquad \textit{iff} \qquad \forall \mathsf{A} : (, \vdash_{\mathcal{G}\bullet} \mathsf{A} \Leftrightarrow \Delta \vdash_{\mathcal{G}\bullet} \mathsf{A})$$

Note that this is defined in terms of $\mathcal{G}\bullet$-derivability, but on $\mathcal{G}$-sequents. In the remainder of this section $, \vdash \mathsf{A}$ will mean $, \vdash_{\mathcal{G}\bullet} \mathsf{A}$. The $\equiv$-equivalence class containing $,$ will be denoted as $[,]$.

On the set $\mathbf{S}/\equiv$ define products $\cdot_i$ $(i \in \mathcal{I})$ by stipulating that:

$$[,] \cdot_i [\Delta] =_{\mathsf{def}} [, \circ_i \Delta]$$

It must be shown that this is well-defined. Suppose $, \equiv , '$, $\Delta \equiv \Delta'$ and $, \circ_i \Delta \vdash \mathsf{A}$. For a structure Θ, let $\Theta^\bullet$ be the $\mathcal{G}\bullet$-type obtained from Θ by replacing each $\circ_i$ with $\bullet_i$. The sequent $\Theta^\bullet \vdash \mathsf{A}$ can be derived from $\Theta \vdash \mathsf{A}$ by a sequence of $[\bullet_i \mathrm{L}]$-rules. By definition of $\equiv$ we know that $, ' \vdash , ^\bullet$ and $\Delta' \vdash \Delta^\bullet$. Now, $, ' \circ \Delta' \vdash \mathsf{A}$ by the derivation below:

$$\dfrac{\dfrac{\dfrac{, \circ_i \Delta \vdash \mathsf{A}}{, ^\bullet \circ_i \Delta^\bullet \vdash \mathsf{A}}\,[\bullet\mathrm{L}]^* \qquad , ' \vdash , ^\bullet}{, ' \circ_i \Delta^\bullet \vdash \mathsf{A}}\,[\mathrm{Cut}] \qquad \Delta' \vdash \Delta^\bullet}{, ' \circ_i \Delta' \vdash \mathsf{A}}\,[\mathrm{Cut}]$$

Evidently, $\mathcal{M} = (\mathbf{S}/\equiv, \{\cdot_i\}_{i\in\mathcal{I}})$ is a $(\Sigma, \mathcal{E})$-algebra.

The subalgebras interpreting the modal connectives are defined as follows:

$$\begin{aligned} \mathcal{M}_{\triangle_j} &= \{[,] \mid \exists \mathsf{A} \in \mathcal{T} : , \vdash \triangle_j \mathsf{A}\} \\ \mathcal{M}_{\triangledown_k} &= \{[,] \mid \exists \mathsf{A} \in \mathcal{T} : , \vdash \triangledown_k \mathsf{A}\} \end{aligned}$$

It must be shown that these have the desired properties.

First I show that $\mathcal{M}_{\triangle_j}$ is a $\cdot_{i_j}$-subalgebra. This means that given $[, _1]$ and $[, _2]$ in $\mathcal{M}_{\triangle_j}$, we must show that $[, _1] \cdot_{i_j} [, _2]$ is also an element of $\mathcal{M}_{\triangle_j}$. This amounts to showing that if there are B_1 and B_2 such that $, _1 \vdash \triangle_j \mathsf{B}_1$ and

$, _1 \vdash \triangle_j \mathsf{B}_1$, then there exists a B such that $, _1 \circ_{i_j} , _2 \vdash \triangle_j \mathsf{B}$. This follows from the derivation below:

$$\dfrac{, _1 \vdash \triangle_j \mathsf{B}_1 \quad , _1 \vdash \triangle_j \mathsf{B}_1 \quad \dfrac{, _1 \vdash , \overset{\bullet}{_1} \quad , _2 \vdash , \overset{\bullet}{_2}}{, _1 \circ_{i_j} , _2 \vdash , \overset{\bullet}{_1} \bullet_{i_j} , \overset{\bullet}{_2}}\ [\bullet_{i_j}\mathrm{R}]}{, _1 \circ_{i_j} , _2 \vdash \triangle_j(, \overset{\bullet}{_1} \bullet_{i_j} , \overset{\bullet}{_2})}\ [\triangle_j \mathrm{R}]$$

Next I will prove that $\mathcal{M}_{\triangle_j}$ satisfies the specification $\mathcal{E}_{\triangle_j}$. Since it would be notationally awkward to have to refer to an arbitrary equational specification, I will do this by means of an example. Let $\mathcal{E}_{\triangle_j} = \{\triangle_j x +_{i_j} \triangle_j y = \triangle_j y +_{i_j} \triangle_j x\}$. Supposing that $[, _1], [, _2] \in \mathcal{M}_{\triangle_j}$ we must prove that $[, _1] \cdot_{i_j} [, _2] = [, _2] \cdot_{i_j} [, _1]$, i.e. that for all types A, $, _1 \circ_{i_j} , _2 \vdash \mathsf{A}$ iff $, _2 \circ_{i_j} , _1 \vdash \mathsf{A}$. This follows from the derivation below, in which the two unmarked inferences are instances of $[\triangle_j \mathrm{R}]$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{, _1 \circ_{i_j} , _2 \vdash \mathsf{A}}{, \overset{\bullet}{_1} \circ_{i_j} , \overset{\bullet}{_2} \vdash \mathsf{A}}\ [\bullet_{i_j}\mathrm{L}]^*}{\triangle_j, \overset{\bullet}{_1} \circ_{i_j} , \overset{\bullet}{_2} \vdash \mathsf{A}}\ [\triangle_j \mathrm{L}]}{\triangle_j, \overset{\bullet}{_1} \circ_{i_j} \triangle_j, \overset{\bullet}{_2} \vdash \mathsf{A}}\ [\triangle_j \mathrm{L}]}{\triangle_j, \overset{\bullet}{_2} \circ_{i_j} \triangle_j, \overset{\bullet}{_1} \vdash \mathsf{A}}\ [\mathcal{E}_{\triangle_j}] \quad \dfrac{, _2 \vdash , \overset{\bullet}{_2} \quad , _2 \vdash \triangle_j \mathsf{B}_2}{, _2 \vdash \triangle_j, \overset{\bullet}{_2}}\ [\mathrm{Cut}]}{, _2 \circ_{i_j} \triangle_j, \overset{\bullet}{_1} \vdash \mathsf{A}} \quad \dfrac{, _1 \vdash , \overset{\bullet}{_1} \quad , _1 \vdash \triangle_j \mathsf{B}_1}{, _1 \vdash \triangle_j, \overset{\bullet}{_1}}\ [\mathrm{Cut}]}{, _2 \circ_{i_j} , _1 \vdash \mathsf{A}}\ [\mathrm{Cut}]$$

The proof for $\mathcal{M}_{\triangledown_k}$ is similar.

Finally, we set $[\![\mathsf{p}]\!] = \{[,] \mid , \vdash \mathsf{p}\}$ for all $\mathsf{p} \in \mathcal{B}$, which completes our definition of the canonical model.

Lemma 4.4 (Canonical Lemma) $[\![\mathsf{A}]\!] = \{[,] \mid , \vdash \mathsf{A}\}$ *for all* $\mathsf{A} \in \mathcal{T}$.

Proof This is proved by induction on the complexity of the type A. I omit the part of the proof that is similar to the completeness proof for **L** given in Section 1.2.3, and concentrate on the case where A is a modalized type. I give the proof for a modality $\triangle_j$; that for a modality $\triangledown_k$ is similar. Cf. also Section 3.1.4.

1. First, suppose $[,] \in [\![\triangle_j \mathsf{B}]\!] = [\![\mathsf{B}]\!] \cap \mathcal{M}_{\triangle_j}$. Then in particular $[,] \in [\![\mathsf{B}]\!]$. From this the induction hypothesis allows us to conclude that $, \vdash \mathsf{B}$. From $[,] \in [\![\mathsf{B}]\!] \cap \mathcal{M}_{\triangle_j}$ we also infer that $[,] \in \mathcal{M}_j$, which by definition implies the existence of some type C such that $, \vdash \triangle_j \mathsf{C}$. By applying the $[\triangle_j \mathrm{R}]$ rule to both sequents just found we arrive at the required sequent $, \vdash \triangle_j \mathsf{B}$.

2. Conversely, suppose that $, \vdash \triangle_j \mathsf{B}$. Then $, \vdash \mathsf{B}$:

$$\dfrac{, \vdash \triangle_j \mathsf{B} \quad \dfrac{\mathsf{B} \vdash \mathsf{B}}{\triangle_j \mathsf{B} \vdash \mathsf{B}}\ [\triangle_j \mathrm{L}]}{, \vdash \mathsf{B}}\ [\mathrm{Cut}]$$

   From this we conclude on the basis of the induction hypothesis that $[,] \in [\![\mathsf{B}]\!]$. Also, $[,] \in \mathcal{M}_j$ by definition. Therefore, $[,] \in [\![\mathsf{B}]\!] \cap \mathcal{M}_j = [\![\triangle_j \mathsf{B}]\!]$.

□

Completeness follows from the canonical lemma by a slight modification of the argument of Section 1.2.3. Indeed, suppose that the $\mathcal{G}$-sequent $, \vdash \mathsf{A}$ is not derivable in that system. Then it isn't derivable in $\mathcal{G}\bullet$ either, because the latter logic is a conservative extension of the former. Lemma 4.4 now allows us to conclude that $[,] \notin [\![\mathsf{A}]\!]$ in $\mathcal{M}$. Since $[,] \in [\![[,]]\!]$, this implies that $[\![[,]]\!] \not\subseteq [\![\mathsf{A}]\!]$. In other words, $, \vdash \mathsf{A}$ is not valid in the canonical model, and hence it is not generally valid.

4.10 Venema's approach

Algebraic semantics for multimodal categorial logics was also investigated by Venema (1994). His approach differs from the one I presented in the last section in several respects.

Recall that the basic intuition underlying the definition of the modalities was that they mark elements as being in some sense special. In the semantics, this was modelled by interpreting a modalized type $\Box\mathsf{A}$ as the intersection of the interpretation of A with a subalgebra of special elements $\mathbf{L}_\Box$. Venema's proof theory reflects this semantic definition more directly than does ours. He adds to the inventory of the languge a type constant Q, which explicitly refers to $\mathbf{L}_\Box$.

Venema's semantics is based on algebraic structures called *semi-lattice-ordered groupoids*, or *slogs* for short, which were introduced in the categorial literature by Došen (1988, 1989). The main difference between slogs and the algebras I used in this chapter is that slogs are endowed with a partial order $\leq$ that is well-behaved with respect to the multiplication operation (Venema only considers systems having one mode of binary combination). The partial ordering is used in the definition of the interpretation of $\circ$, which in Venema's setup looks as follows:

$$[\![\mathsf{A} \circ \mathsf{B}]\!] = \{c \in \mathcal{A} \mid \exists a \in [\![\mathsf{A}]\!], b \in [\![\mathsf{B}]\!] : a \cdot b \leq c\}$$

This semantics originated from Wansing's (1992) theory of information structures, and can be motivated by seeing $\mathcal{E}$ as an information ordering. The advantage of this more complicated semantics is that it is capable of dealing with inequalities.

It is possible to incorporate Venema's idea of having type constants which explicitly refer to subalgebras into a logical system set up in a way similar to those presented in this chapter. The resulting proof system looks as follows:

$$\mathsf{A} \vdash \mathsf{A}$$

$$\frac{, \vdash \mathsf{A} \quad \Delta[\mathsf{B}] \vdash \mathsf{C}}{\Delta[(\mathsf{B}/_i\mathsf{A}) \circ_i ,] \vdash \mathsf{C}}\ [/_i\mathrm{L}] \qquad \frac{, \circ_i \mathsf{A} \vdash \mathsf{B}}{, \vdash \mathsf{B}/_i\mathsf{A}}\ [/_i\mathrm{R}]$$

$$\frac{, \vdash \mathsf{A} \quad \Delta[\mathsf{B}] \vdash \mathsf{C}}{\Delta[, \circ_i (\mathsf{A}\backslash_i\mathsf{B})] \vdash \mathsf{C}}\ [\backslash_i\mathrm{L}] \qquad \frac{\mathsf{A} \circ_i , \vdash \mathsf{B}}{, \vdash \mathsf{A}\backslash_i\mathsf{B}}\ [\backslash_i\mathrm{R}]$$

$$\frac{,[\mathsf{A}]\vdash \mathsf{B}}{,[\triangle_j \mathsf{A}]\vdash \mathsf{B}}\ [\triangle_j \mathrm{L1}] \qquad \frac{,[\mathsf{Q}_j]\vdash \mathsf{B}}{,[\triangle_j \mathsf{A}]\vdash \mathsf{B}}\ [\triangle_j \mathrm{L2}]$$

$$\frac{,_1\vdash \triangle_j \mathsf{B}_1 \quad \ldots \quad ,_n\vdash \triangle_j \mathsf{B}_n \quad ,\vdash \mathsf{A}}{,\vdash \triangle_j \mathsf{A}}\ [\triangle_j \mathrm{R}]$$

$$\frac{,[\mathsf{A}]\vdash \mathsf{B}}{,[\bigtriangledown_k \mathsf{A}]\vdash \mathsf{B}}\ [\bigtriangledown_k \mathrm{L1}] \qquad \frac{,[\mathsf{Q}_j]\vdash \mathsf{B}}{,[\bigtriangledown_k \mathsf{A}]\vdash \mathsf{B}}\ [\bigtriangledown_k \mathrm{L2}]$$

$$\frac{,_1\vdash \bigtriangledown_k \mathsf{B}_1 \quad \ldots \quad ,_n\vdash \bigtriangledown_k \mathsf{B}_n \quad ,\vdash \mathsf{A}}{,\vdash \bigtriangledown_k \mathsf{A}}\ [\bigtriangledown_k \mathrm{R}]$$

$$\frac{,_1\vdash \mathsf{Q}_\ell \quad ,_2\vdash \mathsf{Q}_\ell \quad ,[\mathsf{Q}_\ell]\vdash \mathsf{A}}{,[,_1 \circ_{i_\ell} ,_2]\vdash \mathsf{A}}\ [\mathsf{Q}_\ell]$$

$$\frac{,\vdash \mathsf{A}}{\Delta\vdash \mathsf{A}}\ [\mathcal{E}_{\mathcal{I}}] \qquad \frac{,\vdash \mathsf{A} \quad ,_1\vdash \mathsf{Q}_1 \quad \ldots \quad ,_n\vdash \mathsf{Q}_n}{\Delta\vdash \mathsf{A}}\ [\mathcal{E}_\ell]$$

$$\frac{,\vdash \mathsf{A} \quad \Delta[\mathsf{A}]\vdash \mathsf{B}}{\Delta[,]\vdash \mathsf{B}}\ [\mathrm{Cut}]$$

The restrictions on the $[\mathcal{E}]$-rules are similar to those formulated in Section 4.9.3. It can be shown that this alternative formulation of Gautam logics is also sound and complete, enjoys Cut-elimination and the subformula property, and is decidable. The reader who has studied this section carefully should find no trouble in adapting our proofs to the new system. The adapted proofs can also be found in Versmissen (1993).

In linear logic there is a conjunction connective $\sqcap$ with the following proof rules:

$$\frac{,\vdash \mathsf{A} \quad ,\vdash \mathsf{B}}{,\vdash \mathsf{A}\sqcap \mathsf{B}}\ [\sqcap \mathrm{R}] \qquad \frac{,[\mathsf{A}]\vdash \mathsf{C}}{,[\mathsf{A}\sqcap \mathsf{B}]\vdash \mathsf{C}}\ [\sqcap \mathrm{L}]$$

The rules for the modalities in Venema's system are closely related to this conjunction operator: $\Box\mathsf{A}$ can be read as $\mathsf{Q}_\Box \sqcap \mathsf{A}$. The rule for the modalities can be derived from this correspondence. It also suggests a modification of the notion of subtype, in which $\Box\mathsf{A}$ has two subtypes, namely A and $\mathsf{A}_\Box$. With this definition of subtype, Cut-free derivations in Venema's (and the above) system have the subformula property. This means that these calculi are somewhat 'nicer' from a metalogical point of view than Gautam logics, in which the subformula property does not hold in general.[1] It would be possible to add $\sqcap$ to the language of the logic, to get a logic that is even more satisfying from a metalogical viewpoint. However, this logic would have two constants, Q and $\sqcap$ that are not essential to the intended use of the calculus. In Part III a different approach to modalities will be introduced which overcomes these objections.

[1] Of course, since we have shown that every sequent has a derivation satisfying the subformula property, we can restrict the proof system to such derivations. This doesn't really make it any more attractive, though, since this would be an ad hoc restriction.

Part III

Frame semantics, embeddings and conjoinability

The multimodal calculi of the last part were geared towards an algebraic semantics. The ones taking the stage in this part are rather based on the more general *frame semantics* mentioned in Section 2.4.

Chapter 5 starts with an introduction to this semantics, and of the related issue of *residuation*. The case of *unary* residuation, which is of particular interest, is discussed in detail. I discuss the article Kurtonina and Moortgat (1995), and prove some alternative embedding theorems to the ones that can be found there.

In Chapter 6 the related issues of derivability invariants and the decidability of the *conjoinability* relation are discussed. In particular, I show that for all systems presented in Kurtonina and Moortgat (1995) it is decidable whether or not two types are conjoinable.

Chapter 5

Residuation modalities and embeddings

In the last two chapters I studied multimodal categorial grammars from an algebraic point of view. Although this turned out to be quite fruitful, the approach presented there is perhaps not as general as one might sometimes wish. One of the more serious shortcomings of Gautam logics is the fact that they only allow *equalities* as structural rules. In practice it is often desirable to have the possibility of using *implicational* structural rules. For example, in Morrill *et al.* (1990) the introduction of both a weakening and a contraction modality is motivated through linguistic examples. In the case of *restructuring* phenomena, see e.g. Moortgat and Oehrle (1993a), one usually also wants the application of the rules to go in one direction only.

As I already noted in Section 4.10, a partial answer is provided by Venema (1994). However, Venema only discusses a few very basic rules such as weakening and contraction, and has no general result. Moreover, his semantics is not the most intuitive from a linguistic point of view.

This and the next chapter will be based on an alternative semantics for multimodal categorial logics, which expresses the properties of the composition and structuring of linguistic resources through *(Kripke) frames*. This new approach suggests a view of modal operators that is quite different from the one presented earlier.

5.1 Frame semantics and residuation modalities

5.1.1 Frame semantics

As I already indicated in Section 2.4, frame semantics can be seen as a generalization of the algebraic semantics presented in Part II. The generalization consists of no longer having product operations in the semantic algebra which explicitly encode the notion of linguistic structuring. Rather, using ideas stem-

ming from relevant logic[1] the product can be modeled in terms of a ternary relation R, which implicitly encodes structure. Sets endowed with such relations are known as *(Kripke) frames*. To model a single binary product, a single ternary relation is needed, hence a frame $\langle W, R\rangle$ where W is some set, and R a ternary relation on W. In a multimodal setting, a different relation is needed for each mode. I.e. if the set of modes is $\{\circ_i\}_{i\in\mathcal{I}}$, then the corresponding frame will be $\langle W, \{\mathrm{R}_i\}_{i\in\mathcal{I}}\rangle$, where all the R_i are ternary relations on W.

Recall that the basic equivalences relating the behaviour of the connectives $\bullet_i$, $/_i$ and $\backslash_i$ to each other are the residuation equations:

$$\mathsf{A} \vdash \mathsf{C}/_i\mathsf{B} \qquad \textit{iff} \qquad \mathsf{A} \bullet_i \mathsf{B} \vdash \mathsf{C} \qquad \textit{iff} \qquad \mathsf{B} \vdash \mathsf{A}\backslash_i\mathsf{C}$$

$\mathrm{R}_i cab$ is to be read as: in mode i, the structures a and b can combine into the structure c. The clauses for the interpretation of the connectives need to be modified accordingly:

$$\begin{aligned}
[\![\mathsf{A} \bullet_i \mathsf{B}]\!] &= \{c \in W \mid \exists a \in [\![\mathsf{A}]\!], b \in [\![\mathsf{B}]\!] : \mathrm{R}_i cab\} \\
[\![\mathsf{C}/_i\mathsf{B}]\!] &= \{a \in W \mid \forall c \in W((b \in [\![\mathsf{B}]\!] \;\&\; \mathrm{R}_i cab) \rightarrow c \in [\![\mathsf{C}]\!])\} \\
[\![\mathsf{A}\backslash_i\mathsf{C}]\!] &= \{b \in W \mid \forall c \in W((a \in [\![\mathsf{A}]\!] \;\&\; \mathrm{R}_i cab) \rightarrow c \in [\![\mathsf{C}]\!])\}
\end{aligned}$$

Let me say a little more about the intuitions behind this approach. In a simple system such as the Lambek calculus, the combinatory properties of linguistic entitites on the string level are coded very directly into the types assigned to them. In other words, **L** can be described as being strongly *surface oriented*. This is why semigroups are such a natural starting point for the semantics of **L**. The move to multimodal systems is partly motivated by the recognition that the surface order of strings can be due to the interaction of several different modes of structuring, none of which may be clearly recognizable on the string level exactly because of this interaction. This shows itself in the difficulties encountered by attempts to define the semantics of multimodal systems in terms of concatenation.

The algebraic semantics of Part II can be seen as an attempt to *explicitly* model the structural behaviour and interaction properties of several coexisting modes of composition. Of course, such explicit modeling requires a thorough understanding of what goes on in actual fact. Furthermore, a *uniform* approach to such an explicit interpretation presupposes that all modes of composition display largely similar behavior. These are some of the reasons why the semantics of Part II was only partly successful.

Frame semantics can be contrasted with the algebraic approach by saying that it takes a more abstract stance where the actual structures and processes involved in linguistic composition are concerned. Rather than pretending to know exactly what goes on, we describe the relevant behavioral characteristics of linguistic resources. This explains the move in the semantics to relations. These do exactly that: they encode the properties of the presumed structuring operators, without telling exactly how or on what they operate. Frame semantics can thus be seen as very close in spirit to *constraint-based* grammar formalisms.

[1] And ultimately from modal logic, cf. Kripke (1963).

5.1.2 Residuation

I will now examine residuation a little closer. The exposition is based on Blyth and Janowitz (1972). Another basic reference is Fuchs (1963).

The equivalences that I gave earlier can be seen as instances of a more general scheme. Let $\mathcal{A} = \langle A, \leq_A \rangle$ and $\mathcal{B} = \langle B, \leq_B \rangle$ be partially ordered sets. The pair of functions (f, g), where $f : A \to B$ and $g : B \to A$ is called *residuated* if they are related in the following way:

$$fx \leq_B y \qquad \textit{iff} \qquad x \leq_A gy$$

It is easy to show that the following four clauses together are equivalent to the above definition (cf. Section 1.2.4):

$$\text{if } x \leq_B y, \text{ then } fx \leq_B fy$$

$$\text{if } x \leq_A y, \text{ then } gx \leq_A gy$$

$$x \leq_A gfx$$

$$fgx \leq_B x$$

In the present setting $\mathcal{A}$ and $\mathcal{B}$ will always be the same partially ordered set, namely $\langle \mathcal{T}, \vdash \rangle$. The binary residuation equations can be seen as instances of the general scheme by setting $f = - \bullet \mathsf{B}$ and $g = -/\mathsf{B}$ for the first one (giving $\mathsf{A} \bullet \mathsf{B} \vdash \mathsf{C}$ *iff* $\mathsf{A} \vdash \mathsf{C}/\mathsf{B}$), and $f = \mathsf{A} \bullet -$ and $g = \mathsf{A}\backslash -$ for the second one (giving $\mathsf{A} \bullet \mathsf{B} \vdash \mathsf{C}$ *iff* $\mathsf{B} \vdash \mathsf{A}\backslash\mathsf{C}$). The alternative formulation in this case amounts to the following:

$$\frac{\mathsf{A} \vdash \mathsf{A}'}{\mathsf{A} \bullet \mathsf{B} \vdash \mathsf{A}' \bullet \mathsf{B}} \qquad \frac{\mathsf{A} \vdash \mathsf{A}'}{\mathsf{A}/\mathsf{B} \vdash \mathsf{A}'/\mathsf{B}} \qquad \frac{\mathsf{A} \vdash \mathsf{A}'}{\mathsf{B} \bullet \mathsf{A} \vdash \mathsf{B} \bullet \mathsf{A}'} \qquad \frac{\mathsf{A} \vdash \mathsf{A}'}{\mathsf{B}\backslash\mathsf{A} \vdash \mathsf{B}\backslash\mathsf{A}'}$$

$$\mathsf{A} \vdash (\mathsf{A} \bullet \mathsf{B})/\mathsf{B} \qquad\qquad \mathsf{A} \vdash \mathsf{B}\backslash(\mathsf{B} \bullet \mathsf{A})$$

$$(\mathsf{A}/\mathsf{B}) \bullet \mathsf{B} \vdash \mathsf{A} \qquad\qquad \mathsf{B} \bullet (\mathsf{B}\backslash\mathsf{A}) \vdash \mathsf{A}$$

More generally, just like a binary product gives rise to a pair of residuation equations, an n-ary product gives rise to n residuation equations as follows. Denote the n-ary product by $\bullet(x_1, \ldots, x_n)$ and its i-th place residual as $/_i(x_1, \ldots, x_n)$. The n residuation equations are then found by setting $f = \bullet(\mathsf{A}_1, \ldots, \mathsf{A}_{i-1}, -, \mathsf{A}_{i+1}, \ldots, \mathsf{A}_n)$ and $g = /_i(\mathsf{A}_1, \ldots, \mathsf{A}_{i-1}, -, \mathsf{A}_{i+1}, \ldots, \mathsf{A}_n)$. This gives us:

$$\bullet(\mathsf{A}_1, \ldots, \mathsf{A}_{i-1}, \mathsf{B}, \mathsf{A}_{i+1}, \ldots, \mathsf{A}_n) \vdash \mathsf{C} \quad \textit{iff} \quad \mathsf{B} \vdash /_i(\mathsf{A}_1, \ldots, \mathsf{A}_{i-1}, \mathsf{C}, \mathsf{A}_{i+1}, \ldots, \mathsf{A}_n)$$

The semantic equivalent of such an n-ary product is an $n+1$-ary relation R. The interpretation clauses for the connectives become:

$$[\![\bullet(\mathsf{A}_1, \ldots, \mathsf{A}_n)]\!] = \{c \mid \exists a_1 \cdots a_n (Rca_1 \ldots a_n \;\&\; a_j \in [\![\mathsf{A}_j]\!] \; (1 \leq j \leq n))\}$$

$$[\![/_i(\mathsf{A}_1, \ldots, \mathsf{A}_n)]\!] = \{c \mid \forall a_1 \cdots a_n ((Ra_i a_1 \cdots a_{i-1} c a_{i+1} \cdots a_n \;\&\; a_j \in [\![\mathsf{A}_j]\!] (j \neq i)) \to a_i \in [\![\mathsf{A}_i]\!])\}$$

A general discussion of generalized residuation and its relation to non-classical logics can be found in Dunn (1991), while Buszkowski (1988) was

the first author to consider it from a categorial point of view. A recent, comprehensive discussion of residuation from a categorial perspective can be found in Moortgat and Oehrle (1993b).

5.1.3 Unary residuation

A case of particular interest is that of *unary* residuation. In the type logical setting this means we have two unary connectives, or *modalities*. These will be written as $\Diamond$ and $\Box$. They are connected through the following equivalence:

$$\Diamond A \vdash B \qquad \textit{iff} \qquad A \vdash \Box B$$

Alternatively, unary residuation can be characterized as follows:

$$\frac{A \vdash B}{\Diamond A \vdash \Diamond B} \qquad \frac{A \vdash B}{\Box A \vdash \Box B}$$

$$\Diamond\Box A \vdash A \vdash \Box\Diamond A$$

The interpretation of $\Diamond$ and $\Box$ is defined with respect to a binary relation R as follows:

$$\begin{aligned} [\![\Diamond A]\!] &= \{b \mid \exists a(R^2 ba \ \&\ a \in [\![A]\!]\} \\ [\![\Box B]\!] &= \{a \mid \forall b(R^2 ba \rightarrow b \in [\![B]\!]\} \end{aligned}$$

At this point, a warning about the notation is in order. The interrelationship between $\Diamond$ and $\Box$ is fundamentally different from that of the similarly notated connectives in e.g. tense logic. There, $\Diamond$ and $\Box$ express existential and universal quantification over the same relation R. Here, $\Diamond$ also expresses existential quantification over R, but $\Box$ expresses universal quantification over the *converse* relation $R^{\smile}$.

For the Gentzen formulation of the rules for $\Diamond$ and $\Box$ we need to introduced a unary structural operator for sequent antecedents, just like binary structuring of antecedents is used in the account of the product. Given this structural operator, which will be written as $(.)^\Diamond$, the rules for $\Diamond$ and $\Box$ can be formulated thus:

$$\frac{, \vdash A}{(,)^\Diamond \vdash \Diamond A}\ [\Diamond R] \qquad \frac{,[(A)^\Diamond] \vdash B}{,[\Diamond A] \vdash B}\ [\Diamond L]$$

$$\frac{(,)^\Diamond \vdash A}{, \vdash \Box A}\ [\Box R] \qquad \frac{,[A] \vdash B}{,[(\Box A)^\Diamond] \vdash B}\ [\Box L]$$

As a simple illustration of how these rules interact, I show below the derivation of the four sequents which show that the sequents in the alternative characterization of residuation given above are indeed theorems:

$$\dfrac{\dfrac{A \vdash B}{(A)^\Diamond \vdash \Diamond B}\ [\Diamond R]}{\Diamond A \vdash \Diamond B}\ [\Diamond L] \qquad \dfrac{\dfrac{A \vdash B}{(\Box A)^\Diamond \vdash B}\ [\Box L]}{\Box A \vdash \Box B}\ [\Box R]$$

$$\dfrac{\dfrac{A \vdash A}{(\Box A)^\Diamond \vdash A}\ [\Box L]}{\Diamond\Box A \vdash A}\ [\Diamond L] \qquad \dfrac{\dfrac{A \vdash A}{(A)^\Diamond \vdash \Diamond A}\ [\Diamond R]}{A \vdash \Box\Diamond A}\ [\Box R]$$

Proofs of the equivalence of the axiomatic and Gentzen rules and a Cut-elimination for this calculus can be found in Moortgat (to appear).

5.1.4 Extended residuation logics

Of course, our real interest isn't so much in *basic* logics of residuation such as **NL**, but rather in *extended* systems. Like in Chapter 4, the extensions that we are interested in are structural rules and interaction postulate, for both products and modalities. The question is whether the semantics for basic residuation logics can be adapted in such a way that soundness and completeness results for extended logics can be obtained.

The first author to deal with some of these questions was Došen (1992). He proved that associativity and commutativity for the product correspond to the following conditions on ternary frames:

(5.1) $$\mathrm{R}cba \quad \textit{iff} \quad \mathrm{R}cab$$

(5.2) $$\exists d : \mathrm{R}dab \,\&\, \mathrm{R}edc \quad \textit{iff} \quad \exists d' : \mathrm{R}d'bc \,\&\, \mathrm{R}ad'e$$

The notion *correspondence* here means that the frames characterized by these first-order formulas are exactly the ones on which the structural rules of associativity and commutativity hold, respectively.

These results were generalized by Kurtonina (1995), who also considered interaction postulates and modalities. She proves several correspondence and completeness results based on a generalization of the Sahlqvist-van Benthem algorithm for modal logic (cf. van Benthem (1985)). Important for our purposes is her completeness theorem, stated below:

Theorem 5.1 *Let* $\mathsf{A} \vdash \mathsf{B}$ *be a categorial sequent which satisfies the following conditions:*

1. *Both* A *and* B *are constructed entirely out of atoms, products and diamonds;*
2. *The type* B *is not an atom;*
3. *No proposition letter occurs more than once in* A*;*
4. *All proposition letters occurring in* B *occur in* A *as well.*

Then there exists a first-order condition φ*, effectively computable from* $\mathsf{A} \vdash \mathsf{B}$*, such that* $\mathbf{L}\Diamond$ *with* $\mathsf{A} \vdash \mathsf{B}$ *added is complete with respect to the class of frames satisfying* φ*.*

Proof See Kurtonina (1995). □

5.2 Structural control

5.2.1 Introduction

The motivation for this section was the 1995 article *Structural Control* by Natasha Kurtonina and Michael Moortgat. In that paper, the authors present

a theory of *communication* between several systems in a hierarchy of categorial logics through embedding theorems. Their aim is the formulation of logics in which it is possible to specify to which structural rules certain material has access by employing embeddings between the different calculi in the structural hierarchy.

Kurtonina & Moortgat concentrate on three structural parameters, those of precedence, dominance and dependency. Their most sensitive system, **DNL**, is a truly multimodal one, having two binary operators, $\bullet_l$ and $\bullet_r$. The postulates corresponding to the linguistic parameters just mentioned are the following.

$$\begin{array}{lrcl}
(\mathrm{A}_l) & (\mathsf{A} \bullet_l \mathsf{B}) \bullet_l \mathsf{C} & \Leftrightarrow & \mathsf{A} \bullet_l (\mathsf{B} \bullet_l \mathsf{C}) \\
(\mathrm{A}_r) & (\mathsf{A} \bullet_r \mathsf{B}) \bullet_r \mathsf{C} & \Leftrightarrow & \mathsf{A} \bullet_r (\mathsf{B} \bullet_r \mathsf{C}) \\
(\mathrm{P}_{l,r}) & \mathsf{A} \bullet_l \mathsf{B} & \Leftrightarrow & \mathsf{B} \bullet_r \mathsf{A} \\
(\mathrm{D}) & \mathsf{A} \bullet_l \mathsf{B} & \Leftrightarrow & \mathsf{A} \bullet_r \mathsf{B}
\end{array}$$

Insensitivity to the parameters mentioned above is expressed by adding the appropriate postulates: A_l and A_r together indicate insensitivity to dominance, addition of $\mathrm{P}_{l,r}$ implies lack of sensitivity to precedence, and finally D expresses insensitivity to the dominance parameter. In the Gentzen presentation these postulates are implemented as restructuring rules for sequent antecedents:

$$\frac{, [(\mathsf{A} \bullet_l \mathsf{B}) \bullet_l \mathsf{C}] \vdash \mathsf{D}}{, [\mathsf{A} \bullet_l (\mathsf{B} \bullet_l \mathsf{C})] \vdash \mathsf{D}}\ [\mathrm{A}_l] \qquad \frac{, [\mathsf{A} \bullet_l \mathsf{B}] \vdash \mathsf{C}}{, [\mathsf{B} \bullet_r \mathsf{A}] \vdash \mathsf{C}}\ [\mathrm{P}_{r,l}]$$

$$\frac{, [(\mathsf{A} \bullet_r \mathsf{B}) \bullet_r \mathsf{C}] \vdash \mathsf{D}}{, [\mathsf{A} \bullet_r (\mathsf{B} \bullet_r \mathsf{C})] \vdash \mathsf{D}}\ [\mathrm{A}_r] \qquad \frac{, [\mathsf{A} \bullet_l \mathsf{B}] \vdash \mathsf{C}}{, [\mathsf{A} \bullet_r \mathsf{B}] \vdash \mathsf{C}}\ [\mathrm{D}]$$

The different instantiations of the options with respect to each of the three parameters mentioned above lead to the following eight calculi:

$$\begin{array}{rcl}
\mathbf{DNL} & = & \text{the calculus given above} \\
\mathbf{DL} & = & \mathbf{DNL} + \mathrm{A}_{l,r} \\
\mathbf{DNLP} & = & \mathbf{DNL} + \mathrm{P}_{r,l} \\
\mathbf{NL} & = & \mathbf{DNL} + \mathrm{D} \\
\mathbf{DLP} & = & \mathbf{DNL} + \mathrm{A}_{l,r} + \mathrm{P}_{r,l} \\
\mathbf{L} & = & \mathbf{DNL} + \mathrm{A}_{l,r} + \mathrm{D} \\
\mathbf{NLP} & = & \mathbf{DNL} + \mathrm{P}_{r,l} + \mathrm{D} \\
\mathbf{LP} & = & \mathbf{DNL} + \mathrm{A}_{l,r} + \mathrm{P}_{r,l} + \mathrm{D}
\end{array}$$

The relationships between these systems are shown in Figure 5.1, which can be interpreted as the Hasse diagram of the subset ordering on the sets of derivable sequents of the calculi listed.

5.2.2 Embeddings

Kurtonina & Moortgat's goal is to arrive at a theory of *communication* between modes of composition with different structural properties. They achieve this by

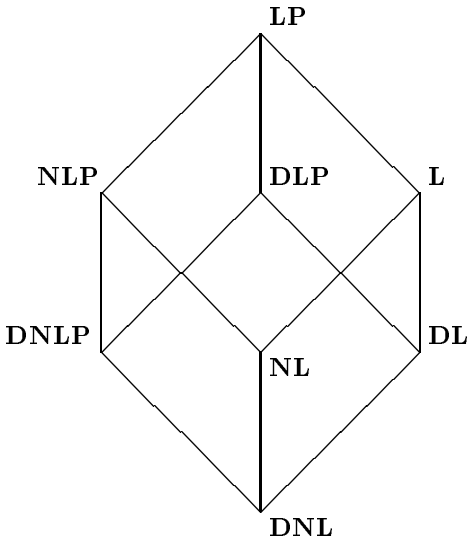


Figure 5.1: *Resource-sensitive logics: precedence, dominance, dependency*

establishing *embeddings* between such systems. More concretely, given any two systems that are on the same edge in Figure 5.1, they show how each system can in a sense be embedded in the other. Note that this involves two different kinds of embeddings. Going down along the edges one arrives at a system in which fewer structural rules are present. This means that something is needed to make the extra structural rules that hold in the 'higher' calculus available on the lower level. In other words, the embedding needs to somehow license the structural rules present in the original calculus. Kurtonina & Moortgat achieve this by adding a structural modality to the goal calculus which exhibits the required behavior. The product from the original calculus is then translated in terms of this modality. The case that is more relevant in the present context is the other direction, where types from the weaker original calculus must be prevented from gaining access to the additional structural rules present in the goal calculus. Modalities can be used for this too. In most cases, a simple residuation modality without structural rules suffices for this. The following is an instance of such an embedding result, which will be used in the next chapter.

Theorem 5.2 *Let* $(.)^\flat : \mathbf{NL} \to \mathbf{L}\Diamond$ *be defined inductively as follows:*

$$\begin{aligned} \mathsf{p}^\flat &= \mathsf{p} \\ (\mathsf{A} \bullet \mathsf{B})^\flat &= \Diamond(\mathsf{A}^\flat \bullet \mathsf{B}^\flat) \\ (\mathsf{C}/\mathsf{B})^\flat &= \Box\mathsf{A}^\flat/\mathsf{B}^\flat \\ (\mathsf{A}\backslash\mathsf{C})^\flat &= \mathsf{B}^\flat\backslash\Box\mathsf{A}^\flat \end{aligned}$$

Then $\mathsf{A} \vdash_{\mathbf{NL}} \mathsf{B}$ *iff* $\mathsf{A}^\flat \vdash_{\mathbf{L}\Diamond} \mathsf{B}^\flat$.

Proof See Kurtonina and Moortgat (1995). □

Note that the translation of $\mathsf{C/B}$ and $\mathsf{A\backslash C}$ follows from that of $\mathsf{A} \bullet \mathsf{B}$ through the residuation equivalences. In other words, the essence of the translation can be gathered from the clause for $\mathsf{A} \bullet \mathsf{B}$. This is true of all embeddings in this and the following chapter.

Some reservations can be had about the question to what extent the embeddings results of Kurtonina & Moortgat achieve their purported goals. After all, an embedding $\mathcal{L}_1 \to \mathcal{L}_2$ is not sufficient for establishing real communication between the different modes of composition in these calculi. What is required in addition is knowledge about the *interaction* of 'genuine' $\mathcal{L}_2$-sequents with translations of $\mathcal{L}_1$-sequents. Some examples given in the paper suggest that these different kinds of sequent do indeed interact in a sensible way, but it would be pleasant if this could somehow be given more formal content. An additional problem in this respect may be the fact that communication between $\mathcal{L}_1$ and $\mathcal{L}_2$ is not established by way of an embedding $\mathcal{L}_1 \to \mathcal{L}_2$, but rather through an embedding $\mathcal{L}_1 \to \mathcal{L}_2\Diamond$, where in some cases structural rules need to be posited for $\Diamond$. This aspect of Kurtonina & Moortgat's embeddings raises another question, namely whether they can be chained in order to travel between systems that are not on the same edge in the cube. This seems to be an underlying assumption of the paper, since embedding theorems are only established for systems that are on adjacent corners in the cube of Figure 5.1. However, the addition of modalities along the way prevents a straightforward composition of the embeddings go along different edges of the cube.

In the next section I will discuss an alternative kind of embedding to those of Kurtonina & Moortgat. The advantage of this new type of embedding lies in the fact that they do not use additional modalities, but rather additional basic types, which doesn't change the character of the logic. This way, certain properties of logics with unary residuals can be reduced to properties of better-known calculi with product connectives. In particular, I will show in the next chapter how the embeddings of Section 5.3 can be used to find derivability invariants for logics with unary residuals.

5.3 Embedding without modalities

In the absence of structural rules, the only difference between the unary modality $\Diamond$ and the binary modality $\bullet$ is their arity. This suggests that it might be possible to translate the behaviour of $\Diamond$ in terms of $\bullet$ by letting the two arguments of $\bullet$ be the one argument of $\Diamond$ and a dummy argument, for which an extra basic type could be used. If we let $\mathcal{L}(\mathcal{B})$ denote the calculus $\mathcal{L}$ with $\mathcal{B}$ for its set of basic types, the embedding $(.)^\flat : \mathbf{L}\Diamond(\mathcal{B}) \hookrightarrow \mathbf{L}(\mathcal{B} \cup \{\mathsf{m}\})$ based on this idea would look as follows:

$$\begin{aligned} \mathsf{p}^\flat &= \mathsf{p} \\ (\mathsf{A} \bullet \mathsf{B})^\flat &= \mathsf{A}^\flat \bullet \mathsf{B}^\flat \\ (\mathsf{C/B})^\flat &= \mathsf{C}^\flat / \mathsf{B}^\flat \end{aligned}$$

$$\begin{array}{rcl}(\mathsf{A}\backslash\mathsf{C})^\flat & = & \mathsf{A}^\flat\backslash\mathsf{C}^\flat \\ (\Diamond\mathsf{A})^\flat & = & \mathsf{m}\bullet\mathsf{A}^\flat \\ (\Box\mathsf{A})^\flat & = & \mathsf{m}\backslash\mathsf{A}^\flat\end{array}$$

However, this idea turns out to be just a little too simple. Indeed, consider the following sequent:[2]

$$\Diamond\mathsf{A}\vdash(\Diamond\mathsf{B}/\mathsf{B})\bullet\mathsf{A}$$

This sequent is easily seen to be underivable, unlike its translation:

$$\mathsf{m}\bullet\mathsf{A}^\flat\vdash(\mathsf{m}\bullet\mathsf{B}^\flat/\mathsf{B}^\flat)\bullet\mathsf{A}^\flat$$

This problem can be overcome by doubling the construction:

Theorem 5.3 *Let* $(.)^\flat : \mathbf{L}\Diamond(\mathcal{B})\hookrightarrow\mathbf{L}(\mathcal{B}\cup\{\mathsf{m},\mathsf{n}\})$ *be defined inductively as follows:*

$$\begin{array}{rcl}\mathsf{p}^\flat & = & \mathsf{p} \\ (\mathsf{A}\bullet\mathsf{B})^\flat & = & \mathsf{A}^\flat\bullet\mathsf{B}^\flat \\ (\mathsf{C}/\mathsf{B})^\flat & = & \mathsf{C}^\flat/\mathsf{B}^\flat \\ (\mathsf{A}\backslash\mathsf{C})^\flat & = & \mathsf{A}^\flat\backslash\mathsf{C}^\flat \\ (\Diamond\mathsf{A})^\flat & = & \mathsf{m}\bullet\mathsf{A}^\flat\bullet\mathsf{n} \\ (\Box\mathsf{A})^\flat & = & \mathsf{m}\backslash\mathsf{A}^\flat/\mathsf{n}\end{array}$$

Then

$$\vdash_{\mathbf{L}\Diamond(\mathcal{B})}\mathsf{A}\vdash\mathsf{B}\quad\textit{iff}\quad\vdash_{\mathbf{L}(\mathcal{B}\cup\{\mathsf{m},\mathsf{n}\})}\mathsf{A}^\flat\vdash\mathsf{B}^\flat$$

Proof The first step is to extend the mapping $(.)^\flat$ to structures (sequent antecedents) in the following way:

$$\begin{array}{rcl}(,\,,\Delta)^\flat & = & (,\,^\flat,\Delta^\flat) \\ ((,\,)^\Diamond)^\flat & = & (\mathsf{m},(,\,)^\flat,\mathsf{n})\end{array}$$

The idea of the proof is to rewrite derivations of sequents in $\mathbf{L}(\mathcal{B}\cup\{\mathsf{m},\mathsf{n}\})$ in such a way that they can be "translated back" into valid $\mathbf{L}(\mathcal{B})$ derivations. For most types and structures it is clear how they must be translated back. The problems are caused by types such as $\mathsf{m}\backslash\mathsf{A}$ and $\mathsf{A}\bullet\mathsf{n}$, since these are not in the range of $(.)^\flat$. The goal of the rewriting procedure is therefore to group together the two inference steps leading from a type A to a type $\mathsf{m}\backslash\mathsf{A}/\mathsf{n}$ or $\mathsf{m}\bullet\mathsf{A}\bullet\mathsf{n}$. The resulting subderivation can then be translated back as a single inference step in $\mathbf{L}\Diamond(\mathcal{B})$.

The rewriting procedure consists of two stages. I first discuss occurrences of types $\mathsf{m}\bullet\mathsf{A}\bullet\mathsf{n}$ on the left hand side and occurrences of types $\mathsf{m}\backslash\mathsf{A}/\mathsf{n}$ on the right hand side. The second stage is concerned with the two complementary cases.

[2] This counterexample is due to Natasha Kurtonina.

Stage one The first stage deals with occurrences of $\mathsf{m} \bullet \mathsf{A} \bullet \mathsf{n}$ on the left and occurrences of $\mathsf{m}\backslash\mathsf{A}/\mathsf{n}$ on the right. For these we make use of the reversibility of the rules $\bullet$L, $\backslash$R and /R. In other words, we use the inversion lemma proved in Section 1.2.2 and repeated below for the reader's convenience:

Lemma 5.4

$$\begin{array}{rcl} \mathsf{,\ ,A} \vdash \mathsf{B} & \textit{iff} & \mathsf{,} \vdash \mathsf{B/A} \\ \mathsf{,[A,B]} \vdash \mathsf{C} & \textit{iff} & \mathsf{,[A \bullet B]} \vdash \mathsf{C} \end{array}$$

By using this lemma we can rewrite an arbitrary derivation of a sequent $\mathsf{,}^\flat \vdash \mathsf{A}^\flat$ in such a way that occurrences of $\mathsf{m} \bullet \mathsf{A} \bullet \mathsf{n}$ on the left or of $\mathsf{m}\backslash\mathsf{A}/\mathsf{n}$ on the right are always obtained from an occurrence of A by means of two subsequent inference steps. We prove this by means of induction on the complexity of a derivation (which equals the complexity of the conclusion sequent, i.e. the number of connectives it contains).

- ▷ (Derivations of) sequents with zero complexity, i.e. axioms, are already in the desired form, so they don't need to be rewritten.
- ▷ Let the active type in the conclusion sequent be different from an occurrence of $\mathsf{m} \bullet \mathsf{A} \bullet \mathsf{n}$ on the left or of $\mathsf{m}\backslash\mathsf{A}/\mathsf{n}$ on the right. Then the derivation can be rewritten if this can be done with the derivation(s) of the premise(s), which we know is the case on the basis of the induction hypothesis.
- ▷ Suppose the active type is an occurrence of $\mathsf{m} \bullet \mathsf{A} \bullet \mathsf{n}$ on the left. Then the conclusion of the derivation is a sequent $\mathsf{,[m \bullet A \bullet n]} \vdash \mathsf{B}$. The inversion lemma now tells us that there also exists a derivation $\mathcal{D}$ of $\mathsf{,[m,A,n]} \vdash \mathsf{B}$. We can therefore replace the derivation of $\mathsf{,[m \bullet A \bullet n]} \vdash \mathsf{B}$ by the following one:

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ \vdots \\ \mathsf{,[m,A,n]} \vdash \mathsf{B}\end{array}}{\mathsf{,[m,A \bullet n]} \vdash \mathsf{B}}\,[\bullet\mathrm{L}]}{\mathsf{,[m \bullet A \bullet n]} \vdash \mathsf{B}}\,[\bullet\mathrm{L}]$$

  This derivation can be rewritten since $\mathcal{D}$ can be rewritten, which follows from the induction hypothesis.
- ▷ Finally, suppose the active type is an occurrence of $\mathsf{m}\backslash\mathsf{A}/\mathsf{n}$ on the right. Then the conclusion of the derivation is a sequent $\mathsf{,} \vdash \mathsf{m}\backslash\mathsf{A}/\mathsf{n}$. The inversion lemma now tells us that there also exists a derivation $\mathcal{D}$ of $\mathsf{m,,\ ,n} \vdash \mathsf{A}$. We can therefore replace the derivation of $\mathsf{,} \vdash \mathsf{m}\backslash\mathsf{A}/\mathsf{n}$ by the following one:

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ \vdots \\ \mathsf{m,,\ ,n} \vdash \mathsf{A}\end{array}}{\mathsf{m,,} \vdash \mathsf{A/n}}\,[/\mathrm{R}]}{\mathsf{,} \vdash \mathsf{m}\backslash\mathsf{A/n}}\,[\backslash\mathrm{R}]$$

Again, we find that this derivation can be rewritten since the induction hypothesis tells us that $\mathcal{D}$ can be rewritten.

Stage two The second step of the rewriting procedure is concerned with occurrences of $\mathsf{m} \bullet \mathsf{A} \bullet \mathsf{n}$ on the right and $\mathsf{m}\backslash\mathsf{A}/\mathsf{n}$ on the left. While further rewriting a derivation in order to ensure that such occurrences are also derived from an occurrence of type A in two subsequent inference steps, it is of course important that we make sure that the property introduced in the first stage is retained. However, it is easy to see from the definition of the rewriting steps that this is indeed the case.

The second stage of the rewriting procedure is based on the fact that if the two inference steps leading from A to $\mathsf{m} \bullet \mathsf{A} \bullet \mathsf{n}$ on the right or from A to $\mathsf{m}\backslash\mathsf{A}/\mathsf{n}$ on the left do not immediately follow each other, the lower of the two can be 'pushed up' through the derivation until it meets the other one. We prove this below.

$\mathsf{m}\backslash\mathsf{A}/\mathsf{n}$ on the left

Assume that $\mathsf{m}\backslash\mathsf{A}$ is introduced first, so that we're trying to push up an instance of the /L-rule (the other case is dual). We consider all possible rule applications we can encounter on the way.

•R The subtype $\mathsf{m}\backslash\mathsf{A}$ occurs in either the right or the left premise of the •R-rule.

$$\dfrac{\mathsf{n} \vdash \mathsf{n} \quad \dfrac{, \vdash \mathsf{S} \quad \Delta[\mathsf{m}\backslash\mathsf{A}] \vdash \mathsf{T}}{, , \Delta[\mathsf{m}\backslash\mathsf{A}] \vdash \mathsf{S} \bullet \mathsf{T}}\,[\bullet\mathrm{R}]}{, , \Delta[\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}] \vdash \mathsf{S} \bullet \mathsf{T}}\,[/\mathrm{L}] \quad \rightsquigarrow \quad \dfrac{, \vdash \mathsf{S} \quad \dfrac{\mathsf{n} \vdash \mathsf{n} \quad \Delta[\mathsf{m}\backslash\mathsf{A}] \vdash \mathsf{T}}{\Delta[\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}] \vdash \mathsf{T}}\,[/\mathrm{L}]}{, , \Delta[\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}] \vdash \mathsf{S} \bullet \mathsf{T}}\,[\bullet\mathrm{R}]$$

$$\dfrac{\mathsf{n} \vdash \mathsf{n} \quad \dfrac{, [\mathsf{m}\backslash\mathsf{A}] \vdash \mathsf{S} \quad \Delta \vdash \mathsf{T}}{, [\mathsf{m}\backslash\mathsf{A}], \Delta \vdash \mathsf{S} \bullet \mathsf{T}}\,[\bullet\mathrm{R}]}{, [\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}], \Delta \vdash \mathsf{S} \bullet \mathsf{T}}\,[/\mathrm{L}] \quad \rightsquigarrow \quad \dfrac{\dfrac{\mathsf{n} \vdash \mathsf{n} \quad , [\mathsf{m}\backslash\mathsf{A}] \vdash \mathsf{S}}{, [\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}] \vdash \mathsf{S}}\,[/\mathrm{L}] \quad \Delta \vdash \mathsf{T}}{, [\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}], \Delta \vdash \mathsf{S} \bullet \mathsf{T}}\,[\bullet\mathrm{R}]$$

•L The subtype $\mathsf{m}\backslash\mathsf{A}$ occurs in either the left part or the right part of the context surrounding the •-type.

$$\dfrac{\mathsf{m} \vdash \mathsf{m} \quad \dfrac{, , \mathsf{S}, \mathsf{T}, \Delta[\mathsf{m}\backslash\mathsf{A}] \vdash \mathsf{U}}{, , \mathsf{S} \bullet \mathsf{T}, \Delta[\mathsf{m}\backslash\mathsf{A}] \vdash \mathsf{U}}\,[\bullet\mathrm{L}]}{, , \mathsf{S} \bullet \mathsf{T}, \Delta[\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}] \vdash \mathsf{U}}\,[/\mathrm{L}] \quad \rightsquigarrow \quad \dfrac{\dfrac{\mathsf{m} \vdash \mathsf{m} \quad , , \mathsf{S}, \mathsf{T}, \Delta[\mathsf{m}\backslash\mathsf{A}] \vdash \mathsf{U}}{, , \mathsf{S}, \mathsf{T}, \Delta[\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}] \vdash \mathsf{U}}\,[/\mathrm{L}]}{, , \mathsf{S} \bullet \mathsf{T}, \Delta[\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}] \vdash \mathsf{U}}\,[\bullet\mathrm{L}]$$

$$\dfrac{\mathsf{m} \vdash \mathsf{m} \quad \dfrac{, [\mathsf{m}\backslash\mathsf{A}], \mathsf{S}, \mathsf{T}, \Delta \vdash \mathsf{U}}{, [\mathsf{m}\backslash\mathsf{A}], \mathsf{S} \bullet \mathsf{T}, \Delta \vdash \mathsf{U}}\,[\bullet\mathrm{L}]}{, [\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}], \mathsf{S} \bullet \mathsf{T}, \Delta \vdash \mathsf{U}}\,[/\mathrm{L}] \quad \rightsquigarrow \quad \dfrac{\dfrac{\mathsf{m} \vdash \mathsf{m} \quad , [\mathsf{m}\backslash\mathsf{A}], \mathsf{S}, \mathsf{T}, \Delta \vdash \mathsf{U}}{, [\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}], \mathsf{S}, \mathsf{T}, \Delta \vdash \mathsf{U}}\,[/\mathrm{L}]}{, [\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}], \mathsf{S} \bullet \mathsf{T}, \Delta \vdash \mathsf{U}}\,[\bullet\mathrm{L}]$$

/R Here there is only one possibility.

$$\dfrac{\mathsf{n} \vdash \mathsf{n} \quad \dfrac{, [\mathsf{m}\backslash\mathsf{A}], \mathsf{S} \vdash \mathsf{T}}{, [\mathsf{m}\backslash\mathsf{A}] \vdash \mathsf{T}/\mathsf{S}}\,[/\mathrm{R}]}{, [\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}] \vdash \mathsf{T}/\mathsf{S}}\,[/\mathrm{L}] \quad \rightsquigarrow \quad \dfrac{\dfrac{\mathsf{n} \vdash \mathsf{n} \quad , [\mathsf{m}\backslash\mathsf{A}], \mathsf{S} \vdash \mathsf{T}}{, [\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}], \mathsf{S} \vdash \mathsf{T}}\,[/\mathrm{L}]}{, [\mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n}] \vdash \mathsf{T}/\mathsf{S}}\,[/\mathrm{R}]$$

\R This is completely analogous to /R.

/L In this case there are three possibilities (depending on which part of the conclusion sequent the type $\mathsf{m}\backslash\mathsf{A}/\mathsf{n}$ occurs in) which are however completely analogous to each other. I illustrate one of them.

$$\dfrac{\mathsf{n}\vdash\mathsf{n}\quad\dfrac{,\,[\mathsf{m}\backslash\mathsf{A}]\vdash\mathsf{S}\quad\Delta,\mathsf{T},\Delta'\vdash\mathsf{U}}{\Delta,\mathsf{T}/\mathsf{S},,\,[\mathsf{m}\backslash\mathsf{A}],\Delta'\vdash\mathsf{U}}\,[/\mathrm{L}]}{\Delta,\mathsf{T}/\mathsf{S},,\,[\mathsf{m}\backslash\mathsf{A}/\mathsf{n},\mathsf{n}],\Delta'\vdash\mathsf{U}}\,[/\mathrm{L}]\quad\rightsquigarrow\quad\dfrac{\dfrac{\mathsf{n}\vdash\mathsf{n}\quad,\,[\mathsf{m}\backslash\mathsf{A}]\vdash\mathsf{S}}{,\,[\mathsf{m}\backslash\mathsf{A}/\mathsf{n},\mathsf{n}],\mathsf{S}\vdash\mathsf{T}}\,[/\mathrm{L}]}{\Delta,\mathsf{T}/\mathsf{S},,\,[\mathsf{m}\backslash\mathsf{A}/\mathsf{n},\mathsf{n}],\Delta'\vdash\mathsf{U}}\,[/\mathrm{L}]$$

\L This is in principle completely analogous to the previous case. The only exception occurs when in the upper \L-inference the active formula is $\mathsf{m}\backslash\mathsf{A}$. However, this means that we've arrived exactly at the situation that we were aiming for.

$\mathsf{m}\bullet\mathsf{A}\bullet\mathsf{n}$ on the right

Assume that $\mathsf{m}\bullet\mathsf{A}$ is introduced first, so that we're trying to push up an instance of the /L-rule involving n (the other case is dual). Note that for a sequent $\Delta\vdash\mathsf{n}$ in a Cut-free derivation it always holds that $\Delta=\mathsf{n}$ (this follows from an inspection of the inference rules). This means that we are dealing with instances of the following inference pattern:

$$\dfrac{,\,\vdash\mathsf{m}\bullet\mathsf{A}\quad\mathsf{n}\vdash\mathsf{n}}{,\,,\mathsf{n}\vdash\mathsf{m}\bullet\mathsf{A}\bullet\mathsf{n}}\,[\bullet\mathrm{R}]$$

We consider all possible rule applications that may have led to the sequent $,\,,\mathsf{n}\vdash\mathsf{m}\bullet\mathsf{A}\bullet\mathsf{n}$.

/R,\R Impossible.

$\bullet$R As before, we find that the left premise of this rule has to be $\mathsf{n}\vdash\mathsf{n}$, which means that we have reached our goal.

/L

$$\dfrac{\dfrac{,\,\vdash\mathsf{S}\quad\Delta[\mathsf{T}]\vdash\mathsf{m}\bullet\mathsf{A}}{\Delta[\mathsf{T}/\mathsf{S},,\,]\vdash\mathsf{m}\bullet\mathsf{A}}\,[/\mathrm{L}]\quad\mathsf{n}\vdash\mathsf{n}}{\Delta[\mathsf{T}/\mathsf{S},,\,],\mathsf{n}\vdash\mathsf{m}\bullet\mathsf{A}\bullet\mathsf{n}}\,[\bullet\mathrm{R}]\quad\rightsquigarrow\quad\dfrac{,\,\vdash\mathsf{S}\quad\dfrac{\Delta[\mathsf{T}]\vdash\mathsf{m}\bullet\mathsf{A}\quad\mathsf{n}\vdash\mathsf{n}}{\Delta[\mathsf{T}],\mathsf{n}\vdash\mathsf{m}\bullet\mathsf{A}\bullet\mathsf{n}}\,[\bullet\mathrm{R}]}{\Delta[\mathsf{T}/\mathsf{S},,\,],\mathsf{n}\vdash\mathsf{m}\bullet\mathsf{A}\bullet\mathsf{n}}\,[/\mathrm{L}]$$

\L Analogous to /L.

$\bullet$L

$$\dfrac{\dfrac{,\,[\mathsf{S},\mathsf{T}]\vdash\mathsf{m}\bullet\mathsf{A}}{,\,[\mathsf{S}\bullet\mathsf{T}]\vdash\mathsf{m}\bullet\mathsf{A}}\,[\bullet\mathrm{L}]\quad\mathsf{n}\vdash\mathsf{n}}{,\,[\mathsf{S}\bullet\mathsf{T}],\mathsf{n}\vdash\mathsf{m}\bullet\mathsf{A}\bullet\mathsf{n}}\,[\bullet\mathrm{R}]\quad\rightsquigarrow\quad\dfrac{\dfrac{,\,[\mathsf{S},\mathsf{T}]\vdash\mathsf{m}\bullet\mathsf{A}\quad\mathsf{n}\vdash\mathsf{n}}{,\,[\mathsf{S},\mathsf{T}],\mathsf{n}\vdash\mathsf{m}\bullet\mathsf{A}\bullet\mathsf{n}}\,[\bullet\mathrm{R}]}{,\,[\mathsf{S}\bullet\mathsf{T}],\mathsf{n}\vdash\mathsf{m}\bullet\mathsf{A}\bullet\mathsf{n}}\,[\bullet\mathrm{L}]$$

Now consider a derivation, the output from stage one, in which there are several occurrences of types $\mathsf{m} \bullet \mathsf{A} \bullet \mathsf{n}$ on the right and/or types $\mathsf{m} \backslash \mathsf{A} / \mathsf{n}$ on the left of which the lower inference step leading to them needs to be pushed up. Among these types, choose one of those with a highest occurrence of the first of these two inference steps. Now if we push up the matching second inference step using the procedure sketched above, the resulting subderivation is in the desired form and won't be affected by any subsequent applications of this procedure. The rest of the derivation has one less problematic occurrence. Among these, we again select one with a highest occurrence of the first of the two inference steps leading to it, and so on until the whole derivation has been brought into the desired form.

Completion of the proof A derivation that has been rewritten according to the above procedure can be translated back in much the same way as this was done for Theorem 3. From the balance invariant it follows that in most sequents occurrences on the left hand side of types m and n can always be matched and mapped back to a configuration $(.)^\diamond$ in a unique way. The only sequents for which this is not true are the ones occurring in those subderivations of which it is shown below how they translate back.

$$\dfrac{\mathsf{m} \vdash \mathsf{m} \quad \dfrac{\mathsf{n} \vdash \mathsf{n} \quad \mathsf{A} \vdash \mathsf{A}}{\mathsf{A}/\mathsf{n}, \mathsf{n} \vdash \mathsf{A}}\,[/\mathrm{L}]}{\mathsf{m}, \mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n} \vdash \mathsf{A}}\,[\backslash\mathrm{L}] \quad \rightsquigarrow \quad \dfrac{\mathsf{A} \vdash \mathsf{A}}{(\Box\mathsf{A})^\diamond \vdash \mathsf{A}}\,[\Box\mathrm{L}]$$

$$\dfrac{\mathsf{n} \vdash \mathsf{n} \quad \dfrac{\mathsf{m} \vdash \mathsf{m} \quad \mathsf{A} \vdash \mathsf{A}}{\mathsf{m}, \mathsf{m}\backslash\mathsf{A} \vdash \mathsf{A}}\,[\backslash\mathrm{L}]}{\mathsf{m}, \mathsf{m}\backslash\mathsf{A}/\mathsf{n}, \mathsf{n} \vdash \mathsf{A}}\,[/\mathrm{L}] \quad \rightsquigarrow \quad \dfrac{\mathsf{A} \vdash \mathsf{A}}{(\Box\mathsf{A})^\diamond \vdash \mathsf{A}}\,[\Box\mathrm{L}]$$

$$\dfrac{\mathsf{m} \vdash \mathsf{m} \quad \dfrac{, \vdash \mathsf{A} \quad \mathsf{n} \vdash \mathsf{n}}{, , \mathsf{n} \vdash \mathsf{A} \bullet \mathsf{n}}\,[\bullet\mathrm{R}]}{\mathsf{m}, , , \mathsf{n} \vdash \mathsf{m} \bullet \mathsf{A} \bullet \mathsf{n}}\,[\bullet\mathrm{R}] \quad \rightsquigarrow \quad \dfrac{, \vdash \mathsf{A}}{(,)^\diamond \vdash \diamond\mathsf{A}}\,[\diamond\mathrm{R}]$$

$$\dfrac{\dfrac{\mathsf{m} \vdash \mathsf{m} \quad , \vdash \mathsf{A}}{\mathsf{m}, , \vdash \mathsf{m} \bullet \mathsf{A}}\,[\bullet\mathrm{R}] \qquad \mathsf{n} \vdash \mathsf{n}}{\mathsf{m}, , , \mathsf{n} \vdash \mathsf{m} \bullet \mathsf{A} \bullet \mathsf{n}}\,[\bullet\mathrm{R}] \quad \rightsquigarrow \quad \dfrac{, \vdash \mathsf{A}}{(,)^\diamond \vdash \diamond\mathsf{A}}\,[\diamond\mathrm{R}]$$

□

A final remark: note that nothing in the proofs of Theorem 5.3 really hinges on the actual calculus used. In particular, it can easily be adapted for $\mathbf{NL}\diamond$. What is important is just that the product connective used for the translation of $\diamond$ and $\Box$ doesn't have access to any unwanted structural rules. In other words, it seems that any calculus $\mathcal{L}\diamond(\mathcal{B})$ with a product $\bullet$ for which no structural rules hold, can be faithfully embedded into $\mathcal{L}\diamond(\mathcal{B} \cup \{\mathsf{m}, \mathsf{n}\})$ with the translation of Theorem 5.3.

Chapter 6

Invariants and Conjoinability

This chapter starts with the introduction in Section 6.1 of the notions of *type equivalence* and *conjoinability*, which are shown to be equivalent to each other. Section 6.2 continues with a discussion of invariants on the derivability relation between types. These can be useful for determining the non-derivability of sequents early on. Section 6.3 concerns Pentus' proof that the invariants for **L** and **LP** completely characterize conjoinability. Finally, in Section 6.4 the question of how the invariants and conjoinability results for **L** and **LP** can be extended to other calculi is investigated.

6.1 Conjoinability

Lambek (1958) studied not only the derivability relation $\vdash$ in **L**, but also its *equivalence closure* $\equiv$, the smallest equivalence relation containing $\vdash$. In other words, $\mathsf{A} \equiv \mathsf{B}$ iff there exists a sequence $\mathsf{A} = \mathsf{C}_1, \ldots, \mathsf{C}_n = \mathsf{B}$ $(n \geq 1)$, such that $\mathsf{C}_i \vdash \mathsf{C}_{i+1}$ or $\mathsf{C}_{i+1} \vdash \mathsf{C}_i$ $(1 \leq i < n)$.

Lambek also showed that $\mathsf{A} \equiv \mathsf{B}$ is equivalent to each of the following two statements:

(6.1) $$\exists \mathsf{C} : (\mathsf{A} \vdash \mathsf{C} \quad \& \quad \mathsf{B} \vdash \mathsf{C})$$

(6.2) $$\exists \mathsf{D} : (\mathsf{D} \vdash \mathsf{A} \quad \& \quad \mathsf{D} \vdash \mathsf{B})$$

To prove this, one starts with demonstrating the equivalence of (6.1) and (6.2). First, assume that (6.1) holds. Set:

$$\mathsf{D} = (\mathsf{A}/((\mathsf{C}/\mathsf{C})\backslash\mathsf{C})) \bullet ((\mathsf{C}/\mathsf{C})\backslash\mathsf{B})$$

Then D satisfies (6.2), as is shown by the following derivations:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\mathsf{C/C} \vdash \mathsf{C/C}} \quad \mathsf{B} \vdash \mathsf{C}}{\mathsf{C/C}, (\mathsf{C/C})\backslash\mathsf{B} \vdash \mathsf{C}}\,[\backslash\mathrm{L}]}{(\mathsf{C/C})\backslash\mathsf{B} \vdash (\mathsf{C/C})\backslash\mathsf{C}}\,[\backslash\mathrm{R}] \quad \mathsf{A} \vdash \mathsf{A}}{\mathsf{A}/((\mathsf{C/C})\backslash\mathsf{C}), (\mathsf{C/C})\backslash\mathsf{B} \vdash \mathsf{A}}\,[/\mathrm{L}]}{(\mathsf{A}/((\mathsf{C/C})\backslash\mathsf{C})) \bullet ((\mathsf{C/C})\backslash\mathsf{B}) \vdash \mathsf{A}}\,[\bullet\mathrm{L}]$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{C} \vdash \mathsf{C} \quad \mathsf{C} \vdash \mathsf{C}}{\mathsf{C/C}, \mathsf{C} \vdash \mathsf{C}}\,[/\mathrm{L}]}{\mathsf{C} \vdash (\mathsf{C/C})\backslash\mathsf{C}}\,[\backslash\mathrm{R}] \quad \mathsf{A} \vdash \mathsf{C}}{\mathsf{A}/((\mathsf{C/C})\backslash\mathsf{C}), \mathsf{C} \vdash \mathsf{C}}\,[/\mathrm{L}]}{\mathsf{A}/((\mathsf{C/C})\backslash\mathsf{C}) \vdash \mathsf{C/C}}\,[/\mathrm{R}] \quad \mathsf{B} \vdash \mathsf{B}}{\mathsf{A}/((\mathsf{C/C})\backslash\mathsf{C}), (\mathsf{C/C})\backslash\mathsf{B} \vdash \mathsf{B}}\,[\backslash\mathrm{L}]}{(\mathsf{A}/((\mathsf{C/C})\backslash\mathsf{C})) \bullet ((\mathsf{C/C})\backslash\mathsf{B}) \vdash \mathsf{B}}\,[\bullet\mathrm{L}]$$

Conversely, suppose that (6.2) holds. Then C can be defined as follows:

$$\mathsf{C} = (\mathsf{A} \bullet (\mathsf{D}\backslash\mathsf{D}))/(\mathsf{B}\backslash(\mathsf{D} \bullet (\mathsf{D}\backslash\mathsf{D})))$$

The following derivations show that C thus defined satisfies (6.1):

$$\dfrac{\dfrac{\mathsf{A} \vdash \mathsf{A} \quad \dfrac{\mathsf{B} \vdash \mathsf{B} \quad \dfrac{\dfrac{\dfrac{\mathsf{D} \vdash \mathsf{D} \quad \mathsf{D} \vdash \mathsf{D}}{\mathsf{D}, \mathsf{D}\backslash\mathsf{D} \vdash \mathsf{D}}\,[\backslash\mathrm{L}]}{\mathsf{D} \bullet (\mathsf{D}\backslash\mathsf{D}) \vdash \mathsf{D}}\,[\bullet\mathrm{L}]}{\mathsf{D}, \mathsf{B}\backslash(\mathsf{D} \bullet (\mathsf{D}\backslash\mathsf{D})) \vdash \mathsf{D}}\,[\backslash\mathrm{L}]}{\mathsf{B}\backslash(\mathsf{D} \bullet (\mathsf{D}\backslash\mathsf{D})) \vdash \mathsf{D}\backslash\mathsf{D}}\,[\backslash\mathrm{R}]}{\mathsf{A}, \mathsf{B}\backslash(\mathsf{D} \bullet (\mathsf{D}\backslash\mathsf{D})) \vdash \mathsf{A} \bullet (\mathsf{D}\backslash\mathsf{D})}\,[\bullet\mathrm{R}]}{\mathsf{A} \vdash (\mathsf{A} \bullet (\mathsf{D}\backslash\mathsf{D}))/(\mathsf{B}\backslash(\mathsf{D} \bullet (\mathsf{D}\backslash\mathsf{D})))}\,[/\mathrm{R}]$$

$$\dfrac{\dfrac{\mathsf{B} \vdash \mathsf{B} \quad \dfrac{\mathsf{D} \vdash \mathsf{A} \quad \dfrac{\vdots}{\mathsf{D}\backslash\mathsf{D} \vdash \mathsf{D}\backslash\mathsf{D}}}{\mathsf{D}, \mathsf{D}\backslash\mathsf{D} \vdash \mathsf{A} \bullet (\mathsf{D}\backslash\mathsf{D})}\,[\bullet\mathrm{R}]}{\mathsf{D} \bullet (\mathsf{D}\backslash\mathsf{D}) \vdash \mathsf{A} \bullet (\mathsf{D}\backslash\mathsf{D})}\,[\bullet\mathrm{L}]}{\mathsf{B}, \mathsf{B}\backslash(\mathsf{D} \bullet (\mathsf{D}\backslash\mathsf{D})) \vdash \mathsf{A} \bullet (\mathsf{D}\backslash\mathsf{D})}\,[\backslash\mathrm{L}]}{\mathsf{B} \vdash (\mathsf{A} \bullet (\mathsf{D}\backslash\mathsf{D}))/(\mathsf{B}\backslash(\mathsf{D} \bullet (\mathsf{D}\backslash\mathsf{D})))}\,[/\mathrm{R}]$$

Pentus (1994a) showed that simpler definitions than those of Lambek can be found. Given C, we can define D as $(\mathsf{A/C}) \bullet \mathsf{C} \bullet (\mathsf{C}\backslash\mathsf{B})$, and C can be defined from D as $(\mathsf{D/A})\backslash\mathsf{D}/(\mathsf{B}\backslash\mathsf{D})$. This is illustrated by the derivations below:

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{C} \vdash \mathsf{C} \quad \dfrac{\mathsf{B} \vdash \mathsf{C} \quad \mathsf{A} \vdash \mathsf{A}}{\mathsf{A/C}, \mathsf{B} \vdash \mathsf{A}}\,[/\mathrm{L}]}{(\mathsf{A/C}), \mathsf{C}, (\mathsf{C}\backslash\mathsf{B}) \vdash \mathsf{A}}\,[\backslash\mathrm{L}]}{(\mathsf{A/C}), \mathsf{C} \bullet (\mathsf{C}\backslash\mathsf{B}) \vdash \mathsf{A}}\,[\bullet\mathrm{L}]}{(\mathsf{A/C}) \bullet \mathsf{C} \bullet (\mathsf{C}\backslash\mathsf{B}) \vdash \mathsf{A}}\,[\bullet\mathrm{L}]$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{C} \vdash \mathsf{C} \quad \dfrac{\mathsf{A} \vdash \mathsf{C} \quad \mathsf{B} \vdash \mathsf{B}}{\mathsf{A}, \mathsf{C}\backslash\mathsf{B} \vdash \mathsf{B}}\,[\backslash\mathrm{L}]}{(\mathsf{A/C}), \mathsf{C}, (\mathsf{C}\backslash\mathsf{B}) \vdash \mathsf{B}}\,[/\mathrm{L}]}{(\mathsf{A/C}) \bullet \mathsf{C}, (\mathsf{C}\backslash\mathsf{B}) \vdash \mathsf{B}}\,[\bullet\mathrm{L}]}{(\mathsf{A/C}) \bullet \mathsf{C} \bullet (\mathsf{C}\backslash\mathsf{B}) \vdash \mathsf{B}}\,[\bullet\mathrm{L}]$$

$$\dfrac{\dfrac{\dfrac{\mathsf{A}\vdash\mathsf{A}\quad\dfrac{\mathsf{D}\vdash\mathsf{B}\quad\mathsf{D}\vdash\mathsf{D}}{\mathsf{D},\mathsf{B}\backslash\mathsf{D}\vdash\mathsf{D}}\,[\backslash\mathrm{L}]}{\mathsf{D}/\mathsf{A},\mathsf{A},\mathsf{B}\backslash\mathsf{D}\vdash\mathsf{D}}\,[\backslash\mathrm{L}]}{\mathsf{D}/\mathsf{A},\mathsf{A}\vdash\mathsf{D}/(\mathsf{B}\backslash\mathsf{D})}\,[/\mathrm{R}]}{\mathsf{A}\vdash(\mathsf{D}/\mathsf{A})\backslash\mathsf{D}/(\mathsf{B}\backslash\mathsf{D})}\,[\backslash\mathrm{R}]$$

$$\dfrac{\dfrac{\dfrac{\mathsf{B}\vdash\mathsf{B}\quad\dfrac{\mathsf{D}\vdash\mathsf{A}\quad\mathsf{D}\vdash\mathsf{D}}{\mathsf{D}/\mathsf{A},\mathsf{D}\vdash\mathsf{D}}\,[/\mathrm{L}]}{\mathsf{D}/\mathsf{A},\mathsf{A},\mathsf{B}\backslash\mathsf{D}\vdash\mathsf{D}}\,[/\mathrm{L}]}{\mathsf{B},\mathsf{B}\backslash\mathsf{D}\vdash(\mathsf{D}/\mathsf{A})\backslash\mathsf{D}}\,[\backslash\mathrm{R}]}{\mathsf{B}\vdash(\mathsf{D}/\mathsf{A})\backslash\mathsf{D}/(\mathsf{B}\backslash\mathsf{D})}\,[/\mathrm{R}]$$

The advantage of Lambek's definition of the types C and D, however, is that it also works in the case of a non-associative product. Pentus' types only work under the assumption of associativity: in a non-associative setting the four sequents below are still derivable, but because of their different bracketing both pairs of types involved are no longer the same:

$$(\mathsf{A}/\mathsf{C})\bullet(\mathsf{C}\bullet(\mathsf{C}\backslash\mathsf{B}))\vdash\mathsf{A}\qquad\mathsf{A}\vdash((\mathsf{D}/\mathsf{A})\backslash\mathsf{D})/(\mathsf{B}\backslash\mathsf{D})$$
$$((\mathsf{A}/\mathsf{C})\bullet\mathsf{C})\bullet(\mathsf{C}\backslash\mathsf{B})\vdash\mathsf{A}\qquad\mathsf{A}\vdash(\mathsf{D}/\mathsf{A})\backslash(\mathsf{D}/(\mathsf{B}\backslash\mathsf{D}))$$

It still needs to be shown that $\mathsf{A}\equiv\mathsf{B}$ is equivalent to either of (6.1) and (6.2). One direction is simple, since both (6.1) and (6.2) are special instances of type equivalence. Conversely, suppose a sequence $\mathsf{C}_1,\ldots,\mathsf{C}_n$ shows the equivalence of C_1 and C_n. If $\mathsf{C}_i\vdash\mathsf{C}_{i+1}$ and $\mathsf{C}_{i+1}\vdash\mathsf{C}_{i+2}$ (or $\mathsf{C}_{i+2}\vdash\mathsf{C}_{i+1}$ and $\mathsf{C}_{i+1}\vdash\mathsf{C}_i$) for some i, then the sequence $\mathsf{C}_1,\ldots,\mathsf{C}_{i-1},\mathsf{C}_{i+1},\ldots,\mathsf{C}_n$ also shows the equivalence of C_1 and C_n. If no such i can be found, then we have sequents $\mathsf{C}_1\vdash\mathsf{C}_2$, $\mathsf{C}_3\vdash\mathsf{C}_2$ and $\mathsf{C}_3\vdash\mathsf{C}_4$.[1] By virtue of the equivalence of (6.1) and (6.2) there exists a type C_2' such that $\mathsf{C}_2'\vdash\mathsf{C}_1$ and $\mathsf{C}_2'\vdash\mathsf{C}_3$. The latter sequent and $\mathsf{C}_3\vdash\mathsf{C}_4$ together imply the derivability of the sequent $\mathsf{C}_2'\vdash\mathsf{C}_4$. From this it follows that the sequence $\mathsf{C}_1,\mathsf{C}_2',\mathsf{C}_4,\ldots,\mathsf{C}_n$ also shows the equivalence of C_1 and C_n. In other words, I have shown that any sequence of types proving the equivalence of two types can be reduced to length at most three, which must be either simple derivability or one of (6.1) and (6.2).

Types satisfying condition (6.1) are said to be *conjoinable*. This definition stems from linguistics, where conjoinability can be used in categorial accounts of conjunction and disjunction. A natural type assignment for Boolean particles such as *and* and *or* is the polymorphic type $X\backslash X/X$, where X can be any type. In order for two types A and B to be able to combine with $X\backslash X/X$, there needs to be a type C such that both $\mathsf{A}\vdash\mathsf{C}$ and $\mathsf{B}\vdash\mathsf{C}$:

$$\dfrac{\dfrac{\mathsf{A},\mathsf{C}\backslash\mathsf{C}/\mathsf{C},\mathsf{B}\vdash\mathsf{C}}{\mathsf{A},X\backslash X/X,\mathsf{B}\vdash\mathsf{C}}\,[\backslash\mathrm{L}]}{}\qquad\dfrac{\mathsf{A}\vdash\mathsf{C}\quad\dfrac{\mathsf{C}\vdash\mathsf{C}}{\mathsf{B}\vdash\mathsf{C}}\,[/\mathrm{L}]}{}$$

It turns out, however, that conjoinability is a necessary but not a sufficient condition.[2] This follows from the fact that $(\mathsf{B}/(\mathsf{A}\backslash\mathsf{B}))/\mathsf{A}$ is derivable from both A/A and B/B, which causes the following ungrammatical sentence to be accepted by the grammar:

(6.3) *I think that and a friend of Mary walks*

[1] Or the converse situation, which is dealt with analogously.

[2] The following argument is due to Paul Dekker.

The first two derivations below show that both *I think that* and *a friend of* are of type $(S/(NP\backslash S))/NP$, while the third one demonstrates that this type combines into a sentence with *Mary walks*:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{NP \vdash NP \quad S \vdash S}{NP, NP\backslash S \vdash S}}{S \vdash S \quad NP, (NP\backslash S)/S, S \vdash S}}{S \vdash S \quad \textit{I}, \textit{think}, S \vdash S}}{\textit{I}, \textit{think}, S/S, S \vdash S}}{NP \vdash NP \quad \textit{I}, \textit{think}, \textit{that}, S \vdash S}}{\dfrac{\textit{I}, \textit{think}, \textit{that}, NP, NP\backslash S \vdash S}{\dfrac{\textit{I}, \textit{think}, \textit{that}, NP \vdash S/(NP\backslash S)}{\textit{I}, \textit{think}, \textit{that} \vdash (S/(NP\backslash S))/NP}}}$$

$$\dfrac{\dfrac{\dfrac{NP \vdash NP \quad S \vdash S}{NP, NP\backslash S \vdash S}}{NP \vdash NP \quad NP \vdash S/(NP\backslash S)}}{\dfrac{NP/NP, NP \vdash S/(NP\backslash S)}{\dfrac{\textit{a friend of}, NP \vdash S/(NP\backslash S)}{\textit{a friend of} \vdash (S/(NP\backslash S))/NP}}}$$

$$\dfrac{\textit{Mary} \vdash NP \quad \dfrac{\textit{walks} \vdash NP\backslash S \quad S \vdash S}{S/(NP\backslash S), \textit{walks} \vdash S}}{(S/(NP\backslash S))/NP, \textit{Mary}, \textit{walks} \vdash S}$$

From now on I will use the more familiar term 'conjoinability' and the more usual notation $A \sim B$ instead of 'equivalence' and $A \equiv B$.

6.2 Derivability invariants

For a basic type p, define the function $\#_p$ on **LP** types as follows:

$$\begin{aligned} \#_p(p) &= 1 \\ \#_p(q) &= 0 \quad (q \neq p) \\ \#_p(A \otimes B) &= \#_p A + \#_p B \\ \#_p(A \rightarrow C) &= \#_p C \Leftrightarrow \#_p A \end{aligned}$$

Van Benthem (1986) noted that these functions are invariants of the derivability relation among types in **LP**. In other words, if $A \vdash_{\mathbf{LP}} B$, then $\#_p A = \#_p B$ for all basic types p. The proof is a straightforward induction on the length of derivations. Of course, this invariant is also valid for **L**, where its exact definition is as follows:

$$\begin{aligned} \#_p(p) &= 1 \\ \#_p(q) &= 0 \quad (q \neq p) \\ \#_p(A \bullet B) &= \#_p A + \#_p B \\ \#_p(C/B) &= \#_p C \Leftrightarrow \#_p B \\ \#_p(A\backslash C) &= \#_p C \Leftrightarrow \#_p A \end{aligned}$$

Roorda (1991) showed that the result for **L** can be strengthened by proving that if $A \vdash_{\mathbf{L}} B$, then $[\![A]\!] = [\![B]\!]$, where the equality is in the free group generated

by the basic types, and $[\![.]\!]$ is defined as follows:

$$\begin{aligned} [\![\mathsf{p}]\!] &= \mathsf{p} \\ [\![\mathsf{A} \bullet \mathsf{B}]\!] &= [\![\mathsf{A}]\!] \cdot [\![\mathsf{B}]\!] \\ [\![\mathsf{C}/\mathsf{A}]\!] &= [\![\mathsf{C}]\!] \cdot [\![\mathsf{A}]\!]^{-1} \\ [\![\mathsf{A}\backslash\mathsf{C}]\!] &= [\![\mathsf{A}]\!]^{-1} \cdot [\![\mathsf{C}]\!] \end{aligned}$$

Note that the count invariant can alternatively be characterized in similar terms, namely by interpreting types in the free *abelian* group.

Since equality of invariants that are expressed as unary functions on types is an equivalence relation, any derivability invariant is also a conjoinability invariant.

6.3 Deciding conjoinability

Pentus (1994a) proved the converse of the last observation. That is, equality with respect to the invariants given above is not only a necessary, but also a sufficient condition for two types to be conjoinable. An easy corollary to this result is the fact that the above derivability invariants are the strongest ones possible. I will briefly outline Pentus' proof below.

The basis of the proof is the observation that the equivalence classes of the conjoinability relation can be turned into a group by the following definitions:

$$\begin{aligned} [\mathsf{A}]_\sim \cdot [\mathsf{B}]_\sim &= [\mathsf{A} \bullet \mathsf{B}]_\sim \\ \mathbf{1}_\sim &= [\mathsf{A}/\mathsf{A}]_\sim \\ [\mathsf{A}]_\sim^{-1} &= [\mathsf{A}\backslash\mathsf{A}/\mathsf{A}]_\sim \end{aligned}$$

Now consider the mapping h which maps basic types to their conjoinability classes, i.e. $h : \mathsf{p} \mapsto [\mathsf{p}]_\sim$. It turns out that the homomorphic extension of h to the free group generated by the basic types maps the interpretation of any type to the conjoinability class of that type. Now if $[\![\mathsf{A}]\!] = [\![\mathsf{B}]\!]$, then $h([\![\mathsf{A}]\!]) = h([\![\mathsf{B}]\!])$. Since $h([\![\mathsf{A}]\!]) = [\mathsf{A}]_\sim$ and $h([\![\mathsf{B}]\!]) = [\mathsf{B}]_\sim$, this implies that $[\mathsf{A}]_\sim = [\mathsf{B}]_\sim$, i.e. $\mathsf{A} \sim \mathsf{B}$.

6.4 Extended calculi

In this section I will investigate invariants and conjoinability in other calculi besides **L** and **LP**. For concreteness' sake I will initially limit myself to the eight systems in Figure 5.1 and their extensions with a pair of unary residuals $\Diamond$ and $\Box$. Actually, I won't have very much to say about the eight systems which lack the rule [D]. The reason for this is that what goes on there is basically just twice what happens in their non dependency sensitive counterparts. Therefore, the discussion of these systems is deferred until Section 6.4.3. This leaves a total of six calculi besides **L** and **LP**, namely these: **NL**, **NLP**, **NL**$\Diamond$, **NLP**$\Diamond$, **L**$\Diamond$, and **LP**$\Diamond$.

In Section 6.4.1, I will discuss invariants defined for some of these systems by Moortgat (1994). Moreover, I show how such invariants can also be obtained by using the embedding theorems of the last chapter. It is not clear if these invariants are also sufficient as a characterization of conjoinability, although my conjecture is that this is actually the case.

In Section 6.4.2.2 the decidability of conjoinability for $\mathbf{L}\Diamond$ and $\mathbf{LP}\Diamond$ will be shown to follow easily from a straightforward extension of Pentus' proof. For the other four cases a different strategy is needed, since the lack of associativity means that Pentus' proof method is not available. Instead, I will show in Section 6.4.2.3 that conjoinability can be characterized by a set of equations for which a complete term rewriting system exists, which entails decidability.

Finally, in Section 6.4.3 the issue of how these results can be further generalized will be addressed.

6.4.1 Invariants

Moortgat (1994) gives several strengthenings of Roorda's balance invariant for **NL**. They are based on the fact that in **NL** what matters is not only the linear order of resources, but also their hierarchical structuring. I will discuss two of Moortgat's invariants, which are based on different encodings of partial information about the hierarchical position of subtypes.[3]

Moortgat's first invariant is based on the fact that cancelling two literals p and p^{-1} against each other is only valid from an **NL** point of view when the nesting depths of the two corresponding atomic types have different polarity. Interpretation is therefore done in the free group generated by the set $\{\mathsf{p}, \overline{\mathsf{p}}\}$ of basic types marked for polarity. The polarity flipping function defined on types in the obvious way has the expected properties: $\overline{\overline{\mathsf{A}}} = \mathsf{A}$ and $\overline{\mathsf{A}^{-1}} = \overline{\mathsf{A}}^{-1}$ for all types A. Writing $(.)^{\perp}$ for this latter combined function, the invariant is defined as follows:

$$\begin{array}{rcccl} [\![\mathsf{A} \bullet \mathsf{B}]\!] & = & \overline{[\![\mathsf{A}]\!] \cdot [\![\mathsf{B}]\!]} & = & \overline{[\![\mathsf{A}]\!]} \cdot \overline{[\![\mathsf{B}]\!]} \\ [\![\mathsf{C}/\mathsf{B}]\!] & = & \overline{[\![\mathsf{C}]\!] \cdot [\![\mathsf{B}]\!]^{-1}} & = & \overline{[\![\mathsf{C}]\!]} \cdot [\![\mathsf{B}]\!]^{\perp} \\ [\![\mathsf{A}\backslash\mathsf{C}]\!] & = & \overline{[\![\mathsf{A}]\!]^{-1} \cdot [\![\mathsf{C}]\!]} & = & [\![\mathsf{A}]\!]^{\perp} \cdot \overline{[\![\mathsf{C}]\!]} \\ [\![\mathsf{p}]\!] & = & \mathsf{p} & & \end{array}$$

Moortgat's second invariant encodes structural information more directly by adding a *structure marker* $\wr$ and its inverse in the appropriate places. This invariant is related to the embedding of **NL** in $\mathbf{L}\Diamond$ discussed in Section 5.2.2 (more about this in the next section). Interpretation in this case is in the free group generated by the set containing the basic types and $\wr$.

$$\begin{array}{rcl} [\![\mathsf{A} \bullet \mathsf{B}]\!] & = & \wr \cdot [\![\mathsf{A}]\!] \cdot [\![\mathsf{B}]\!] \\ [\![\mathsf{C}/\mathsf{B}]\!] & = & \wr^{-1} \cdot [\![\mathsf{C}]\!] \cdot [\![\mathsf{B}]\!]^{-1} \\ [\![\mathsf{A}\backslash\mathsf{C}]\!] & = & [\![\mathsf{A}]\!]^{-1} \cdot \wr^{-1} \cdot [\![\mathsf{C}]\!] \\ [\![\mathsf{p}]\!] & = & \mathsf{p} \end{array}$$

[3] My definitions are somewhat different from those of Moortgat's, but equivalent to them.

Now consider the following three sequents:

$$q \vdash p/(q\backslash p)$$
$$p \bullet (q \bullet r) \vdash (p \bullet q) \bullet r$$
$$p/q \vdash (p/r)/(q/r)$$

The first of these is derivable in both **L** and **NL**, while the other two are typical non-**NL**-derivable sequents. The invariants are able to discern this: for both of them we find that:

$$\begin{aligned} [\![q]\!] &= [\![p/(q\backslash p)]\!] \\ [\![p \bullet (q \bullet r)]\!] &\neq [\![(p \bullet q) \bullet r]\!] \\ [\![p/q]\!] &\neq [\![(p/r)/(q/r)]\!] \end{aligned}$$

I start by calculating the value of the first invariant for the types involved:

$$\begin{aligned} &[\![p/(q\backslash p)]\!] \\ &= \overline{[\![p]\!]} \cdot [\![q\backslash p]\!]^{\perp} \\ &= \overline{[\![p]\!]} \cdot ([\![q]\!]^{\perp} \cdot \overline{[\![p]\!]})^{\perp} \\ &= \overline{[\![p]\!]} \cdot \overline{[\![p]\!]}^{\perp} \cdot [\![q]\!]^{\perp\perp} \\ &= [\![q]\!]^{\perp\perp} \\ &= [\![q]\!] \end{aligned}$$

$$\begin{aligned} &[\![p \bullet (q \bullet r)]\!] \\ &= \overline{[\![p]\!]} \cdot \overline{[\![q \bullet r]\!]} \\ &= \overline{[\![p]\!]} \cdot \overline{\overline{[\![q]\!]} \cdot \overline{[\![r]\!]}} \\ &= \overline{[\![p]\!]} \cdot [\![q]\!] \cdot [\![r]\!] \\ &= \overline{p} \cdot q \cdot r \end{aligned}$$

$$\begin{aligned} &[\![(p \bullet q) \bullet r]\!] \\ &= \overline{[\![p \bullet q]\!]} \cdot \overline{[\![r]\!]} \\ &= \overline{\overline{[\![p]\!]} \cdot \overline{[\![q]\!]}} \cdot \overline{[\![r]\!]} \\ &= [\![p]\!] \cdot [\![q]\!] \cdot \overline{[\![r]\!]} \\ &= p \cdot q \cdot \overline{r} \end{aligned}$$

$$\begin{aligned} &[\![p/q]\!] \\ &= [\![p]\!] \cdot [\![q]\!]^{\perp} \\ &= \overline{p} \cdot \overline{q}^{-1} \end{aligned}$$

$$\begin{aligned} &[\![(p/r)/(q/r)]\!] \\ &= \overline{[\![p/r]\!]} \cdot \overline{[\![q/r]\!]}^{\perp} \\ &= \overline{\overline{[\![p]\!]} \cdot [\![r]\!]^{\perp}} \cdot (\overline{[\![q]\!]} \cdot [\![r]\!]^{\perp})^{\perp} \\ &= [\![p]\!] \cdot [\![r]\!]^{-1} \cdot [\![r]\!] \cdot [\![q]\!]^{-1} \\ &= p \cdot q^{-1} \end{aligned}$$

For the second invariant the values are as follows:

$$\begin{aligned} &[\![p/(q\backslash p)]\!] \\ &= \imath^{-1} \cdot [\![p]\!] \cdot [\![q\backslash p]\!]^{-1} \\ &= \imath^{-1} \cdot [\![p]\!] \cdot ([\![q]\!]^{-1} \cdot \imath^{-1} \cdot [\![p]\!])^{-1} \\ &= \imath^{-1} \cdot [\![p]\!] \cdot [\![p]\!]^{-1} \cdot \imath \cdot [\![q]\!] \\ &= [\![q]\!] \end{aligned}$$

$$\begin{aligned} &[\![p \bullet (q \bullet r)]\!] \\ &= \imath \cdot [\![p]\!] \cdot [\![q \bullet r]\!] \\ &= \imath \cdot [\![p]\!] \cdot \imath \cdot [\![q]\!] \cdot [\![r]\!] \\ &= \imath \cdot p \cdot \imath \cdot q \cdot r \end{aligned}$$

$$\begin{aligned} &[\![(p \bullet q) \bullet r]\!] \\ &= \imath \cdot [\![p \bullet r]\!] \cdot [\![q]\!] \\ &= \imath \cdot \imath \cdot [\![p]\!] \cdot [\![q]\!] \cdot [\![r]\!] \\ &= \imath^2 \cdot p \cdot q \cdot r \end{aligned}$$

$$\begin{array}{ll}
[\![p/q]\!] & [\![(p/r)/(q/r)]\!] \\
\quad = \iota^{-1} \cdot [\![p]\!] \cdot [\![q]\!]^{-1} & \quad = \iota^{-1} \cdot [\![p/r]\!] \cdot [\![q/r]\!]^{-1} \\
\quad = \iota^{-1} \cdot p \cdot q^{-1} & \quad = \iota^{-1} \cdot \iota^{-1} \cdot [\![p]\!] \cdot [\![r]\!]^{-1} \cdot (\iota^{-1} \cdot [\![q]\!] \cdot [\![r]\!]^{-1})^{-1} \\
 & \quad = \iota^{-2} \cdot p \cdot r^{-1} \cdot r \cdot q^{-1} \cdot \iota \\
 & \quad = \iota^{-2} \cdot p \cdot q^{-1} \cdot \iota
\end{array}$$

On closer inspection, though, the second invariant turns out to be stronger than the first one. This can be seen from their values on the types $p \bullet (q \backslash r)$ and $p/(r \backslash q)$. For the first invariant, these are equal:

$$\begin{array}{ll}
[\![p \bullet (q \backslash r)]\!] & [\![p/(r \backslash q)]\!] \\
\quad = \overline{[\![p]\!]} \cdot \overline{[\![q \backslash r]\!]} & \quad = \overline{[\![p]\!]} \cdot [\![r \backslash q]\!]^{\perp} \\
\quad = \overline{p} \cdot \overline{q^{\perp} \cdot \overline{r}} & \quad = \overline{p} \cdot (r^{\perp} \cdot \overline{q})^{\perp} \\
\quad = \overline{p} \cdot \overline{q^{\perp}} \cdot \overline{\overline{r}} & \quad = \overline{p} \cdot \overline{q}^{\perp} \cdot r^{\perp^{\perp}} \\
\quad = \overline{p} \cdot q^{-1} \cdot r & \quad = \overline{p} \cdot q^{-1} \cdot r
\end{array}$$

As the second invariant shows, however, both types nonetheless aren't conjoinable:

$$\begin{array}{ll}
[\![p \bullet (q \backslash r)]\!] & [\![p/(r \backslash q)]\!] \\
\quad = \iota^{-1} \cdot [\![p]\!] \cdot [\![q \backslash r]\!] & \quad = \iota^{-1} \cdot [\![p]\!] \cdot [\![r \backslash q]\!]^{-1} \\
\quad = \iota^{-1} \cdot [\![p]\!] \cdot [\![q]\!]^{-1} \cdot \iota^{-1} \cdot [\![r]\!]^{-1} & \quad = \iota^{-1} \cdot [\![p]\!] \cdot ([\![r]\!]^{-1} \cdot \iota^{-1} \cdot [\![q]\!])^{-1} \\
\quad = \iota^{-1} \cdot p \cdot q^{-1} \cdot \iota^{-1} \cdot r & \quad = \iota^{-1} \cdot p \cdot q^{-1} \cdot \iota \cdot r
\end{array}$$

Moortgat notes that the first of the two above invariants can also be used for **NLP** by switching to interpretation in a free abelian group. Finally, he shows how to extend Roorda's balance and the second invariant to ones for $\mathbf{L}\Diamond$ and $\mathbf{NL}\Diamond$, respectively. I won't discuss these invariants in detail here, since I will introduce stronger ones in the next section.

6.4.1.1 Using embeddings

Consider an embedding $(.)^\flat : \mathcal{L}_0 \to \mathcal{L}_1$ such that $A \vdash_{\mathcal{L}_0} B$ iff $A^\flat \vdash_{\mathcal{L}_1} B^\flat$. Suppose that for $\mathcal{L}_1$ there is a derivability invariant $[\![.]\!]_1$, i.e. if $A \vdash_{\mathcal{L}_1} B$ then $[\![A]\!]_1 = [\![B]\!]_1$. Clearly now $[\![.]\!]_0$ defined by $[\![A]\!]_0 = [\![A^\flat]\!]_1$ is an invariant for $\mathcal{L}_0$. This means that the embeddings proved in the last chapter and those from Kurtonina and Moortgat (1995) can be used to derive derivability invariants for several calculi. Below I explicitly give two of these invariants, namely the ones for **NL** and $\mathbf{L}\Diamond$.

Proposition 6.1 *Let* $[\![.]\!] : \mathbf{NL}(\mathcal{B}) \to \mathcal{FG}(\mathcal{B} \cup \{m, n\})$ *be defined inductively as follows:*

$$\begin{array}{rcl}
[\![p]\!] & = & p \\
[\![A \bullet B]\!] & = & m \cdot [\![A]\!] \cdot [\![B]\!] \cdot n \\
[\![C/B]\!] & = & m^{-1} \cdot [\![C]\!] \cdot n^{-1} \cdot [\![B]\!]^{-1} \\
[\![A \backslash C]\!] & = & [\![A]\!]^{-1} \cdot m^{-1} \cdot [\![C]\!] \cdot n^{-1}
\end{array}$$

Then A *and* B *are conjoinable in* $\mathbf{NL}(\mathcal{B})$ *iff* $[\![\mathsf{A}]\!] = [\![\mathsf{B}]\!]$.

Proposition 6.2 *Let* $[\![.]\!] : \mathbf{L}\Diamond(\mathcal{B}) \to \mathcal{FG}(\mathcal{B} \cup \{\mathsf{m}, \mathsf{n}\})$ *be defined inductively as follows:*

$$\begin{array}{rcl} [\![\mathsf{p}]\!] & = & \mathsf{p} \\ [\![\mathsf{A} \bullet \mathsf{B}]\!] & = & [\![\mathsf{A}]\!] \cdot [\![\mathsf{B}]\!] \\ [\![\mathsf{C}/\mathsf{B}]\!] & = & [\![\mathsf{C}]\!] \cdot [\![\mathsf{B}]\!]^{-1} \\ [\![\mathsf{A}\backslash\mathsf{C}]\!] & = & [\![\mathsf{A}]\!]^{-1} \cdot [\![\mathsf{C}]\!] \\ [\![\Diamond\mathsf{A}]\!] & = & \mathsf{m} \cdot [\![\mathsf{A}]\!] \cdot \mathsf{n} \\ [\![\Box\mathsf{A}]\!] & = & \mathsf{m}^{-1} \cdot [\![\mathsf{A}]\!] \cdot \mathsf{n}^{-1} \end{array}$$

Then A *and* B *are conjoinable in* $\mathbf{L}\Diamond(\mathcal{B})$ *iff* $[\![\mathsf{A}]\!] = [\![\mathsf{B}]\!]$.

The reader will have noted the similarity between the invariant thus obtained for **NL** and the one defined by Moortgat that I discussed in the last section. This is no coincidence, since the intuitions underlying the approach to invariants by way of embeddings also guided Moortgat's definition. An interesting question is now whether his simpler invariant is as strong as the one from Proposition 6.1. This turns out not to be the case, as is shown by a calculation of the value of the invariants for types such as $\mathsf{p}\backslash\mathsf{p}$ and p/p. The invariant of Proposition 6.1 has a different value on all these types, showing that none of them are conjoinable with one another:

$$\begin{array}{rcl} [\![\mathsf{p}\backslash\mathsf{p}]\!] & = & \mathsf{p}^{-1} \cdot \mathsf{m}^{-1} \cdot \mathsf{p} \cdot \mathsf{n}^{-1} \\ [\![\mathsf{p}/\mathsf{p}]\!] & = & \mathsf{m}^{-1} \cdot \mathsf{p} \cdot \mathsf{n}^{-1} \cdot \mathsf{p}^{-1} \end{array}$$

For types of the form $\mathsf{p}\backslash\mathsf{p}$ Moortgat's invariant can make this distinction as well:

$$[\![\mathsf{p}\backslash\mathsf{p}]\!] = \mathsf{p}^{-1} \cdot \imath^{-1} \cdot \mathsf{p} \neq \mathsf{q}^{-1} \cdot \imath^{-1} \cdot \mathsf{q} = [\![\mathsf{q}\backslash\mathsf{q}]\!]$$

However, it can't discern between types of the form p/p:

$$[\![\mathsf{p}/\mathsf{p}]\!] = \imath^{-1} \cdot \mathsf{p} \cdot \mathsf{p}^{-1} = \imath^{-1} = \imath^{-1} \cdot \mathsf{q} \cdot \mathsf{q}^{-1} = [\![\mathsf{q}/\mathsf{q}]\!]$$

Of course, the symmetric dual of the invariant, although able to make this latter distinction, would be equal on all types of the form $\mathsf{p}\backslash\mathsf{p}$.

6.4.2 Deciding conjoinability

In this section I will show that the question whether two types are conjoinable is decidable for all calculi under consideration. I have not been able to extend Pentus' result to the invariants of Propositions 6.1 and 6.2, although my conjecture is that they do in fact completely characterize conjoinability. It is possible to prove conjoinability by means of a straightforward extension of Pentus' proof for $\mathbf{L}\Diamond$ and $\mathbf{LP}\Diamond$, and this is the subject of Section 6.4.2.2. In the non-associative cases Pentus' proof no longer goes through. However, as I show in Section 6.4.2.3, a more direct solution turns out to be available in that case. In both cases I use term rewriting systems for the proofs. These are introduced first in Section 6.4.2.1.

6.4.2.1 Term rewriting systems

In the next two sections, conjoinability for several calculi will be characterized in terms of sets of equations (cf. Section 4.4). In order to prove that equality under these sets of equations is decidable I will make use of term rewriting systems. These are introduced below. My presentation is necessarily terse; more can be found in e.g. Huet and Oppen (1980).

Term rewriting systems (TRS) are sets of equations in which all the equations are *directed*, leading to a notion of *reduction* of terms. A one-step reduction $s \to s'$ of a term s to a term s' consists of replacing a subterm t^σ of s by a term u^σ, where σ is a substitution and $t \to u$ a directed equation of the TRS. The reduction relation $s \to^* t$ defined by a TRS is the reflexive and transitive closure of the one-step reduction relation. A *normal form* of a TRS is a term that can not be reduced.

For deciding equivalence of terms under a set of equations, the following two properties of TRS'es are important:

Definition 6.3 *A TRS is called* strongly normalizing *(SN) iff for no term t there is an infinite sequence of one-step reductions $t = t_0 \to t_1 \to t_2 \to \ldots$.*

Definition 6.4 *A TRS is called* Church-Rosser *(CR) iff whenever $s \to^* t$ and $s \to^* t'$, there exists a u such that $t \to^* u$ and $t' \to^* u$.*

A TRS that is both SN and CR is said to be *complete*. The combination of these two properties implies that in a complete TRS all terms have a unique normal form. This in turn means that equality of terms can be decided by reducing them to their normal forms and checking whether or not these are equal.

Since we will be considering calculi built up from several largely independent parts (e.g. products and modalities), the following results will be useful. In these theorems, $\mathcal{R}_1 \oplus \mathcal{R}_2$ denotes the disjoint sum of $\mathcal{R}_1$ and $\mathcal{R}_2$, and duplicating rules are rules in which a variable x occurs more often on the right hand side than on the left hand side.

Theorem 6.5 (Toyama (1987)) *$\mathcal{R}_1 \oplus \mathcal{R}_2$ is CR iff both $\mathcal{R}_1$ and $\mathcal{R}_2$ are CR.*

Theorem 6.6 (Rusinowitch (1987)) *If $\mathcal{R}_1$ and $\mathcal{R}_2$ are SN, and neither $\mathcal{R}_1$ nor $\mathcal{R}_2$ contains duplicating rules, then $\mathcal{R}_1 \oplus \mathcal{R}_2$ is also SN.*

What I will actually use is the following corollary to these theorems:

Corollary 6.7 *If $\mathcal{R}_1$ and $\mathcal{R}_2$ are complete and neither $\mathcal{R}_1$ nor $\mathcal{R}_2$ contains duplicating rules, then $\mathcal{R}_1 \oplus \mathcal{R}_2$ is also complete.*

For a TRS that isn't complete it is often possible to construct an equivalent complete TRS. A well-known method for finding such an equivalent complete TRS is the Knuth-Bendix critical pair completion algorithm. Basically what this procedure does is adding extra rewrite rules for *critical pairs*: pairs of terms that are the same under the set of equations but do not yet have a common reduct. Completeness results mentioned below without proof were either obtained or checked with the help of the Larch theorem prover.[4]

[4] See `http://larch-www.lcs.mit.edu:8001/larch/`.

6.4.2.2 The associative case: L◇ and LP◇

For **L**◇ and **LP**◇, the interpretation algebra is again the free group generated by the basic types, but now with two additional unary operators ◇ and □ added. The behaviour of these operators is defined by the following equations:

$$\begin{aligned} \Diamond(\Box(x)) &= x \\ \Box(\Diamond(x)) &= x \end{aligned}$$

The alternative characterization of unary residuation that I gave in Section 5.1.3 demonstrates exactly that the following functions ◇ and □ on the group of conjoinability classes are well-defined and satisfy the above equations:

$$\begin{aligned} \Diamond([\mathsf{A}]_\sim) &= [\Diamond\mathsf{A}]_\sim \\ \Box([\mathsf{A}]_\sim) &= [\Box\mathsf{A}]_\sim \end{aligned}$$

All that is left to prove now is that equality of terms in this new algebra is decidable, but that is straightforward. One way of doing this is by using TRS'es, which I will illustrate for the case of **L**◇.

Consider the group equations:

$$\begin{aligned} (x \cdot y) \cdot z &= x \cdot (y \cdot z) \\ x \cdot 1 &= x \\ 1 \cdot x &= x \\ x \cdot x^{-1} &= 1 \\ x^{-1} \cdot x &= 1 \end{aligned}$$

The following is a complete TRS for this system of equations:

$$\begin{array}{rcl@{\qquad}rcl} (x \cdot y) \cdot z & \to & x \cdot (y \cdot z) & 1^{-1} & \to & 1 \\ x \cdot 1 & \to & x & x \cdot (x^{-1} \cdot y) & \to & y \\ 1 \cdot x & \to & x & x^{-1} \cdot (x \cdot y) & \to & y \\ x \cdot x^{-1} & \to & 1 & (x^{-1})^{-1} & \to & x \\ x^{-1} \cdot x & \to & 1 & (x \cdot y)^{-1} & \to & y^{-1} \cdot x^{-1} \end{array}$$

The equations for ◇ and □ lead to the following reductions rules:

$$\begin{aligned} \Diamond(\Box(x)) &\to x \\ \Box(\Diamond(x)) &\to x \end{aligned}$$

This last simple TRS is easily shown to be complete. Since neither TRS contains any duplicating rules, Corollary 6.7 now immediately yields the completeness of their direct sum, and hence the decidability of the invariant for **L**◇.

6.4.2.3 Non-associative systems

For non-associative calculi Pentus' proof does not work. Luckily, these systems are so simple that the conjoinability of two types can actually be decided more

directly. There is no need for an interpretation algebra, since the set of equations defining conjoinability of types in each case gives rise to a TRS that can easily be completed. This set of equations for conjoinability are obtained from the alternative characterization of residuation given in Section 5.1.2. For the non-associative product, these equations are:

$$\begin{aligned} x \bullet (x\backslash y) &= y \\ y &= x\backslash(x \bullet y) \\ (y/x) \bullet x &= y \\ y &= (y \bullet x)/x \end{aligned}$$

It is clear how the equations should be directed in order to turn this into a suitable TRS, since in all of the equations one side is a proper subterm of the other side. I.e. we get the following TRS:

$$\begin{aligned} x \bullet (x\backslash y) &\to y \\ x\backslash(x \bullet y) &\to y \\ (y/x) \bullet x &\to y \\ (y \bullet x)/x &\to y \end{aligned}$$

Only two rules need to be added to this TRS to make it into a complete one, namely the following ones:

$$\begin{aligned} x/(y\backslash x) &\to y \\ (x/y)\backslash x &\to y \end{aligned}$$

Again, the addition of unary modalities presents no problems whatsoever. This is because we find the same pair of equations as before, and again end up with two complete TRS'es without duplicating rules.

For **NLP** and **NLP**$\Diamond$ things are somewhat more complicated due to the fact that turning the commutativity equation into a rewrite rule doesn't lead to a complete TRS; for instance, for any two constants a and b there is an infinite reductions sequence $a \bullet b \to b \bullet a \to a \bullet b \to \ldots$. There is a well-known solution for this problem, however, which is rewriting *modulo* commutativity. Essentially what this means is that rewriting is not done on single equations, but instead on congruence classes modulo commutativity. This solves the above problem, since $a \bullet b \to b \bullet a$ is no longer a proper reduction under this definition because both terms represent the same congruence class. I will not go into the details of rewriting modulo equations here. The interested reader is referred to Peterson and Stickel (1981). It turns out that under the asumption of rewriting modulo commutativity the TRS'es given before for **NL** and **NL**$\Diamond$ are complete characterizations of conjoinability for **NLP** and **NLP**$\Diamond$, respectively.

6.4.3 Extending the results

In this final section I will advance some speculations about the extensibility of the results of the previous section. Two equational characterizations of conjoinability are available:

1. Equations between types
2. Equations between elements in free groups

The second characterization requires that the product be associative.

The possible extensions to the calculi considered can be divided thus:

1. Additional structural rules for product or modalities
2. Multimodal systems
   (a) without interaction postulates
   (b) with interaction postulates

I discuss these different kinds of extensions below:

1. As we have seen, it is not in general possible to add structural rules for connectives to the conjoinability equations on types. What does work sometimes is rewriting modulo the structural rules. The question is when this results in a TRS that can be completed. It seems unlikely that this question could be answered in general, but it may be possible to obtain partial answers from the relevant literature, e.g. Jouannaud and Kirchner (1986).

   For the free group interpretation no general result seems easily obtainable either. However, suppose that the structural rules have the Gautam property (cf. Section 4.4). Then terms equal under the equations introduced by these rules are always built up from the same generators. This might imply decidability of the the word problem, which would mean that conjoinability would be decidable.

2. (a) Both characterizations of conjoinability have no problems dealing with disjoint unions of calculi. For the first one this holds because all TRS'es involved are free of duplicating rules. For the second one, the proofs for the separate parts do not interfere with each other, and therefore together constitute a proof for the combined system.
   (b) The first characterization does not easily handle multimodal calculi with interaction postulates. The reason is that the theorems about combinations of TRS'es all assume that we are dealing with a disjoint union, which can never be the case when there interaction postulates. Therefore, no general result can be stated for this case on the basis of those theorems.

      For the free group characterization similar arguments to those advanced in the previous case obtain. That is, interaction postulates observing the Gautam property might leave decidability of conjoinability unaffected as long as all products involved are associative.

Finally, it should be mentioned that combinations of both approaches may also give interesting results. For instance, it seems likely that a multimodal calculus consisting of an associative and a non-associative product could be dealt with by interpreting in an algebra which has $\cdot$ and $(.)^{-1}$ for the associative product, and $\bullet$, $\backslash$ and $/$ for the non-associative one.

Part IV

Word order domains and verb raising

The previous chapters were strongly oriented towards the logical properties of extensions to Lambek categorial grammars that have been proposed for dealing with problems that arise from discontinuity phenomena in natural language. In this part I will present a linguistic illustration of such systems. The phenomenon that will be considered is that of *verb-raising* in Dutch.

I start in the first section of Chapter 7 with a review of the pertinent data. This is followed in Section 7.2.1 by a discussion of accounts for Verb-raising in some present theories of grammar other than CG. Previous categorial accounts of Verb-raising are the subject of Section 7.2.2.

In Chapter 8 it is shown how one particular account of Verb raising, that of Reape discussed in Section 7.2.1.4, can be rephrased in terms of a multimodal categorial grammar.

Chapter 7

Verb-raising in Dutch

In this chapter I will discuss a construction found in Dutch (and other West-Germanic languages) with verbs subcategorizing for verbal complements. This construction is commonly referred to as *verb-raising*, after the name of the rule introduced by Evers (1975) to account for it.

I start in Section 7.1 with a review of the pertinent data, focussing on verb-raising in Dutch. In Section 7.2, I briefly discuss a number of accounts of verb-raising that can be found in the literature. Non-categorial accounts are the subject of Section 7.2.1, the discussion of Reape's theory of word order domains in Section 7.2.1.4 being of particular relevance for the next chapter. Section 7.2.2 features the main categorial proposals.

7.1 Verb-raising data

In Dutch, verbs selecting a verbal complement can give rise to a construction illustrated by sentences (1b), (1d) and (1e) below, in which a verb and its argument noun phrase are separated from one another by other verbs and/or noun phrases involved in the construction.

(1)a. *dat* *de kraanvogels* *vliegen*
that the cranes fly
'that the cranes fly'

(1)b. *dat* *Cecilia* *de kraanvogels* *ziet* *vliegen*
that Cecilia the cranes sees fly
'that Cecilia sees the cranes fly'

(1)c. *dat* *Cecilia* *de nijlpaarden* *voert*
that Cecilia the hippos feeds
'that Cecilia feeds the hippos'

d. *dat* *Henk* *Cecilia* *de nijlpaarden* *helpt* *voeren*
that Henk Cecilia the hippos helps feed

'that Henk helps Cecilia feed the hippos'

e. *dat ik Henk Cecilia de nijlpaarden zie helpen voeren*
that I Henk Cecilia the hippos see help feed
'that I see Henk help Cecilia feed the hippos'

This construction is commonly referred to as *verb-raising*, after the rule proposed to account for it by Evers (1975), which will be discussed in Section 7.2.1.1.

The *crossed dependencies* in (1) are not the only possible word order, since there is also an *extraposition* construction which causes verbal complements to occur to the right of their governor:

(2)a. *dat Cecilia afspreekt de nijlpaarden te voeren*
that Cecilia agrees the hippos to feed
'that Cecilia agrees to feed the hippos'

b. *dat Cecilia Henk belooft de nijlpaarden te voeren*
that Cecilia Henk promises the hippos to feed
'that Cecilia promises Henk to feed the hippos'

Both constructions are independent of each other in the sense that some verbs allow only verb-raising, others only extraposition, while still others allow both constructions.[1]

Although the linear order of the verbs is pretty much fixed, there is one important exception, illustrated in (3) below. It concerns the head of the verbal complement of auxiliaries and finite modals, which may optionally be ordered at the beginning of the verb cluster. For modals, this inversion is only possible if the verbal head of the argument doesn't itself subcategorize for yet another verb.

(3)a. *dat Cecilia de nijlpaarden heeft gevoerd*
that Cecilia the hippos has fed
'that Cecilia has fed the hippos'

b. *dat Cecilia de nijlpaarden gevoerd heeft*

c. *dat Cecilia de nijlpaarden moet hebben gevoerd*
that Cecilia the hippos must have fed
'that Cecilia must have fed the hippos'

d. *dat Cecilia de nijlpaarden gevoerd moet hebben*

e. *dat Cecilia de nijlpaarden wil voeren*
that Cecilia the hippos wants feed
'that Cecilia wants to feed the hippos'

f. *dat Cecilia de nijlpaarden voeren wil*

[1] Another option is *partial extraction*, also known as 'the Third Construction' (see Den Besten and Rutten (1989)), which combines properties of both verb-raising and extraposition. I will have nothing to say about partial extraction here.

g. *dat Cecilia de nijlpaarden zou willen voeren*
that Cecilia the hippos would want feed
'that Cecilia would like to feed the hippos'

h. * *dat Cecilia de nijlpaarden voeren willen zou*

As (4)b below indicates, the extraposed complement of a preposed verb is still placed after the verb cluster:

(4)a. *dat Cecilia heeft beloofd de nijlpaarden te voeren*
that Cecilia has agreed the hippos to feed
'that Cecilia has agreed to feed the hippos'

b. *dat Cecilia beloofd heeft de nijlpaarden te voeren*

7.2 Accounts of verb-raising

In this section I will present an outline of several accounts of verb-raising in Dutch, both categorial and non-categorial ones. The point is merely to give the reader an idea of what mechanisms and assumptions have been proposed for dealing with verb-raising phenomena such as those mentioned in the last section. Therefore, the descriptions (especially those of the non-categorial accounts) will all be very brief. For more details, the reader is referred to the literature.

7.2.1 Non-CG accounts

In this section I first discuss accounts of verb-raising in three non-categorial theories of grammar: generative grammar (Section 7.2.1.1), lexical-functional grammar (LFG, Section 7.2.1.2) and head-driven phrase structure grammar (HPSG, Section 7.2.1.3). Finally, Section 7.2.1.4 is concerned with Reape's theory of *word order domains*, which in the next chapter will be the basis for my categorial verb-raising fragment.

7.2.1.1 Generative grammar

The only generative account of verb-raising that I will discuss is the seminal study by Evers (1975).

An important element of Evers' account[2] is the assumption that what I have called verbal complements in Section 7.1 are in fact *sentential* complements. For example, the deep structure of (1)b, *dat Cecilia de kraanvogels zag vliegen* is assumed to be as follows:

$$[_S\ \textit{Cecilia}_{NP}\ [_S\ \textit{de kraanvogels}_{NP}\ \textit{vliegen}_V]\ \textit{zag}_V]$$

The surface structure, on the other hand, is considered to be as follows:

$$[_S\ \textit{Cecilia}_{NP}\ \textit{de kraanvogels}_{NP}\ [_V\ \textit{vliegen}_V\ \textit{zag}_V]]$$

[2] And, in some way or other, of all subsequent generative accounts.

To arrive at this surface structure from the deep structure, Evers posits a rule called V-Raising, which moves the head verb out of the embedded sentence and adjoins it to the one in the embedding sentence, creating a so-called *verb cluster*. Applying this rule to the deep structure given above yields the intermediate structure displayed below:

$$[_S\ \textit{Cecilia}_{NP}\ [\textit{de kraanvogels}_{NP}\ \_][_V\ \textit{vliegen}_V\ \textit{zag}_V]]$$

The surface structure is obtained from this intermediate structure by pruning of the deeper S node. More intricate verb-raising examples can be accounted for by recursive (or, as generative grammarians will have it, 'cyclic') application of the rules of V-Raising and S-pruning.

7.2.1.2 Lexical-functional grammar

The LFG account of verb-raising that I will discuss here is that of Bresnan *et al.* (1982).

In LFG grammars syntactic descriptions consist of two components: constituent structure and functional structure, usually abbreviated to *c-structure* and *f-structure*, respectively. Constituent structure is described by a context-free grammar. For instance, Bresnan *et al.* have the following CFG rules to account for the consituent structures underlying the basic Dutch verb-raising data:

$$\begin{aligned} \text{S} &\rightarrow \text{NP VP} \\ \text{VP} &\rightarrow (\text{NP})\ (\text{VP})\ (\text{V}') \\ \text{V}' &\rightarrow \text{V}\ (\text{V}') \end{aligned}$$

Obviously, these rules in themselves wildly overgenerate. This is where f-structure, which is formalized as an attribute-value matrix, comes into play. Its job is to impose the appropriate constraints on the connections between the elements occurring in the above rules, so that all ungrammatical examples are filtered out. The constraints are added in two places:

1. On the category symbols occurring in the context-free rewrite rules;
2. In lexical entries.

For example, the last rewrite rule above generates the equivalent of Evers' verb cluster. It is annotated in such a way that the f-structure representing the verbal complements of the mother V coincides with that of the daughter V. Lexical constraints on f-structures play a role in, among other things, the description of control phenomena. For instance, the lexical entry for *zag* has a constraint to the effect that the value of its object argument coincides with that of the subject of its verbal complement.

7.2.1.3 Head-driven phrase structure grammar

In this section I will discuss two accounts of verb-raising in HPSG, Rentier (1994) and van Noord and Bouma (1995).

Whereas LFG has two distinct components for dealing with immediate dominance and functional structure, in HPSG everything is encoded in a single dimension, which consists of feature structures. However, some of the information can still be presented in the form of context-free rewrite rules.

Rentier's (1994) account is based on the following two rules:

$$\begin{array}{rcl} \text{XP}[\text{LEX}\Leftrightarrow] & \rightarrow & \text{S},\ \text{C}_1,\ldots,\text{C}_n,\ \text{H}[\text{GOV}\langle\,\rangle,\ \text{LEX}+] \\ \text{XP}[\text{LEX}+] & \rightarrow & \text{H}[\text{GOV}\ \langle \text{C}_i\rangle,\ \text{LEX}+],\ \text{C}_i \end{array}$$

Verbal heads have a feature GOV which indicates the verbal material governed by the head. This is effectuated by the second rule, which can be described as a rule of cluster formation. Notice that the left hand side is specified as LEX+, which means that a 'lexical' verb cluster is built up. The selection of nominal arguments is based on *argument composition*, a technique closely related to categorial approaches, cf. Section 7.2.2 and the next chapter.. The idea of argument selection is that a governing verb inherits the argument list of the verb it governs. For example, Rentier's lexical entry for *zag* specifies that its argument list is the concatenation of the subject and argument list of the verb it governs. At the top level, the verb cluster combines with the accumulated list of arguments by rule 1. In this set up, the basic linear order in the verb cluster can be described by a single LP constraint: $[\text{GOV}\langle \text{X}\rangle] \prec \text{X}$.

The proposal of Van Noord and Bouma (1995) is likewise based on argument composition. The authors claim the main innovation of their approach to be the fact that only a single rule is needed to derive subordinate clauses, namely the following one:

$$\left[\begin{array}{ll} \text{SUBCAT} & \boxed{1} \\ \text{LEX} & \textit{phrasal} \end{array}\right] \rightarrow \text{L}^* \left[\begin{array}{ll} \text{SUBCAT} & \boxed{1}\cdot\langle\ldots\rangle \\ \text{LEX} & \textit{lexical} \end{array}\right] \text{R}^*$$

General principles of HPSG, in particular the valency principle, ensure that the material in $\langle\ldots\rangle$ is exactly the concatenation of L^* and R^*. The similarity with Rentier's approach extends to the lexicon. For instance, Van Noord and Bouma's entry for *zag* has the following SUBCAT value:

$$\langle \text{NP}[\textit{acc}]\rangle \cdot \boxed{1} \cdot \langle \left[\begin{array}{ll} \text{CAT} & \textit{verb}[\textit{inf}] \\ \text{SUBCAT} & \boxed{1} \\ \text{LEX} & \textit{lexical} \end{array}\right]\rangle$$

Word order is accounted for on the basis of a directionality feature, which indiactes whether constituents occur tho the left or to the right of their head. The default vlues for this feature are *right* for verbal comploements and *left* for *NP*s. An additional constraint on the rule given above now ensures that all daughters on the right hand side retain the order they have on the SUBCAT list of the head verb, while the order of left hand side daughters is reversed with respect to the order on the SUBCAT list. This yields exactly the basic Dutch verb-raising word order.

7.2.1.4 Reape's word order domains

A radically different approach to those presented so far, and one which will be of considerable interest in the sequel, was proposed by Reape (1992, to

appear, 1994). This proposal is cast in an HPSG framework. I will not go into the HPSG formalization, but limit myself to a discussion of the underlying ideas. Also, I will illustrate the theory with Dutch examples, whereas Reape focusses on German.

In his introduction to Reape (to appear), the author notes that "[n]early all modern grammatical theories derive word order from the terminal yield of phrase structure trees". As an alternative he presents his theory of word order domains, which "rejects surface syntax and its role in determining word order." Reape lists the following as the five main claims of his approach in its simplest, most general form:

1. Phrasal word order is determined within locally definable *word order domains*, which are ordered sequences of constituents.
2. Phrasal word order domains are composed compositionally from their daughter word order domains.
3. Lexical entries do not have word order domains.
4. The *functor* of a phrasal constituent is an element of its mother's domain.
5. Either
   (a) a nonfunctor daughter is an element of its mother's domain, or
   (b) the elements of a nonfunctor daughter's domain are elements of its mother's domain, and furthermore they may appear discontinuously or nonadjacently in the mother's domain — provided the relative order of the elements of the daughter domain [is] preserved in the mother's domain.

When the last option is chosen, the daughter domain is said to have been *domain unioned*[3] into its mother's domain. I will follow Kathol (1995) in referring to the first alternative as *domain insertion*.

The above list is incomplete inasmuch as it doesn't mention another important component of Reape's account, which is that the order of elements in domains is further constrained by a set of linear precedence (LP) constraints, expressed on *domain labels*. In other words, the following addition would render the list more complete:

6. In any domain, the order of the elements respects the LP constraints.

The domain labels that I referred to above are provided by a level of unordered syntactic functor-argument structure. As an example, consider the following simple grammar, where in both cases the V is the functor:

$$\begin{array}{rcl} \mathsf{S} & \Leftrightarrow & \mathsf{V},\ \mathsf{VP},\ \mathsf{NP} \\ \mathsf{VP} & \Leftrightarrow & \mathsf{V},\ \mathsf{NP} \end{array} \qquad \begin{array}{rcl} \textit{Cecilia}, \textit{Henk} & \Leftrightarrow & \mathsf{NP} \\ \textit{ziet}, \textit{lopen} & \Leftrightarrow & \mathsf{V} \end{array}$$

$$\mathsf{NP} \prec \mathsf{V}$$

[3] Domain union is similar to the *shuffle* operator of formal language theory.

The sentence *(dat) Cecilia Henk ziet lopen* is derivable in this grammar, with the following syntactic structure:

$$\{ziet_{\mathsf{V}}, \{lopen_{\mathsf{V}}, Henk_{\mathsf{NP}}\}_{\mathsf{VP}}, Cecilia_{\mathsf{NP}}\}_{\mathsf{S}}$$

Both verbs need to specify for each of their arguments whether it is unioned or inserted into the mother domain. Since in this simple example both NPs are lexical (and therefore don't project a domain), the only interesting question is what happens with the domain associated with the VP argument in the first rule. In the derivation of the example sentence, the domain associated with this VP will be the one given below — ordered in this way due to the LP constraint $\mathsf{NP} \prec \mathsf{V}$:

$$\langle Henk_{\mathsf{NP}}\ lopen_{\mathsf{V}}\rangle_{\mathsf{VP}}$$

The first option is that this VP is inserted into the mother domain. Since there are no LP constraints involving VPs, the only thing that needs to be taken care of in constructing the domain of S is that it respects the constraint $\mathsf{NP} \prec \mathsf{V}$. Concretely, this means that $Cecilia_{\mathsf{NP}}$ must come before $ziet_{\mathsf{V}}$. From this, it follows that there are three possibilities for the domain of S:

$$\langle\langle Henk_{\mathsf{NP}}\ lopen_{\mathsf{V}}\rangle_{\mathsf{VP}}\ Cecilia_{\mathsf{NP}}\ ziet_{\mathsf{V}}\rangle_{\mathsf{S}}$$
$$\langle Cecilia_{\mathsf{NP}}\ \langle Henk_{\mathsf{NP}}\ lopen_{\mathsf{V}}\rangle_{\mathsf{VP}}\ ziet_{\mathsf{V}}\rangle_{\mathsf{S}}$$
$$\langle Cecilia_{\mathsf{NP}}\ ziet_{\mathsf{V}}\ \langle Henk_{\mathsf{NP}}\ lopen_{\mathsf{V}}\rangle_{\mathsf{VP}}\rangle_{\mathsf{S}}$$

The other possibility is that *ziet* specifies that its VP argument is unioned. In that case, the domain of S will contain both elements of the daughter domain $\langle Henk_{\mathsf{NP}}\ lopen_{\mathsf{V}}\rangle_{\mathsf{VP}}$ separately, but in the same order. Moreover, $Cecilia_{\mathsf{NP}}$ will have to precede $ziet_{\mathsf{V}}$ like before. This means that in case the VP is unioned, the domain associated with S can be any of the following ones:

$$\langle Cecilia_{\mathsf{NP}}\ Henk_{\mathsf{NP}}\ ziet_{\mathsf{V}}\ lopen_{\mathsf{V}}\rangle$$
$$\langle Henk_{\mathsf{NP}}\ Cecilia_{\mathsf{NP}}\ ziet_{\mathsf{V}}\ lopen_{\mathsf{V}}\rangle$$
$$\langle Cecilia_{\mathsf{NP}}\ Henk_{\mathsf{NP}}\ lopen_{\mathsf{V}}\ ziet_{\mathsf{V}}\rangle$$
$$\langle Henk_{\mathsf{NP}}\ Cecilia_{\mathsf{NP}}\ lopen_{\mathsf{V}}\ ziet_{\mathsf{V}}\rangle$$

What isn't illustrated yet in the above example is the important case where ordering constraints apply to elements that occur on different levels in the syntactic derivation. For example, suppose the grammar were to be extended in such a way that in the example sentence *Cecilia* would be a nominative NP and *Henk* an accusative NP. The LP constraint $\mathsf{NP}[nom] \prec \mathsf{NP}[acc]$ would then be able to rule out the second and fourth of the options listed above on the basis of a constraint violation by the elements $Cecilia_{\mathsf{NP}[nom]}$ and $Henk_{\mathsf{NP}[acc]}$, even though these elements occur on different syntactic levels.

Next, let me point out a redundancy in Reape's set up that will become important in the next chapter, where I will propose a categorial account of domains. This redundancy consists of the overlap in clauses (5b) and (6), repeated below for the reader's convenience:

5. (b) the elements of a nonfunctor daughter's domain are elements of its mother's domain, and furthermore they may appear discontinuously or nonadjacently in the mother's domain — provided the relative order of the elements of the daughter domain [is] preserved in the mother's domain.

6. In any domain, the order of the elements respects the LP constraints.

Suppose we would drop the requirement that in the case of domain unioning, the relative order of the elements of the daughter domain be preserved in the mother's domain. Would this result in an increase in the number of admissible orderings? The answer is negative, since the order of the elements in the daughter's domain only needs to be consistent with the LP constraints in the first place. Therefore, if a unioned domain satisfies the condition in (6), then the condition in (5b) is satisfied automatically. Incidentally, note that the converse does not hold, due to the fact that the condition in (6) also constrains the order of elements occurring on different syntactic levels.

The import of all this is that it allows us to separate two components of Reape's proposal:

- ▷ Domain structure
- ▷ Linear precedence

First, the domain structure accompanying some expression can be determined. After that, the LP constraints can be imposed on this domain (and, if necessary, recursively on the domains occurring in it) in order to determine the admissible orderings of the elements it contains.

Referring back to the example sentence, it is easy to see that in the case of domain unioning of the VP argument, $Henk_{\mathsf{NP}}$ will necessarily be ordered before $lopen_{\mathsf{V}}$ in the domain of S, even if the condition in (5b) is dropped. The separation of the two components of Reape's proposal can be illustrated by the case of domain insertion of the VP. First, the following domain structure is built up:[4]

$$\{ziet_{\mathsf{V}}, \{lopen_{\mathsf{V}}, Henk_{\mathsf{NP}}\}_{\mathsf{VP}}, Cecilia_{\mathsf{NP}}\}_{\mathsf{S}}$$

Next, the top level domain is ordered, giving these three possibilities:

$$\langle \{lopen_{\mathsf{V}}, Henk_{\mathsf{NP}}\}_{\mathsf{VP}}\ Cecilia_{\mathsf{NP}}\ ziet_{\mathsf{V}} \rangle_{\mathsf{S}}$$
$$\langle Cecilia_{\mathsf{NP}}\ \{lopen_{\mathsf{V}}, Henk_{\mathsf{NP}}\}_{\mathsf{VP}}\ ziet_{\mathsf{V}} \rangle_{\mathsf{S}}$$
$$\langle Cecilia_{\mathsf{NP}}\ ziet_{\mathsf{V}}\ \{lopen_{\mathsf{V}}, Henk_{\mathsf{NP}}\}_{\mathsf{VP}} \rangle_{\mathsf{S}}$$

Finally, the LP constraints are applied recursively to determine the linear order of the elements in the VP-subdomain, leading to the same three ordered domains that were obtained earlier by directly applying Reape's rules:

$$\langle \langle Henk_{\mathsf{NP}}\ lopen_{\mathsf{V}} \rangle_{\mathsf{VP}}\ Cecilia_{\mathsf{NP}}\ ziet_{\mathsf{V}} \rangle_{\mathsf{S}}$$
$$\langle Cecilia_{\mathsf{NP}}\ \langle Henk_{\mathsf{NP}}\ lopen_{\mathsf{V}} \rangle_{\mathsf{VP}}\ ziet_{\mathsf{V}} \rangle_{\mathsf{S}}$$
$$\langle Cecilia_{\mathsf{NP}}\ ziet_{\mathsf{V}}\ \langle Henk_{\mathsf{NP}}\ lopen_{\mathsf{V}} \rangle_{\mathsf{VP}} \rangle_{\mathsf{S}}$$

[4] This domain structure is similar to the syntactic structure of the sentence given earlier only because no domain unioning has taken place.

7.2.2 Categorial accounts

In this section I will discuss four categorial approaches to verb-raising:

- ▷ Bach (1984);
- ▷ Steedman (1984, 1985), which was refined in Houtman (1984);
- ▷ Moortgat and Oehrle (1993a);
- ▷ Bouma and van Noord (1994b).

7.2.2.1 Bach

The first published account of verb-raising in categorial grammar is due to Emmon Bach (1984). His discussion is limited to the most basic instance of verb-raising in Dutch, as illustrated in the by the now familiar sentence:

(7.1) (*dat*) *ik Henk Cecilia de nijlpaarden zag helpen voeren*

Bach uses a mixture of directed and non-directed categories, for which he gives the usual **AB** reduction schemata:

$$\begin{array}{ll} \mathsf{a/b, b} \vdash \mathsf{a} & \mathsf{b} \rightarrow \mathsf{a, b} \vdash \mathsf{a} \\ \mathsf{b, b\backslash a} \vdash \mathsf{a} & \mathsf{a, b} \rightarrow \mathsf{a} \vdash \mathsf{a} \end{array}$$

Furthermore, in order to explain the cross-serial dependencies, another kind of slash is introduced. These *infix* slashes are defined by means of the following reduction schemata:

$$\left.\begin{array}{l} \mathsf{a//b}, \varphi, \mathsf{b} \vdash \mathsf{a}, \varphi \\ \mathsf{b} \Rightarrow \mathsf{a}, \varphi, \mathsf{b} \vdash \mathsf{a}, \varphi \end{array}\right\} \quad \text{for } \mathsf{a} \neq \mathsf{b} \qquad \left.\begin{array}{l} \mathsf{b}, \varphi, \mathsf{b\backslash\backslash a} \vdash \varphi, \mathsf{a} \\ \mathsf{b}, \varphi, \mathsf{b} \Rightarrow \mathsf{a} \vdash \varphi, \mathsf{a} \end{array}\right\} \quad \text{for } \mathsf{a} \neq \mathsf{b}$$

$$\mathsf{a//a}, \varphi, \mathsf{a} \vdash \varphi, \mathsf{a}$$

$$\mathsf{a}, \varphi, \mathsf{a\backslash\backslash a} \vdash \varphi, \mathsf{a}$$

As can be seen from these rules, Bach has different definitions depending on whether the argument category of the discontinuous slash equals its result category or not. This is a familiar distinction within CG, introduced by Vennemann and Harlow (XXXX): when an $\mathsf{A} \rightarrow \mathsf{B}$ combines with an A, the functor is the head of the resulting combination, unless $\mathsf{A} = \mathsf{B}$, in which case the argument is the head. Bach's rules can therefore be conceived as defining the discontinuous slashes to be *head attraction* operators, since whenever the application rule is used, the head determines where the resulting complex expression ends up. Bach doesn't motivate this distinction, but it need not immediately concern us here, since it doesn't play a role in the analysis of the example sentence. Compare however the proposal of Moortgat and Oehrle in Section 7.2.2.3. The words occurring in the example sentence are assigned the following types:

ik, Henk, Cecilia, de nijlpaarden:	T
zag:	$(\mathsf{T\backslash\backslash S'})/\mathsf{S}$
laten:	$(\mathsf{T\backslash\backslash S})/\mathsf{S}$
voeren:	$\mathsf{T\backslash\backslash(T\backslash\backslash S)}$

Given these lexical type assignments and some additional assumptions, there is exactly one way to derive a sentence on the basis of these type assignments, which is illustrated in the derivation below.

$$\frac{\underline{ik}}{\mathsf{T}}\quad\frac{\underline{Henk}}{\mathsf{T}}\quad\frac{\underline{Cecilia}}{\mathsf{T}}\quad\frac{\underline{\textit{de nijlpaarden}}}{\mathsf{T}}\quad\frac{\underline{zag}}{(\mathsf{T}\backslash\!\backslash\overline{\mathsf{S}'})/\mathsf{S}}\quad\frac{\underline{helpen}}{(\overline{\mathsf{T}}\backslash\!\backslash\mathsf{S})/\mathsf{S}}\quad\frac{\underline{voeren}}{\mathsf{T}\backslash\!\backslash(\mathsf{T}\backslash\!\backslash\mathsf{S})}$$

$$\frac{\underline{ik}}{\mathsf{T}}\quad\frac{\underline{Henk}}{\mathsf{T}}\quad\frac{\underline{Cecilia}}{\mathsf{T}}\quad\frac{\underline{zag}}{(\mathsf{T}\backslash\!\backslash\overline{\mathsf{S}'})/\mathsf{S}}\quad\frac{\underline{helpen}}{(\overline{\mathsf{T}}\backslash\!\backslash\mathsf{S})/\mathsf{S}}\quad\frac{\underline{\textit{de nijlpaarden voeren}}}{\mathsf{T}\backslash\!\backslash\mathsf{S}}$$

$$\frac{\underline{ik}}{\mathsf{T}}\quad\frac{\underline{Henk}}{\mathsf{T}}\quad\frac{\underline{zag}}{(\mathsf{T}\backslash\!\backslash\overline{\mathsf{S}'})/\mathsf{S}}\quad\frac{\underline{helpen}}{(\overline{\mathsf{T}}\backslash\!\backslash\mathsf{S})/\mathsf{S}}\quad\frac{\underline{\textit{Cecilia de nijlpaarden voeren}}}{\mathsf{S}}$$

$$\frac{\underline{ik}}{\mathsf{T}}\quad\frac{\underline{Henk}}{\mathsf{T}}\quad\frac{\underline{zag}}{(\mathsf{T}\backslash\!\backslash\overline{\mathsf{S}'})/\mathsf{S}}\quad\frac{\underline{\textit{helpen Cecilia de nijlpaarden voeren}}}{\mathsf{T}\backslash\!\backslash\mathsf{S}}$$

$$\frac{\underline{ik}}{\mathsf{T}}\quad\frac{\underline{zag}}{(\mathsf{T}\backslash\!\backslash\overline{\mathsf{S}'})/\mathsf{S}}\quad\frac{\underline{\textit{Henk helpen Cecilia de nijlpaarden voeren}}}{\mathsf{S}}$$

$$\frac{\underline{zag}}{\mathsf{T}\backslash\!\backslash\overline{\mathsf{S}'}}\quad\frac{\underline{\textit{ik Henk helpen Cecilia de nijlpaarden voeren}}}{\mathsf{S}}$$

$$\frac{\underline{\textit{zag ik Henk helpen Cecilia de nijlpaarden voeren}}}{\mathsf{S}'}$$

7.2.2.2 Steedman/Houtman

An early detailed CG account of verb-raising is that of Steedman (1984, 1985]. His starting point is a system on non-directed categories. Word-order constraints are implemented by means of restrictions on the combination rules. To illustrate how this works, I will again use the by now familiar example sentence (*dat*) *ik Henk Cecilia de nijlpaarden zag helpen voeren*. The lexical items occurring in this sentence are assigned the following categories by Steedman:

ik:	$\mathsf{FVP}\rightarrow\mathsf{S}$
Cecilia, *Henk*, *de nijlpaarden*:	NP
zag:	$\mathsf{Sinf}\rightarrow\mathsf{FVP}$
helpen:	$\mathsf{Sinf}\rightarrow(\mathsf{NP}\rightarrow\mathsf{Sinf})$
voeren:	$\mathsf{NP}\rightarrow(\mathsf{NP}\rightarrow\mathsf{Sinf})$

Note that Steedman basically assigns *zag* and *helpen* the category $\mathsf{S}\rightarrow(\mathsf{NP}\rightarrow\mathsf{S})$, rather than most other authors' preferred choice of $\mathsf{VP}\rightarrow(\mathsf{NP}\rightarrow\mathsf{VP})$. The unusual type assignment for the subject *ik* is needed to account for verb second, but doesn't add anything to the analysis of the example.

The combination rules that are used in the derivation of the example are the following ones:

Forward Partial Combination
$\mathsf{Y} \rightarrow \mathsf{X}, \mathsf{Z} \rightarrow \mathsf{Y\$} \vdash \mathsf{Z} \rightarrow \mathsf{X\$}$

Forward Combination
$\mathsf{Y} \rightarrow \mathsf{X}, \mathsf{Y} \vdash \mathsf{X}$ where $\mathsf{X} \notin \{\mathsf{S\$}, \mathsf{Sinf\$}, \mathsf{FVP\$}, \mathsf{VP\$}, \ldots\}$
or $\mathsf{Y} \notin \{\mathsf{NP}, \mathsf{PP}, \mathsf{AP}\}$

Backward Combination
$\mathsf{Y} \vdash \mathsf{X}, \mathsf{Y} \rightarrow \mathsf{X}$ where $\mathsf{X} \in \{\mathsf{S\$}, \mathsf{Sinf\$}, \mathsf{FVP\$}, \mathsf{VP\$}, \ldots\}$
and $\mathsf{Y} \notin \{\mathsf{NP}, \mathsf{PP}, \mathsf{AP}\}$

In these rules, the notation $\mathsf{X\$}$ denotes any category whose ultimate result category is X. The example sentence now has the following derivation:

$$\dfrac{\mathsf{FVP} \rightarrow \mathsf{S} \qquad \dfrac{\mathsf{NP} \qquad \dfrac{\mathsf{NP} \qquad \dfrac{\mathsf{NP} \qquad \dfrac{\mathsf{Sinf} \rightarrow \mathsf{FVP} \quad \mathsf{Sinf} \rightarrow (\mathsf{NP} \rightarrow \mathsf{Sinf})}{\mathsf{Sinf} \rightarrow (\mathsf{NP} \rightarrow \mathsf{FVP})}\,FP \qquad \mathsf{NP} \rightarrow (\mathsf{NP} \rightarrow \mathsf{Sinf})}{\mathsf{NP} \rightarrow (\mathsf{NP} \rightarrow (\mathsf{NP} \rightarrow \mathsf{FVP}))}\,B}{\mathsf{NP} \rightarrow (\mathsf{NP} \rightarrow \mathsf{FVP})}\,B}{\mathsf{NP} \rightarrow \mathsf{FVP}}\,B}{\mathsf{FVP}}\,F}{\mathsf{S}}$$

A proposal which draws heavily on Steedman's is [Houtman, 1984]. Nonetheless, there are a few differences between both accounts even where the analysis of the basic case is concerned.

First of all, Houtman assigns verbs such as *leren* and *helpen* the category $\mathsf{VP}(1) \rightarrow (\mathsf{NP} \rightarrow \mathsf{VP}(1))$ rather than Steedman's somewhat idiosyncratic $\mathsf{S} \rightarrow (\mathsf{NP} \rightarrow \mathsf{S})$. This category assignment includes the distinction between various statuses of verbs introduced in Bech (1955) — $\mathsf{V}(i)$ denotes a verb that has ith status.

Furthermore, it turns out that the original proposal freely allowed VPR (Verb Projection Raising) to occur, something that was apparently overlooked by Steedman. Thus, in Steedman's system, it is easy to derive a sentence like:

(7.2) (*dat*) *ik Cecilia Henk zag helpen de nijlpaarden voeren*

Houtman solves this problem by introducing a feature L which indicates whether or not a certain phrase is lexical. Verb-raising verbs are then exactly those verbs that require their verbal complements to bear this feature.

Houtman also considers optional and obligatory extraposition. He proposes to account for obligatory extraposition by assigning verbs like *zich voornemen* the type $\mathsf{VP}(2) \rightarrow (\mathsf{NP} \rightarrow \mathsf{VP}(0))$, whereas optional extraposition is explained in terms of underspecification of the VP complement of e.g. *proberen* for the feature L.

7.2.2.3 Moortgat & Oehrle

A multimodal account of verb-raising is given by Moortgat and Oehrle (1993a). Their proposal shares with that of Bach discussed in Section 7.2.2.1 the reliance on a notion of headedness. However, while Bach's notion of head is implicit, and a function of the types involved in a combination, Moortgat and Oehrle add an explicit dimension of *dependency* to their system. They have a total of six modes of combination: two modes of dependency-sensitive concatenation $\bullet_l$ and $\bullet_r$, in which the subscript indicates whether the left or the right argument is the head, and four modes of *head wrapping* (infixing an expression in another one, next to its head) $\bullet_{l/r,h/d}$, where the first subscript indicates which argument is the infix, and the second subscript indicates whether the infix is the head or the dependent part of the combination. These intuitive definitions, and the fact that dependency concatenation can be seen as a special case of head wrapping, suggest interactions between both types of combination modes that are formalized in the following interaction postulates, where w stands for either r or l:

$$\begin{array}{ll} \mathsf{A} \bullet_l \mathsf{B} \vdash \mathsf{A} \bullet_{lh} \mathsf{B} & (\mathsf{A} \bullet_{rw} \mathsf{B}) \bullet_l \mathsf{C} \vdash (\mathsf{A} \bullet_l \mathsf{C}) \bullet_{rw} \mathsf{B} \\ \mathsf{A} \bullet_l \mathsf{B} \vdash \mathsf{A} \bullet_{rd} \mathsf{B} & \mathsf{A} \bullet_r (\mathsf{B}) \bullet_{rw} \mathsf{C} \vdash (\mathsf{A} \bullet_r \mathsf{B}) \bullet_{rw} \mathsf{C} \\ \mathsf{A} \bullet_r \mathsf{B} \vdash \mathsf{A} \bullet_{ld} \mathsf{B} & \mathsf{A} \bullet_r (\mathsf{B}) \bullet_{lw} \mathsf{C} \vdash \mathsf{B} \bullet_{lw} (\mathsf{A}) \bullet_r \mathsf{C} \\ \mathsf{A} \bullet_r \mathsf{B} \vdash \mathsf{A} \bullet_{rh} \mathsf{B} & (\mathsf{A} \bullet_{lw} \mathsf{B}) \bullet_l \mathsf{C} \vdash \mathsf{A} \bullet_{lw} (\mathsf{B}) \bullet_l \mathsf{C} \end{array}$$

The lexical type assignments for the verbs occurring in Moortgat and Oehrle's example verb phrase *boeken wil kunnen lezen* are as follows:

$$\begin{array}{rl} \textit{wil}: & \mathsf{VP}/_{lh}\mathsf{INF} \\ \textit{kunnen} & \mathsf{INF}/_{lh}\mathsf{INF} \\ \textit{lezen} & \mathsf{NP}\backslash_r\mathsf{INF} \end{array}$$

The schematic presentation below (in which the last step, where head wrapping reduces to simple dependency concatenation isomitted) now illustrates how the appropriate word order is arrived at:

Moortgat and Oehrle contrast their approach with accounts positing a so-called rule of *disharmonic composition* $\mathsf{C/B, A\backslash B \vdash A\backslash C}$, which can be seen as the directed version of the proposal of Steedman that I discussed in the last section. The addition of such rules very easily lead to collapse of the directed system to **LP**, so that no predictions about word order can be made. In Moortgat and Oehrle's set up disharmonic composition doesn't need to be added as a basic rule. Rather, the following version of it is derivable in the system:

$$\mathsf{VP}/_{lh}\mathsf{INF}, \mathsf{NP}\backslash_r\mathsf{INF})^{lh} \vdash \mathsf{NP}\backslash_r\mathsf{VP}$$

7.2.2.4 Bouma & Van Noord

A recent and fairly detailed treatment of verb-raising in Dutch is Bouma and Van Noord (1994b). The most noticable feature of their account, especially when compared to that of Steedman discussed in Section 7.2.2.2 is perhaps that they make no explicit use of function composition. Rather, they derive all cases where Steedman's system uses function composition by means of combinations of the rules of division and application. Their lexical type assignments for verbs are of the following sort, where VP abbreviates NP\S:

wil :	($\backslash$VP)/($\backslash$VP)
laten :	($\backslash$(NP\VP))/($\backslash$VP)
voeren :	NP\VP

Here, '$' represents an arbitrary number of arguments, all of which must be dominated by a '\'. The following derivation then illustrates the sentencehood of the sentence (*dat*) *Cecilia Henk de nijlpaarden wil laten voeren*:

$$\dfrac{\mathsf{NP}\quad\dfrac{\mathsf{NP}\quad\dfrac{\mathsf{NP}\quad\dfrac{(\$\backslash\mathsf{VP})/(\$\backslash\mathsf{VP})\quad\dfrac{(\$\backslash(\mathsf{NP}\backslash\mathsf{VP}))/(\$\backslash\mathsf{VP})\quad\mathsf{NP}\backslash\mathsf{VP}}{\mathsf{NP}\backslash(\mathsf{NP}\backslash\mathsf{VP})}}{\mathsf{NP}\backslash(\mathsf{NP}\backslash\mathsf{VP})}}{\mathsf{NP}\backslash\mathsf{VP}}}{\mathsf{VP}}}{\mathsf{S}}$$

Bouma and Van Noord argue that this way of doing things is superior to the use of a rule of disharmonic composition $\mathsf{C/B, A\backslash B \vdash A\backslash C}$ (cf. the end of the last section), since it automatically limits the (implicit) applications of that rule to the cases where they are needed. Finally, verbs that require extraposition, such as *verbieden*, are assigned the single category (NP\VP)/VP by Bouma and Van Noord.

Chapter 8

Word order domains in categorial grammar

In this chapter I will re-examine Reape's theory of word order domains from a multimodal categorial perspective. The two components of Reape's proposal individuated earlier are discussed separately: domain structure is the subject of Sections 8.1 (basic domain structure) and 8.2 (labeling), while Section 8.4 shows how to incorporate linear order constraints. In between is a brief interlude in which two related proposals are briefly discussed (Section 8.3). A final ingredient needed to get the basic verb-raising examples right, a modal rendering of the nesting depth of resources, is presented in Section 8.5. Section 8.6 then lists the lexical type assignments for the verb-raising fragment, accompanied by a few illustrative derivations. The two sections following that one concern extensions to the basic fragment. In Section 8.7, I show that this system has no problem whatsoever in dealing with adjuncts and their scope (as opposed to several other categorial accounts), while in Section 8.8 separable verb prefixes are discussed. The chapter ends in Section 8.9 with an evaluation.

8.1 Basic domain structure

As I showed in Section 5.1.3, the use of unary residuation modalities naturally introduces a notion of structuring into categorial grammar which is orthogonal to that provided by the product. In this section, I will show how this fact makes it possible to import the ideas behind Reape's word order domains into categorial grammar in a straightforward way.

Recall from Section 5.2.2 that there are two basic approaches to describe systems whose behavior falls in between that of **L** and that of **LP**. One possibility is to start from the commutative system **LP** and then try to regain the required order sensitivity through the addition of modalities. Alternatively, **L** can be taken as the basic system, with order restrictions being relaxed through modal postulates. The division of Reape's proposal that I made in Section 7.2.1.4

suggests that it is most compatible with the first of the above approaches: domain structure is independent of linear order and can therefore be described in a commutative setting; additional modal mechanisms will then need to take care of imposing the right LP constraints.

The idea underlying the multimodal implementation of the structural component of word order domains is to translate the subcategorisation requirements into **LP**$\Diamond$ type assignments, where the modal status of an argument determines whether it is unioned or inserted into its mother's domain. The categorial machinery accounting for domain structure and linear order will be explained by a method of stepwise refinement. To illustrate what is going on, after each step I will present lexical type assignments and a derivation for the following simple Dutch sentence:

(1) *de nijlpaarden* *eten*
the hippos eat
'The hippos are eating'

Below is the first such illustration, for the basic system **LP**:

$$\begin{aligned} \textit{de}: &\quad \mathsf{N} \to \mathsf{NP} \\ \textit{nijlpaarden}: &\quad \mathsf{N} \\ \textit{eten}: &\quad \mathsf{NP} \to \mathsf{S} \end{aligned}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\textit{nijlpaarden} \vdash \mathsf{N} \quad \mathsf{NP} \vdash \mathsf{NP}}{\mathsf{N} \to \mathsf{NP}, \textit{nijlpaarden} \vdash \mathsf{NP}}\,[\to\mathrm{L}]}{\textit{de}, \textit{nijlpaarden} \vdash \mathsf{NP}}\,[\mathrm{Lex}] \qquad \mathsf{S} \vdash \mathsf{S}}{\textit{de}, \textit{nijlpaarden}, \mathsf{NP} \to \mathsf{S} \vdash \mathsf{S}}\,[\to\mathrm{L}]}{\textit{de}, \textit{nijlpaarden}, \textit{eten} \vdash \mathsf{S}}\,[\mathrm{Lex}]$$

In all its simplicity this first approximation already incorporates an important component of Reape's proposal, namely the syntactic level of functor-argument structure.

Note that although from a derivational point of view an NP constituent consisting of a determiner and a noun is present, this is not reflected in the flat structure of the antecedent of the final sequent. Since antecedents of **LP** sequents are always flat, domain unioning in fact becomes the default operation for syntactic combination when **LP** is chosen as the basic system.

How can the fact be expressed that the domain of the NP in the above sentence needs to be inserted rather than unioned into its mother's domain? It is here that the structuring capabilities of modalities come into the picture. Suppose we have some elements whose respective types combine into the type A. From the [$\Diamond$R]-rule it can be seen that from this material the type $\Diamond$A is derivable provided that the material is structured in the appropriate way. This structure is expressed in sequent antecedents through the unary structuring operator, which is written as $(.)^{\Diamond}$. In other words, for a functor to subcategorize for a $\Diamond$A rather than just a A means that it forces the material making up the argument to be structured. It is this structure that will play a role similar to that of domain structuring in Reape's theory.

Let's have another look at the example sentence from this new perspective. In Reape's view, the fact that the domain of NP combines with the intransitive verb by domain insertion is determined by the functor, i.e. the verb. This can be expressed in **LP**$\Diamond$ by changing the type assignment for the verb from $\mathsf{NP} \rightarrow \mathsf{S}$ to $\Diamond\mathsf{NP} \rightarrow \mathsf{S}$, which would yield the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathit{nijlpaarden} \vdash \mathsf{N} \quad \mathsf{NP} \vdash \mathsf{NP}}{\mathsf{N} \rightarrow \mathsf{NP}, \mathit{nijlpaarden} \vdash \mathsf{NP}}\,[\mathrm{Lex}]}{\mathit{de}, \mathit{nijlpaarden} \vdash \mathsf{NP}}\,[\Diamond\mathrm{R}] \qquad \mathsf{S} \vdash \mathsf{S}}{(\mathit{de}, \mathit{nijlpaarden})^{\Diamond}, \Diamond\mathsf{NP} \rightarrow \mathsf{S} \vdash \mathsf{S}}\,[\mapsto\mathrm{L}]}{(\mathit{de}, \mathit{nijlpaarden})^{\Diamond}, \mathit{eten} \vdash \mathsf{S}}\,[\mathrm{Lex}]$$

Notice carefully that the additional structure of the antecedent is not only *licensed*, but in fact *required*[1] by the presence of the modality, which is of course what we want.

Reape's approach is unusual in that it always lets the functor decide for each of its elements whether its domain is to be unioned or inserted into the mother domain. The standard view for at least a number of important cases, such as noun phrases and sentences in most languages, is exactly the opposite one. This is usually expressed by saying that categories such as NP and S are so-called *bounding nodes*, cf. the discussion of Dowty (to appear) in Section 8.3.

The duality of the modalities $\Diamond$ and $\Box$ ensures that this alternative can be expressed just as easily in a multimodal categorial grammar. More specifically, the [$\Box$R]-rule requires the same modal structuring as the [$\Diamond$L]-rule, so that an argument can be made to project a domain by ensuring that its type is prefixed with a $\Box$. For example, if we want noun phrases consisting of a determiner and a noun to project a domain, this can be achieved by changing the type assignment for determiners from $\mathsf{N} \rightarrow \mathsf{NP}$ to $\mathsf{N} \rightarrow \Box\mathsf{NP}$. Similarly, the fact that sentences are domains can be incorporated in the grammar by changing the type assignment for intransitive verbs from $\mathsf{NP} \rightarrow \mathsf{S}$ to $\mathsf{NP} \rightarrow \Box\mathsf{S}$. This does not mean that functors projecting a domain for their argument will not play a role at all in the fragment — they will be used in order to account for extraposition of verbal complements, see Section 8.6. These assignments yield the same structure as before, but with additional structuring of the sentential domain, as can be seen from the derivation below:

de :	$\mathsf{N} \rightarrow \Box\mathsf{NP}$
nijlpaarden :	N
eten :	$\mathsf{NP} \rightarrow \Box\mathsf{S}$

[1] Cf. however the discussion about projecting and erasing below.

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{NP} \vdash \mathsf{NP}}{(\Box\mathsf{NP})^{\Diamond} \vdash \mathsf{NP}}\,[\Box\mathrm{L}] \qquad \mathit{nijlpaarden} \vdash \mathsf{N}}{(\mathsf{N} \to \Box\mathsf{NP}, \mathit{nijlpaarden})^{\Diamond} \vdash \mathsf{NP}}\,[\to\mathrm{L}]}{(\mathit{de}, \mathit{nijlpaarden})^{\Diamond} \vdash \mathsf{NP}}\,[\mathrm{Lex}] \qquad \dfrac{\mathsf{S} \vdash \mathsf{S}}{(\Box\mathsf{S})^{\Diamond} \vdash \mathsf{S}}}{((\mathit{de}, \mathit{nijlpaarden})^{\Diamond}, \mathsf{NP} \to \Box\mathsf{S})^{\Diamond} \vdash \mathsf{S}}\,[\to\mathrm{L}]}{((\mathit{de}, \mathit{nijlpaarden})^{\Diamond}, \mathit{eten})^{\Diamond} \vdash \mathsf{S}}\,[\mathrm{Lex}]$$

Again, the extra structure is imposed rather than just allowed by the additional modal decoration.

We've seen that domain structuring can be achieved by adding either a $\Diamond$ in a negative position (i.e. on the right hand side of sequents) or a $\Box$ in a positive position (on the left hand side of sequents). There's a dual side to this too, in the sense that positive occurrences of $\Diamond$ and negative occurrences of $\Box$ have the effect of reversing this behavior. Thus, in a context where expressions of type A normally project domains, this can be overruled by a functor looking for a $\Box\mathsf{A}$ instead of just an A. Conversely, expressions of type $\Diamond\mathsf{A}$ can combine with functors specifying their type A arguments to be domains without actually being structured this way. These observations are summarized in the table below:

	datum (left)	goal (right)
$\Diamond$	erase	project
$\Box$	project	erase

Similar behavior is displayed by Morrill's (1994) unary operators $[]$ and $[]^{-1}$, with matching antecedent structuring $[\ldots]$ (bracketing) and $[\ldots]^{-1}$ (antibracketing). The proof rules for these connectives are quite difference from the ones used here, however. In particular, both are each other's inverses, so that for these modalities it holds that:

$$[][]^{-1}\mathsf{A} \Leftrightarrow \mathsf{A} \Leftrightarrow []^{-1}[]\mathsf{A}$$

This means that Morrill's unary operators are not as strong as residuation modalities, for which the analogs of the above equations hold in only one direction: $\Diamond\Box\mathsf{A} \vdash \mathsf{A} \vdash \Box\Diamond\mathsf{A}$, but not the other way round.

8.2 Labeling

In the previous section I showed how the addition of a modality to **LP** allows an account of the basic structure underlying word order domains. However, this basic domain structure doesn't contain enough information for it to be able to serve as the input to the component which imposes the linear precedence constraints. The reason lies in my earlier observation that the LP constraints are expressed on the labels of domains, and the domains introduced above weren't labeled.

In principle it is very easy to add domain labels: instead of a single pair of modalities $\Diamond$ and $\Box$, just assume a pair of modalities $\langle x\rangle$ and $[x]$ for each label $x \in \mathcal{L}$, where $\mathcal{L}$ denotes the set of all labels.

To illustrate this, consider again the example sentence *de nijlpaarden eten*. The difference between the NP and S domains can be expressed by adding the set of modalities $\{\langle np\rangle, [np], \langle s\rangle, [s]\}$ instead of just $\{\Diamond, \Box\}$. The type assignments are changed accordingly: *de* will now be listed as an $\mathsf{N} \to [np]\mathsf{NP}$, and *eten* as an $\mathsf{NP} \to [s]\mathsf{S}$, yielding the labeled domain structure in the conclusion sequent of the derivation below:

$$\begin{aligned} \textit{de}: &\quad \mathsf{N} \to [np]\mathsf{NP} \\ \textit{nijlpaarden}: &\quad \mathsf{N} \\ \textit{eten}: &\quad \mathsf{NP} \to [s]\mathsf{S} \end{aligned}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{NP} \vdash \mathsf{NP}}{([np]\mathsf{NP})^{np} \vdash \mathsf{NP}}\,[[np]\mathrm{L}] \qquad \textit{nijlpaarden} \vdash \mathsf{N}}{(\mathsf{N} \to [np]\mathsf{NP}, \textit{nijlpaarden})^{np} \vdash \mathsf{NP}}\,[\to\mathrm{L}]}{(\textit{de}, \textit{nijlpaarden})^{np} \vdash \mathsf{NP}}\,[\mathrm{Lex}] \qquad \dfrac{\mathsf{S} \vdash \mathsf{S}}{([s]\mathsf{S})^{s} \vdash \mathsf{S}}\,[[s]\mathrm{L}]}{((\textit{de}, \textit{nijlpaarden})^{np}, \mathsf{NP} \to [s]\mathsf{S})^{s} \vdash \mathsf{S}}\,[\to\mathrm{L}]}{((\textit{de}, \textit{nijlpaarden})^{np}, \textit{eten})^{s} \vdash \mathsf{S}}\,[\mathrm{Lex}]$$

One problem remains, which has to do with the labeling of lexical items. The complication here is that, for instance, a verb such as *eten* can't just be assigned the type $\langle v\rangle(\mathsf{NP} \to [s]\mathsf{S})$ in order to mark it as a verb for the benefit of the LP constraint checking procedure, for how are we ever going to get rid of the $\langle v\rangle$, so that the verb can combine with its subject? The solution is provided by the alternative characterization of the residuation laws for labeled modalities:

$$\langle x\rangle[x]\mathsf{A} \quad \vdash \quad \mathsf{A} \quad \vdash \quad [x]\langle x\rangle\mathsf{A}$$

What the first half of this tells us is that the combination $\langle x\rangle[x]$ can function as a kind of optional feature marking: it is labeled x by virtue of $\langle x\rangle$ being its main connective, but when this label has played its part in the LP constraint checking procedure, it can be discarded thanks to the derivability of the sequent $\langle x\rangle[x]\mathsf{A} \vdash \mathsf{A}$. The combination $\langle x\rangle[x]$ will pop up often enough in the sequel that it is expedient to introduce a special notation for it. In order to keep it as simple as possible to understand the type assignments and derivations, I will introduce several such abbreviations as I go along. These will all be clearly marked.

Abbreviation 1 $\quad (x) =_{\mathsf{def}} \langle x\rangle[x]$

Since the point of this notation is that (x) can be discarded when it is no longer needed, the frequently occurring subderivation that takes care of that will be written down in a single step.

Abbreviation 2

$$\dfrac{\dfrac{\dfrac{, [\mathsf{A}] \vdash \mathsf{B}}{, [([x]\mathsf{A})^{x}] \vdash \mathsf{B}}\,[[x]\mathrm{L}]}{, [\langle x\rangle[x]\mathsf{A}] \vdash \mathsf{B}}\,[\langle x\rangle\mathrm{L}]}{, [(x)\mathsf{A}] \vdash \mathsf{B}}\,[\mathrm{Def}] \quad \rightsquigarrow \quad \dfrac{, [\mathsf{A}] \vdash \mathsf{B}}{, [(x)\mathsf{A}] \vdash \mathsf{B}}\,[(x)\mathrm{L}]$$

Including the modifications suggested by the above discussion leads to the lexical assignments for the example sentence that are given below. With these type assignments, the antecedent of the sequent corresponding to the example sentence will be structured like it was above, the crucial difference being that lexical items are now marked in such a way that they can be checked against the LP constraints.

$$\begin{aligned} \textit{de}: &\quad (det)(\mathsf{N} \rightarrow [np]\mathsf{NP}) \\ \textit{nijlpaarden}: &\quad (n)\mathsf{N} \\ \textit{eten}: &\quad (v)(\mathsf{NP} \rightarrow [s]\mathsf{S}) \end{aligned}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{NP} \vdash \mathsf{NP}}{([np]\mathsf{NP})^{np} \vdash \mathsf{NP}}\,[[np]\mathrm{L}] \quad \mathsf{N} \vdash \mathsf{N}}{(\mathsf{N} \rightarrow [np]\mathsf{NP}, \mathsf{N})^{np})^{np} \vdash \mathsf{NP}}\,[\rightarrow\mathrm{L}] \quad \dfrac{\mathsf{S} \vdash \mathsf{S}}{([s]\mathsf{S})^{s} \vdash \mathsf{S}}\,[[s]\mathrm{L}]}{((\mathsf{N} \rightarrow [np]\mathsf{NP}, \mathsf{N})^{np}, \mathsf{NP} \rightarrow [s]\mathsf{S})^{s} \vdash \mathsf{S}}\,[\rightarrow\mathrm{L}]}{(((det)(\mathsf{N} \rightarrow [np]\mathsf{NP}), (n)\mathsf{N})^{np}, (v)(\mathsf{NP} \rightarrow [s]\mathsf{S}))^{s} \vdash \mathsf{S}}\,[(det)\mathrm{L},\ (n)\mathrm{L},\ (v)\mathrm{L}]}{((\textit{de}, \textit{nijlpaarden})^{np}, \textit{eten})^{s} \vdash \mathsf{S}}\,[\mathrm{Lex}]$$

8.3 Related proposals

In the next section I will show how LP constraints can be added to the systems introduced above. But first, in this section I will discuss two categorial approaches to discontinuity that are similar in spirit to the one introduced in the previous sections: Hoeksema's (1991) 'categorial liberation' and Dowty's (to appear) 'minimalist program'. Both these authors were inspired by Pullum and Zwicky's notion of *liberation*, which I will therefore discuss first.

Liberation was introduced by Pullum (1982) as the following (language-specific) GPSG metarule:

$$\frac{\mathrm{B} \rightarrow \mathrm{Y}}{\mathrm{A} \rightarrow \mathrm{X}, \mathrm{Y}}$$

The right hand sides of the rules in this schema are unordered, and X and Y indicate any sets of categories. The idea behind Pullum's rule is made explicit in Zwicky's (1986b) reformulation:

$$\frac{\mathrm{A} \rightarrow \mathrm{B}, \mathrm{X} \quad \mathrm{B} \rightarrow \mathrm{Y}}{\mathrm{A} \rightarrow \mathrm{X}, \mathrm{Y}}$$

Usually, the combination of the two premises $\mathrm{A} \rightarrow \mathrm{B}, \mathrm{X}$ and $\mathrm{B} \rightarrow \mathrm{Y}$ would yield a nested structure: $[_{\mathrm{A}}\ [_{\mathrm{B}}\ Y]\ X]$. Liberation erases the constituent of category B, allowing the categories occurring in it to mingle with their aunt categories. Clearly, liberation is an operation very close to the version of domain unioning in which the two components of domain structure and linear precedence have been separated.

The above formulation of liberation only allows it as a global option. Also, since it is just an additional rule, flat and structured analyses will always be

present next to each other. To overcome these two problems, Zwicky (1986a) proposed to formulate liberation as a primitive syntactic operation, expressed as a triple $\langle M, C, L \rangle$, where:

- ▷ M is the mother category;
- ▷ C is the set of daughter categories that are concatenated;[2]
- ▷ L is the set of daughter categories that are liberated;

This rule is called *direct liberation* by Zwicky, as opposed to the *indirect liberation* expressed by the other two formulations. Again, the connection with word order domains is apparent: liberation is the counterpart of domain unioning, concatenation that of domain insertion. A difference is that Zwicky doesn't assume any functor-argument structure, so that in particular he can't have an assumption to the effect that functors are always inserted into their mother's domain.

Hoeksema (1991) introduces a categorial grammar that is modeled quite closely on the notion of direct liberation. He adds to a basic categorial grammar a type forming operation [.], so that [A] is a type whenever A is. The function of this operator is to indicate categories that are to be liberated. This means that next to a type such as B/A there will be three other ones: B/[A], [B]/A and [B]/[A]. These four categories are the respective analogs of $\Box B/\Diamond A$, $\Box B/A$, $B/\Diamond A$, B/A in my system. Hoeksema adds a number of ad hoc extensions to this basic system in order to allow him to give a categorial version of a complex predicate analysis of constructions such as *consider June a genius*, which he claims to be superior to the rivalling small clause analysis.

Dowty (to appear) takes a different approach, which is closer to mine. He starts with a categorial grammar in which unordered merging is the default syntactic operation. For each language, a list of *bounding categories* needs to be specified. Parts of expressions of these catgories can't mingle with expressions outside the bounding category and vice-versa. In other words, bounding categories are the equivalent of types projecting domains in my set up. Dowty, like Hoeksema, extends his basic system with several extensions in order to reach descriptive adequacy for the fragment that is to be accounted for, which in his case concerns linear order in English verb phrases.

Comparing the above proposals with the categorial account of word order domains presented in this chapter, it can be said that the major difference is the fact that what is presented here is a fully logical account, as opposed to the more ad hoc systems of Hoeksema and Dowty. This holds both for the basic structure laid out in the previous two sections, but also for the account of LP constraint checking that is to follow. Additional machinery will need to be added to get this to work, but it is all done within well-understood systems, for which for example soundness and completeness results are available.

[2] "Concatenated" is a somewhat unfortunate term in that it strongly suggests an *ordered* setting, whereas in fact we are dealing with unordered ID rules.

8.4 Linear order

In this section I will show how the categorial grammars defined in the first two sections can be extended in such a way that they also take care of the enforcement of LP constraints.

I will take the usual route of expressing linear order in terms of a directed product, which is added to this calculus. This means that the starting point for this type logical account of LP constraint checking is a calculus which has both a directed and an undirected product (both of which are associative), together with a set of modalities $\bigcup_{x \in \mathcal{L}} \{\langle x \rangle, [x]\}$. Here, $\langle \mathcal{L}, \prec \rangle$ is a partially ordered set of labels, the partial order expressing the LP constraints. What needs to be added to this basic calculus are postulates providing a way of replacing the non-directed mode by the directed mode. This must be possible iff the linear order of the two subtypes in the latter type is consistent with the LP constraints as expressed by the partial order on $\mathcal{L}$.

In Reape's theory, LP constraints are a *global* property of the grammar, and they need to be satisfied at every level. This is reminiscent of the ECPO (Exhaustive Constant Partial Order) property of generalized phrase structure grammars, which states that "the set of expansions of any one category observes a partial ordering that is also observed by the expansions of all other categories." (Gazdar *et al.* (1985), p. 49). It has been shown that the ECPO is descriptively inadequate, cf. **??**. For that reason, I am reluctant to follow Reape in adopting a single LP constraint regime for the entire grammar.

Note that the way Reape's grammar is set up, a global LP constraint checking regime is almost obligatory. Indeed, the redundancy that I observed earlier in his assumptions (5b) and (6) could easily turn into a contradiction in the case of local LP constraints: the constraints that are imposed on a daughter domain are required to be respected in the mother domain, but the constraints operating on the mother domain itself may well conflict with them.

No such problem arises in the categorial version of word order domains, thanks to its separation of domain structure and linear order. We will see below that a local approach to LP constraints is in fact quite easily added to these categorial systems. The generalization that was expressed a little too firmly by the ECPO is then incorporated into the system by having a particular type of LP constraint checking as the default option. Another advantage of this approach is that it allows the functor to impose LP constraints on a domain, even in the case the domain structure was projected by the argument rather than the functor. This further separates domain structure from linear order.

Since the grammar fragment to be developed here will address essentially the same data as Reape's system, no exceptions to this default modalization will in fact need to be made. This means that the single case of LP constraint checking mentioned at the beginning of this section, expressed in terms of a partially orderd set of labels $\mathcal{L}$, is the only one that I will consider below.

How, then, can the correct transition from a non-directed to a directed product be mediated by the labeling as represented in the modalities? Note that the direction implicit in this way of phrasing the question pertains to a view of sequent derivations that starts at the axioms. The direction of type

transitions corresponds to the converse view, from the bottom of a sequent derivation up, in the sense that , $[\mathsf{A}] \vdash \mathsf{C}$ is derivable (by Cut) from $\mathsf{A} \vdash \mathsf{B}$ and , $[\mathsf{B}] \vdash \mathsf{C}$.

A first approximation to the solution is that we simply allow the mode of combination of two labeled types to change from non-directed to directed provided the resulting ordering of the labels is consistent with the LP constraints. This would look as follows:

$$\langle x\rangle\mathsf{A} \bullet \langle y\rangle\mathsf{B} \vdash \langle x\rangle\mathsf{A} \otimes \langle y\rangle\mathsf{B} \qquad \frac{,\ [\Delta^x, \Theta^y] \vdash \mathsf{A}}{,\ [\Delta^x\ \Theta^y] \vdash \mathsf{A}} \quad (y \not\prec x)$$

In these rules, I have used the following notational convention.

Abbreviation 3 The structural connective corresponding to the directed product is written as juxtaposition. That is, instead of for instance $(\mathsf{A}, \mathsf{A}\backslash\mathsf{B}) \vdash \mathsf{B}$ I will write $\mathsf{A}\ \mathsf{A}\backslash\mathsf{B} \vdash \mathsf{B}$.

The problem with the above idea is that it only works for ordering two labeled items, but doesn't extend beyond that. For instance, it will never allow us to combine $\langle x\rangle\mathsf{A} \bullet \langle y\rangle\mathsf{B}$ with a third type $\langle z\rangle\mathsf{C}$, because no information is available on the top level about the material constituting the former type.

To solve this problem, a third set of auxiliary modalities is introduced, each of which represents a certain part of a domain by indicating which labels are admissible for the expressions occurring in that part of the domain. This means that we will have modalities $\langle S\rangle$ and $[S]$ for all $S \subseteq \mathcal{L}$. The transition from an unordered domain to an ordered one can be divided up into three steps:

1. The labeling modalities copy their information into a subset modality.
2. Material is ordered in compliance with the information expressed by the subset modalities. Furthermore, this information is moved from the original expressions to the combined expression.
3. Once all material in a domain has been ordered, the surrounding subset modality is removed.

The postulate that expresses the first of these three steps is straightforward:

$$\langle\{x\}\rangle\langle x\rangle\mathsf{A} \vdash \langle x\rangle\mathsf{A} \qquad \frac{,\ [\Delta^x] \vdash \mathsf{A}}{,\ [(\Delta^x)^{\{x\}}] \vdash \mathsf{A}}\ [\in]$$

For the second step, consider material Δ^S combined with material Θ^T in the non-directed mode. Then Δ can be ordered before Θ if this doesn't result in a violation of the LP constraints. That is, there must not be labels $s \in S$ and $t \in T$ such that $t \prec s$. I will write $S \lhd T$ for this condition on sets, i.e. $S \lhd T$ iff $\forall s \in S, t \in T : t \not\prec s$. The rule for the second step can now be stated:

$$\langle S\rangle\mathsf{A} \otimes \langle T\rangle\mathsf{B} \vdash \langle S \cup T\rangle(\mathsf{A} \bullet \mathsf{B}) \qquad \frac{,\ [\Delta^S, \Theta^T] \vdash \mathsf{A}}{,\ [(\Delta\ \Theta)^{S\cup T}] \vdash \mathsf{A}}\ [\mathrm{LP}] \quad (S \lhd T)$$

The final step concerns 'getting rid' of a subset modality that marks the material constituting an entire domain, which must be possible just in case the

domain needs to be checked for linear order. This is perhaps best made clear by means of an example. Suppose that $\mathcal{L} = \{x, y, z\}$ with the following partial order: $x, y \prec z$. Assume furthermore that the following sequent is derivable:

$$\langle x\rangle \mathsf{A}, \langle y\rangle \mathsf{B}, \langle x\rangle \mathsf{C}, \langle z\rangle \mathsf{D} \vdash \mathsf{E}$$

Then, looking at a sequent derivation bottom-up, the labeling diamonds on the individual types can spawn subset diamonds which then move to the top level by combining with each other, yielding the following sequent higher in the derivation:

$$(\langle x\rangle \mathsf{A}, \langle y\rangle \mathsf{B}, \langle x\rangle \mathsf{C}, \langle z\rangle \mathsf{D})^{\{x,y,z\}} \vdash \mathsf{E}$$

In case the E-domain does indeed need to be checked against the LP constraints, the type E must be decorated in such a way that it can impose the structure $(.)^{\{x,y,z\}}$ on the antecedent. Clearly, this can be achieved by prefixing E with $[\{x, y, z\}]$. This type assignment is not general enough, though, since it is based on a particular composition of the domain. In other cases there may be more material in it, or less. The point is that it doesn't really matter exactly which subset modality is present at the top level, since the presence of *any* of these modalities there already implies that the domain under it has been ordered. This can be expressed elegantly if the following postulate is added to the calculus:[3]

$$\langle S\rangle \mathsf{A} \vdash \langle T\rangle \mathsf{A} \qquad \frac{, [\Delta^S] \vdash \mathsf{A}}{, [\Delta^T] \vdash \mathsf{A}}\ [\subseteq] \quad (S \subseteq T)$$

Note that for the LP constraint checking process itself this postulate is harmless in the sense that it allows a subset modality to suggest that more kinds of material are present underneath it than is actually the case. It can therefore never cause underivability that is not due to a violation of the LP constraints. On the other hand, since it is of course in no way required to use this postulate, nothing that was derivable first becomes underivable now. With the above postulate present, the fact that certain material needs to be checked against the LP constraints can now simply be expressed by prefixing the corresponding type with the modality $[\mathcal{L}]$.

Let me illustrate the above rules by giving the derivation of the example

[3] Dörre and Manandhar (1995) consider what they call *constraint-based* Lambek calculi, in which the set $\mathcal{B}$ of basic types is assumed to come with some subtype ordering $\preceq$. A basic assumption underlying their systems is that $\mathsf{p} \vdash \mathsf{q}$ if $\mathsf{p} \preceq \mathsf{q}$. This approach can be carried over to modalities by exploiting the results of Section 5.3, which roughly state that $\Diamond\mathsf{A}$ can be read as $\mathsf{t}_\Diamond \bullet' \mathsf{A}$. Here t is a fresh basic type corresponding to the modality $\Diamond$, and $\bullet'$ is an additional product operator introduced specially for the translation. In the case of labeled modalities, this yields a whole set of fresh basic types, to which Dörre and Manandhar's ideas are applicable. In the case of subsets, we end up with a basic type for each subset, with $\subseteq$ playing the role of $\preceq$. The postulate in the text is then a basic element of the calculus. My reason for not adopting this elegant approach is that it only works for part of the system I'm proposing, so that it's incorporation would actually complicate things.

sequent mentioned above:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\langle x\rangle \mathsf{A}, \langle y\rangle \mathsf{B}, \langle x\rangle \mathsf{C}, \langle z\rangle \mathsf{D} \vdash \mathsf{E}}{(\langle x\rangle \mathsf{A})^{\{x\}}, \langle y\rangle \mathsf{B}, \langle x\rangle \mathsf{C}, \langle z\rangle \mathsf{D} \vdash \mathsf{E}}\,[\in]}{(\langle x\rangle \mathsf{A})^{\{x\}}, (\langle y\rangle \mathsf{B})^{\{y\}}, \langle x\rangle \mathsf{C}, \langle z\rangle \mathsf{D} \vdash \mathsf{E}}\,[\in]}{(\langle x\rangle \mathsf{A}\ \langle y\rangle \mathsf{B})^{\{x,y\}}, \langle x\rangle \mathsf{C}, \langle z\rangle \mathsf{D} \vdash \mathsf{E}}\,[\mathrm{LP}]}{(\langle x\rangle \mathsf{A}\ \langle y\rangle \mathsf{B})^{\{x,y\}}, (\langle x\rangle \mathsf{C})^{\{x\}}, \langle z\rangle \mathsf{D} \vdash \mathsf{E}}\,[\in]}{(\langle x\rangle \mathsf{A}\ \langle y\rangle \mathsf{B}\ \langle x\rangle \mathsf{C})^{\{x,y\}}, \langle z\rangle \mathsf{D} \vdash \mathsf{E}}\,[\mathrm{LP}]}{(\langle x\rangle \mathsf{A}\ \langle y\rangle \mathsf{B}\ \langle x\rangle \mathsf{C})^{\{x,y\}}, (\langle z\rangle \mathsf{D})^{\{z\}} \vdash \mathsf{E}}\,[\in]}{(\langle x\rangle \mathsf{A}\ \langle y\rangle \mathsf{B}\ \langle x\rangle \mathsf{C}\ \langle z\rangle \mathsf{D})^{\{x,y,z\}} \vdash \mathsf{E}}\,[\mathrm{LP}]}{(\langle x\rangle \mathsf{A}\ \langle y\rangle \mathsf{B}\ \langle x\rangle \mathsf{C}\ \langle z\rangle \mathsf{D})^{\mathcal{L}} \vdash \mathsf{E}}\,[\mathrm{Def}]}{\langle x\rangle \mathsf{A}\ \langle y\rangle \mathsf{B}\ \langle x\rangle \mathsf{C}\ \langle z\rangle \mathsf{D} \vdash [\mathcal{L}]\mathsf{E}}\,[[\mathcal{L}]\mathrm{R}]$$

One final issue needs to be resolved, which is the exact lexical type assignments for the types that trigger LP constraint checking. As I remarked before, I will let the functor specify that linear order needs to be imposed on a domain, regardless of whether the domain structure was projected by the functor itself or by the argument. The case where the functor projects the domain as well is the simplest one: just change $\langle x\rangle \mathsf{A} \to \mathsf{B}$ to $\langle x\rangle[\mathcal{L}]\mathsf{A} \to \mathsf{B}$. The case where the argument projects the domain is a little more intricate. Suppose we have a functor $\mathsf{A} \to \mathsf{B}$, and the argument A projects an x domain. What the modal decoration of the A in the functor needs to achieve then, is that the $[\mathcal{L}]$ modality can penetrate the x domain, so to speak. This can be achieved by changing $\mathsf{A} \to \mathsf{B}$ to $\langle x\rangle[\mathcal{L}][x]\mathsf{A} \to \mathsf{B}$. This way, the domain structure is first removed, and then the LP constraints are checked, after which the domain structure is reinstated. Since this mechanism will be used frequently, I introduce a special notation for the modal sequence involved in it:

Abbreviation 4 $\langle\!\langle x\rangle\!\rangle = \langle x\rangle[\mathcal{L}][x]$

This special notation is accompanied by the following shortening of sequent derivations:

Abbreviation 5

$$\dfrac{\dfrac{\dfrac{(,\,)^{\mathcal{L}} \vdash [x]\mathsf{A}}{,\ \vdash [\mathcal{L}][x]\mathsf{A}}\,[[\mathcal{L}]\mathrm{R}]}{(,\,)^{x} \vdash \langle x\rangle[\mathcal{L}][x]\mathsf{A}}\,[\langle x\rangle\mathrm{R}]}{(,\,)^{x} \vdash \langle\!\langle x\rangle\!\rangle\mathsf{A}}\,[\mathrm{Def}] \quad \rightsquigarrow \quad \dfrac{(,\,)^{\mathcal{L}} \vdash [x]\mathsf{A}}{(,\,)^{x} \vdash \langle\!\langle x\rangle\!\rangle\mathsf{A}}\,[\langle\!\langle x\rangle\!\rangle\mathrm{R}]$$

In the derivation below, I will use another abbreviation, namely leaving out the trivial first or second premise of the $[\to\mathrm{L}]$ rule.

Abbreviation 6

$$\dfrac{\Delta \vdash \mathsf{A} \quad \mathsf{B} \vdash \mathsf{B}}{\Delta, \mathsf{A} \to \mathsf{B} \vdash \mathsf{B}}\,[\to\mathrm{L}] \quad \rightsquigarrow \quad \dfrac{\Delta \vdash \mathsf{A}}{,\,[\Delta, \mathsf{A} \to \mathsf{B}] \vdash \mathsf{B}}\,[\to\mathrm{L}^1]$$

$$\dfrac{\mathsf{A} \vdash \mathsf{A} \quad ,\,[\mathsf{B}] \vdash \mathsf{C}}{,\,[\mathsf{A}, \mathsf{A} \to \mathsf{B}] \vdash \mathsf{C}}\,[\to\mathrm{L}] \quad \rightsquigarrow \quad \dfrac{,\,[\mathsf{B}] \vdash \mathsf{C}}{,\,[\mathsf{A}, \mathsf{A} \to \mathsf{B}] \vdash \mathsf{C}}\,[\to\mathrm{L}^1]$$

With these abbreviations, all the necessary elements have been introduced to give a derivation of an ordered sequent corresponding to the example sentence. Note that because of the way we have set things up, the goal type needs to be changed to $(\!(s)\!)\mathsf{S}$.

$$\begin{aligned} \textit{de}: &\quad (det)(\mathsf{N} \to [np]\mathsf{NP}) \\ \textit{nijlpaarden}: &\quad (n)\mathsf{N} \\ \textit{eten}: &\quad (v)\big((\!(np)\!)\mathsf{NP} \to [s]\mathsf{S}\big) \end{aligned}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{N} \vdash \mathsf{N}}{\mathsf{N} \to [np]\mathsf{NP}, \mathsf{N} \vdash [np]\mathsf{NP}}\,[\to\mathrm{L}^1]}{(det)(\mathsf{N} \to [np]\mathsf{NP}), (n)\mathsf{N} \vdash [np]\mathsf{NP}}\,[(det)\mathrm{L},(n)\mathrm{L}]}{((det)(\mathsf{N} \to [np]\mathsf{NP}))^{\{det\}}, ((n)\mathsf{N})^{\{n\}} \vdash [np]\mathsf{NP}}\,[\in\ (2\mathsf{x})]}{((det)(\mathsf{N} \to [np]\mathsf{NP})\ (n)\mathsf{N})^{\{det,n\}} \vdash [np]\mathsf{NP}}\,[\mathrm{LP}]}{((det)(\mathsf{N} \to [np]\mathsf{NP})\ (n)\mathsf{N})^{\mathcal{L}} \vdash [np]\mathsf{NP}}\,[\subseteq]}{(\textit{de nijlpaarden})^{\mathcal{L}} \vdash [np]\mathsf{NP}}\,[\mathrm{Lex}]}{(\textit{de nijlpaarden})^{np} \vdash (\!(np)\!)\mathsf{NP}}\,[(\!(np)\!)\mathrm{R}]}{(\textit{de nijlpaarden})^{np}, (\!(np)\!)\mathsf{NP} \to [s]\mathsf{S} \vdash [s]\mathsf{S}}\,[\to\mathrm{L}^1]}{(\textit{de nijlpaarden})^{np}, (v)\big((\!(np)\!)\mathsf{NP} \to [s]\mathsf{S}\big) \vdash [s]\mathsf{S}}\,[(v)\mathrm{L}]}{(\textit{de nijlpaarden})^{np}, ((v)\big((\!(np)\!)\mathsf{NP} \to [s]\mathsf{S}\big))^{\{v\}} \vdash [s]\mathsf{S}}\,[\in]}{(\textit{de nijlpaarden})^{np}, \textit{eten}^{\{v\}} \vdash [s]\mathsf{S}}\,[\mathrm{Lex}]}{((\textit{de nijlpaarden})^{np})^{\{np\}}, \textit{eten}^{\{v\}} \vdash [s]\mathsf{S}}\,[\in]}{((\textit{de nijlpaarden})^{np}\ \textit{eten})^{\{np,v\}} \vdash [s]\mathsf{S}}\,[\mathrm{LP}]}{((\textit{de nijlpaarden})^{np}\ \textit{eten})^{\mathcal{L}} \vdash [s]\mathsf{S}}\,[\subseteq]}{((\textit{de nijlpaarden})^{np}\ \textit{eten})^{s} \vdash (\!(s)\!)\mathsf{S}}\,[(\!(s)\!)\mathrm{R}]$$

Another, related way of dealing with LP constraints in a multimodal setting was proposed by Kraak (1995a, 1995b). She is concerned with French object clitics, the order restrictions on which are expressed by the following schema:

$$\top \;\prec\; n \;\prec\; \left[\begin{array}{ccccc} \left[\begin{array}{c} 1, d \\ 2, d \end{array}\right] & \prec & 3, a & \prec & 3, d \\ & & \left[\begin{array}{c} 1, a \\ 2, a \end{array}\right] & & \end{array}\right] \;\prec\; \bot$$

Simplifying things somewhat, the LP constraints are expressed through labeled modalities, of which there is one for each entry in the above schema. There are two products, $\bullet_h$ and $\bullet_f$, both of them directed. The crucial inference schema is the following one:

$$\langle\varphi\rangle(\mathsf{A} \bullet_h \mathsf{B}) \vdash \langle\chi\rangle\mathsf{A} \bullet_f \langle\psi\rangle\mathsf{B} \qquad \varphi \prec \chi \preceq \psi$$

The rest of the system is set up in such a way that in this schema χ can be thought of as representing an actual occurrence of a certain clitic, whereas φ and ψ indicate that clitic descriptions 'upto and including' the label are

not allowed under them. The schema can then be explained as follows: since $\langle\varphi\rangle(\mathsf{A} \bullet_h \mathsf{B})$ can't have anything in it with a label that comes before or is equal to φ, its first element χ must be strictly bigger than φ, while the rest of the clitics must be ordered after χ, so that their label ψ needs to be equal to or greater than χ.

Kraak's proposal as it stands is less general than mine since it only allows movement along a *chain* (a linearly ordered subset) of the partial order. Therefore, it can't account for the interspersal of elements whose order with respect to each other is not constrained by $\prec$. This is no problem for the account of French object clitics, since both chains in the schema are indeed mutually exclusive.

8.5 Nesting depth

Even though all elements of the outline of Reape's theory of word order domains that was presented in Section 7.2.1.4 have now been accounted for, the multimodal machinery introduced so far doesn't yet allow us to get the verb-raising data right. The reason for this is that two additional parameters that are relevant to the LP constraints were not mentioned there. Indeed, of the LP constraints introduced so far only $np \prec v$ is directly relevant to verb-raising constructions. This constraint orders the noun phrases before the verbs, but it doesn't say anything about the linear order among either the noun phrases or the verbs. This linear order depends on the following two parameters:

1. Nesting depth: for both noun phrases and verbs it holds that the deeper they are nested, the further to the right they occur in the domain.

2. Obliqueness: if several noun phrases are nested equally deep, the more oblique ones occur further to the right.

Consider for instance the following verb-raising sentence, repeated from Section 7.1:

(1)e. *dat ik Henk Cecilia de nijlpaarden zie helpen voeren*

The **LP** type assignments for the lexical items in this sentence are these (where VP abbreviates $\mathsf{NP} \to \mathsf{S}$):

$$\begin{array}{rl} \textit{Henk}, \textit{Cecilia}, \textit{de nijlpaarden}: & \mathsf{NP} \\ \textit{zie}, \textit{helpen}: & \mathsf{VP} \to (\mathsf{NP} \to \mathsf{VP}) \\ \textit{voeren}: & \mathsf{NP} \to \mathsf{VP} \end{array}$$

Top level verbs will be considered to occur on level 0. If a verb occurs at level n, its nominal arguments occur on level n as well, while its verbal arguments occur on level $n+1$. The levels in the example sentence are as follows:

$$\textit{dat}\textit{ik}_0\ \textit{Henk}_0\ \textit{Cecilia}_1\ \textit{de nijlpaarden}_2\textit{zie}_0\ \textit{helpen}_1\ \textit{voeren}_2$$

The fact that the above linear order is the only grammatical one follows from the nesting levels, plus the fact that the subject *ik* of *zie* is less oblique than its object *Henk*.

Obliqueness can be accounted for by introducing two new labels, *nom* and *acc*. These will be used to differentiate the different NP arguments of a verb: the subject is labeled with *nom*, while a second NP argument (if present) is labeled with *acc*. Since these labels will only occur on NPs they can in fact be used to distinguish NPs from other items, so that a separate *np* label can be dispensed with.

In order to be able to indicate nesting depth, a special purpose modality $\Diamond$ is introduced. The idea is to have $\Diamond$ indicate a single level of nesting. Since the level of noun phrase arguments to verbs is considered the same as that of the verb itself, the nesting modality only needs to be added to the verbal arguments of verbs. For our example sentence this means that while the definitions of VP and TVP remain the same, a $\Diamond$ needs to be added to the VP argument of *zie* and *helpen*, so that their type becomes $\Diamond\mathsf{VP} \rightarrow \mathsf{TVP}$.

If a nesting modality surrounds a combination of types, the nesting depth of all these types increases by one. Note that if *helpen* combines with a verb phrase such as *de nijlpaarden voeren*, the nesting modality ends up surrounding both the embedded verb and the embedded noun phrase. Since nesting depths need to be indicated on each type separately, the nesting modality is allowed to spread over all types it surrounds by means of the following postulate:

$$\Diamond\mathsf{A} \otimes \Diamond\mathsf{B} \vdash \Diamond(\mathsf{A} \otimes \mathsf{B}) \qquad \frac{, [(\Delta, \Theta)^{\Diamond}] \vdash \mathsf{A}}{, [(\Delta)^{\Diamond}, (\Theta)^{\Diamond}] \vdash \mathsf{A}} \; [\Diamond\mathrm{K}]$$

Below, the following abbreviation will be used:

Abbreviation 7 In derivations, implicit use will occasionally be made of the inversion lemma for modalities:

$$, [(\mathsf{A})^{\Diamond}] \vdash \mathsf{B} \quad \text{iff} \quad , [\Diamond\mathsf{A}] \vdash \mathsf{B}$$
$$(,)^{\Diamond} \vdash \mathsf{B} \quad \text{iff} \quad , \vdash \Box\mathsf{B}$$

The following derivation shows how the nesting depths indicated in (1)e can now be interpreted as indicating the number of nesting modalities on an item:

$$
\begin{array}{c l}
\mathsf{NP}, \mathsf{VP} \vdash \mathsf{S} & \\ \hline
\mathit{ik}, \mathsf{VP} \vdash \mathsf{S} & [\mathrm{Lex}] \\ \hline
\mathit{ik}, \mathsf{NP}, \mathsf{TVP} \vdash \mathsf{S} & [\mathrm{Def}, \rightarrow\mathrm{L}^1] \\ \hline
\mathit{ik}, \mathit{Henk}, \mathsf{TVP} \vdash \mathsf{S} & [\mathrm{Lex}] \\ \hline
\mathit{ik}, \mathit{Henk}, \mathsf{VP}^{\Diamond}, \Diamond\mathsf{VP} \rightarrow \mathsf{TVP} \vdash \mathsf{S} & [\rightarrow\mathrm{L}^1] \\ \hline
\mathit{ik}, \mathit{Henk}, \mathsf{VP}^{\Diamond}, \mathit{zie} \vdash \mathsf{S} & [\mathrm{Lex}] \\ \hline
\mathit{ik}, \mathit{Henk}, (\mathsf{NP}, \mathsf{TVP})^{\Diamond}, \mathit{zie} \vdash \mathsf{S} & [\mathrm{Def}, \rightarrow\mathrm{L}^1] \\ \hline
\mathit{ik}, \mathit{Henk}, (\mathit{Cecilia}, \mathsf{TVP})^{\Diamond}, \mathit{zie} \vdash \mathsf{S} & [\mathrm{Lex}] \\ \hline
\mathit{ik}, \mathit{Henk}, \mathit{Cecilia}^{\Diamond}, \mathsf{TVP}^{\Diamond}, \mathit{zie} \vdash \mathsf{S} & [\Diamond\mathrm{K}] \\ \hline
\mathit{ik}, \mathit{Henk}, \mathit{Cecilia}^{\Diamond}, (\mathsf{VP}^{\Diamond}, \Diamond\mathsf{VP} \rightarrow \mathsf{TVP})^{\Diamond}, \mathit{zie} \vdash \mathsf{S} & [\rightarrow\mathrm{L}^1] \\ \hline
\mathit{ik}, \mathit{Henk}, \mathit{Cecilia}^{\Diamond}, (\mathsf{VP}^{\Diamond}, \mathit{helpen})^{\Diamond}, \mathit{zie} \vdash \mathsf{S} & [\mathrm{Lex}] \\ \hline
\mathit{ik}, \mathit{Henk}, \mathit{Cecilia}^{\Diamond}, \mathsf{VP}^{\Diamond\Diamond}, \mathit{helpen}^{\Diamond}, \mathit{zie} \vdash \mathsf{S} & [\Diamond\mathrm{K}] \\ \hline
\mathit{ik}, \mathit{Henk}, \mathit{Cecilia}^{\Diamond}, (\mathsf{NP}, \mathsf{TVP})^{\Diamond\Diamond}, \mathit{helpen}^{\Diamond}, \mathit{zie} \vdash \mathsf{S} & [\mathrm{Def}, \rightarrow\mathrm{L}^1] \\ \hline
\mathit{ik}, \mathit{Henk}, \mathit{Cecilia}^{\Diamond}, (\mathit{de\ nijlpaarden}, \mathit{voeren})^{\Diamond\Diamond}, \mathit{helpen}^{\Diamond}, \mathit{zie} \vdash \mathsf{S} & [\mathrm{Lex}\ (2\mathsf{x})] \\ \hline
\mathit{ik}, \mathit{Henk}, \mathit{Cecilia}^{\Diamond}, ((\mathit{de\ nijlpaarden})^{\Diamond}, \mathit{voeren}^{\Diamond})^{\Diamond}, \mathit{helpen}^{\Diamond}, \mathit{zie} \vdash \mathsf{S} & [\Diamond\mathrm{K}] \\ \hline
\mathit{ik}, \mathit{Henk}, \mathit{Cecilia}^{\Diamond}, \mathit{de\ nijlpaarden}^{\Diamond\Diamond}, \mathit{voeren}^{\Diamond\Diamond}, \mathit{helpen}^{\Diamond}, \mathit{zie} \vdash \mathsf{S} & [\Diamond\mathrm{K}] \\ \hline
\mathit{ik}, \mathit{Henk}, \mathit{Cecilia}^{\Diamond}, \mathit{de\ nijlpaarden}^{\Diamond\Diamond}, \mathit{zie}, \mathit{helpen}^{\Diamond}, \mathit{voeren}^{\Diamond\Diamond} \vdash \mathsf{S} & [\mathrm{Perm}] \\ \hline
\mathit{ik}_0, \mathit{Henk}_0, \mathit{Cecilia}_1, \mathit{de\ nijlpaarden}_2, \mathit{zie}_0, \mathit{helpen}_1, \mathit{voeren}_2 \vdash \mathsf{S} & [\mathrm{Def}] \\ \hline
\mathit{ik}, \mathit{Henk}, \mathit{Cecilia}, \mathit{de\ nijlpaarden}, \mathit{zie}, \mathit{helpen}, \mathit{voeren} \vdash \mathsf{S} &
\end{array}
$$

As a precursor to the explanation of how nesting depth can be used in the order checking process, I will first discuss a phenomenon mentioned in in Section 7.1, namely that the head of the verbal complement of auxiliaries may optionally be ordered at the beginning of the verb cluster. The idea is to account for this by having auxiliaries optionally mark their VP-complement with a new label $\langle pre \rangle$, which will be ordered right before all other verbs:

$$\mathit{hebben}: \qquad \mathsf{VP} \rightarrow \mathsf{VP}, \langle pre \rangle \mathsf{VP} \rightarrow \mathsf{VP}$$

However, it is not the entire VP complement that ends up in front of all the other verbs, but just its verbal head. The initial $\langle pre \rangle$-marking on this verbal head needs to be able to percolate upward until it encompasses the entire VP. This can be effectuated by the following postulate:

$$\mathsf{A} \otimes \langle pre \rangle \mathsf{B} \vdash \langle pre \rangle (\mathsf{A} \otimes \mathsf{B}) \qquad \frac{, [(\Delta, \Theta)^{pre}] \vdash \mathsf{A}}{, [\Delta, (\Theta)^{pre}] \vdash \mathsf{A}}\ [pre]$$

This postulate in itself isn't sufficient yet. The reason for this is that in order to influence the LP constraint checking the $\langle pre \rangle$-modality needs to surround all $\Diamond$-modalities, whereas the VP it needs to surround may be nested. A mechanism is therefore needed to let it descend to the right level. More in general it

can be said that information about the nesting depth of types is independent from other information about the types. This fact can be introduced into the grammar by means of the postulates below, where $\Diamond$ ranges over all labeling modalities:

$$\Diamond\Diamond \mathsf{A} \Leftrightarrow \Diamond\Diamond \mathsf{A} \qquad \dfrac{, [((\Delta)^{\Diamond})^{\Diamond}] \vdash \mathsf{A}}{, [((\Delta)^{\Diamond})^{\Diamond}] \vdash \mathsf{A}}\ [\Diamond\text{Perm}]$$

With these modifications, it is possible to have the $\langle pre \rangle$-marking on the verbal head of a VP-complement, as is borne out by the derivation below (where again case-modalities are omitted):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{NP}, \mathsf{VP} \vdash \mathsf{S}}{\mathit{Cecilia}, \mathsf{VP} \vdash \mathsf{S}}\ [\text{Lex}]}{\mathit{Cecilia}, \mathsf{VP}^{\Diamond}, \Diamond\mathsf{VP} \rightarrow \mathsf{VP} \vdash \mathsf{S}}\ [\rightarrow\text{L}^1]}{\mathit{Cecilia}, \mathsf{VP}^{\Diamond}, \mathit{moet} \vdash \mathsf{S}}\ [\text{Lex}]}{\mathit{Cecilia}, (\mathsf{VP}^{\Diamond \mathit{pre}}, \langle pre \rangle \Diamond \mathsf{VP} \rightarrow \mathsf{VP})^{\Diamond}, \mathit{moet} \vdash \mathsf{S}}\ [\rightarrow\text{L}^1]}{\mathit{Cecilia}, (\mathsf{VP}^{\Diamond \mathit{pre}}, \mathit{hebben})^{\Diamond}, \mathit{moet} \vdash \mathsf{S}}\ [\text{Lex}]}{\mathit{Cecilia}, \mathsf{VP}^{\Diamond \mathit{pre} \Diamond}, \mathit{hebben}^{\Diamond}, \mathit{moet} \vdash \mathsf{S}}\ [\Diamond\text{K}]}{\mathit{Cecilia}, \mathsf{VP}^{\Diamond \mathit{pre} \Diamond}, \mathit{moet}, \mathit{hebben}^{\Diamond} \vdash \mathsf{S}}\ [\text{Perm}]}{\mathit{Cecilia}, \mathsf{VP}^{\Diamond\Diamond \mathit{pre}}, \mathit{moet}, \mathit{hebben}^{\Diamond} \vdash \mathsf{S}}\ [\Diamond\text{Perm}]}{\mathit{Cecilia}, (\mathsf{NP}, \mathsf{TVP})^{\Diamond\Diamond \mathit{pre}}, \mathit{moet}, \mathit{hebben}^{\Diamond} \vdash \mathsf{S}}\ [\text{Def}, \rightarrow\text{L}^1]}{\mathit{Cecilia}, (\mathit{de\ nijlpaarden}, \mathit{gevoerd})^{\Diamond\Diamond \mathit{pre}}, \mathit{moet}, \mathit{hebben}^{\Diamond} \vdash \mathsf{S}}\ [\text{Lex}]}{\mathit{Cecilia}, ((\mathit{de\ nijlpaarden})^{\Diamond}, \mathit{gevoerd}^{\Diamond})^{\Diamond \mathit{pre}}, \mathit{moet}, \mathit{hebben}^{\Diamond} \vdash \mathsf{S}}\ [\Diamond\text{K}]}{\mathit{Cecilia}, ((\mathit{de\ nijlpaarden})^{\Diamond\Diamond}, \mathit{gevoerd}^{\Diamond\Diamond})^{\mathit{pre}}, \mathit{moet}, \mathit{hebben}^{\Diamond} \vdash \mathsf{S}}\ [\Diamond\text{K}]}{\mathit{Cecilia}, (\mathit{de\ nijlpaarden})^{\Diamond\Diamond}, \mathit{gevoerd}^{\Diamond\Diamond \mathit{pre}}, \mathit{moet}, \mathit{hebben}^{\Diamond} \vdash \mathsf{S}}\ [\mathit{pre}]$$

Nesting depth needs to be marked on noun phrases and verbs. In order to do this, I introduce the modalities $\langle v_i \rangle$, $\langle nom_i \rangle$ and $\langle acc_i \rangle$ $(i \geq 1)$. The fact that this set is infinite doesn't cause any major problems, as will become clear below. Due to the postulates introduced earlier, all combinations of $\langle f \rangle$ with i $\Diamond$-modalities are equivalent, for any $i \geq 0$ and $\langle f \rangle \in \{\langle nom \rangle, \langle acc \rangle, \langle v \rangle\}$. The newly added modalities are intended to represent such combinations, which is effectuated by the following postulates:

$$\langle f \rangle (\Diamond^i \mathsf{A}) \Leftrightarrow \langle f \rangle_i \mathsf{A} \qquad \dfrac{, [(\Delta)^{f_i}] \vdash \mathsf{A}}{, [((\Delta)^{f})^{\Diamond^i}] \vdash \mathsf{A}}\ [\Diamond^i]$$

The exact details of how these modalities participate in the LP constraint checking process can be found in the next section.

8.6 A verb raising fragment

At last, all preparations have been made that are needed to present a fully formalized multimodal categorial grammar for a simple verb raising fragment.

The grammar has two products, a directed and a non-directed one. Moreover, there are a number of labeled modalities:

1. $\langle nom \rangle$ and $\langle acc \rangle$ to indicate subject and non-subject noun phrases, respectively;
2. $\langle v \rangle$, $\langle s \rangle$, $\langle n \rangle$ and $\langle det \rangle$ to indicate verbs, sentences, nouns and determiners, respectively;
3. $\langle x \rangle$ to indicate extraposed constituents;
4. $\langle pre \rangle$ to indicate verbal complements whose head is placed at the beginning of the verb cluster;
5. $\Diamond$ to indicate nesting depth;
6. LP constraint checking modalities.

The set $\mathcal{L}$ of labels on which the LP constraints are defined is the following one:

$$\mathcal{L} = \{nom_0, nom_1, \ldots, acc_0, acc_1, \ldots, v_0, v_1, \ldots, det, n, s, pre, x\}$$

The partial order $\prec$ on $\mathcal{L}$ is the following one:

$$nom_0 \prec acc_0 \prec nom_1 \prec \ldots \prec pre \prec v_0 \prec v_1 \prec \ldots \prec x$$

$$det \prec n$$

In order to simplify things, I will ignore the internal structure of NPs in this section. This has a pleasant consequence: the labels *det* and n need no longer be considered, and the restriction of $\prec$ to the subset of remaining labels is in fact a *linear* order. This will allow us to use *intervals*[4] instead of arbitrary subsets, which will turn out to be particularly handy when combined with the Lemma below. It has the additional advantage that it solves the complications that arise from the fact that the set $\mathcal{L}$ is infinite.

Lemma 8.1 *If* $, [\Delta^S, \Theta^T] \vdash \mathsf{A}$, $S \lhd T$ *and* $S, T \subseteq U$, *then* $, [(\Delta\ \Theta)^U] \vdash \mathsf{A}$.

Proof This follows from the derivation below:

$$\dfrac{\dfrac{, [\Delta^S, \Theta^T] \vdash \mathsf{A}}{, [(\Delta\ \Theta)^{S \cup T}] \vdash \mathsf{A}}\ [\mathrm{LP}]}{, [(\Delta\ \Theta)^U] \vdash \mathsf{A}}\ [\subseteq]$$

□

This result will be used in the derivations that are to follow, and will be abbreviated thus:

[4] Recall that the interval $[x, y]$ is defined as $\{z \mid x \preceq z \preceq y\}$

Abbreviation 8

$$\dfrac{\dfrac{,\,[\Delta^S,\Theta^T]\vdash \mathsf{A}}{,\,[(\Delta^S\ \Theta^T)^{S\cup T}]\vdash \mathsf{A}}\,[\mathrm{LP}]}{,\,[(\Delta^S\ \Theta^T)^U]\vdash \mathsf{A}}\,[\subseteq] \quad\rightsquigarrow\quad \dfrac{,\,[\Delta^S,\Theta^T]\vdash \mathsf{A}}{,\,[(\Delta^S\ \Theta^T)^U]\vdash \mathsf{A}}\,[\mathrm{LP}'] \qquad (S\lhd T;\ S,T\subseteq U)$$

As I showed in the last section, the rules for the constraint checking modalities follow automatically from the specification of $\mathcal{L}$ and $\prec$, so I won't repeat them here. Next to these rules, the calculus contains the following modal postulates:

$$\mathsf{A}\otimes\langle pre\rangle\mathsf{B}\vdash\langle pre\rangle(\mathsf{A}\otimes\mathsf{B}) \qquad \dfrac{,\,[(\Delta,\Theta)^{pre}]\vdash \mathsf{A}}{,\,[\Delta,(\Theta)^{pre}]\vdash \mathsf{A}}$$

$$\Diamond\mathsf{A}\otimes\Diamond\mathsf{B}\vdash\Diamond(\mathsf{A}\otimes\mathsf{B}) \qquad \dfrac{,\,[(\Delta,\Theta)^{\Diamond}]\vdash \mathsf{A}}{,\,[(\Delta)^{\Diamond},(\Theta)^{\Diamond}]\vdash \mathsf{A}}\,[\Diamond\mathrm{K}]$$

$$\Diamond\Diamond\mathsf{A}\Leftrightarrow\Diamond\Diamond\mathsf{A} \qquad \dfrac{\dfrac{,\,[((\Delta)^{\Diamond})^{\Diamond}]\vdash \mathsf{A}}{}}{,\,[((\Delta)^{\Diamond})^{\Diamond}]\vdash \mathsf{A}} \qquad \Diamond\in\{nom,acc,v,pre,x\}$$

$$\langle f\rangle(\Diamond^i\mathsf{A})\Leftrightarrow\langle f\rangle_i\mathsf{A} \qquad \dfrac{\dfrac{,\,[(\Delta)^{f_i}]\vdash \mathsf{A}}{}}{,\,[((\Delta)^{f})^{\Diamond^i}]\vdash \mathsf{A}}\,[\Diamond^i] \qquad f\in\{nom,acc,v\}$$

The type assignments for the fragment are given below. Notice that nesting depth is not indicated on extraposed VP arguments. This would be the most principled thing to do, but it isn't necessary and would only complicate matters.

Cecilia, *Henk*, *de nijlpaarden*:	$(\!(nom)\!)\mathsf{NP},\ (\!(acc)\!)\mathsf{NP}$
lopen:	$(v)\mathsf{VP} = (v)\big((\!(nom)\!)\mathsf{NP}\rightarrow[s]\mathsf{S}\big)$
kussen, *voeren*:	$(v)\mathsf{TVP} = (v)\big((\!(acc)\!)\mathsf{NP}\rightarrow\mathsf{VP}\big)$
schijnen:	$(v)\big(\Diamond\mathsf{VP}\rightarrow\mathsf{VP}\big)$
zien, *helpen*, *laten*:	$(v)\big(\Diamond\mathsf{VP}\rightarrow\mathsf{TVP}\big)$
hebben:	$(v)\big(\Diamond\mathsf{VP}\rightarrow\mathsf{VP}\big),(v)\big(\langle pre\rangle\mathsf{VP}\rightarrow\mathsf{VP}\big)$
afspreken:	$(v)\big(\langle x\rangle\langle\mathcal{L}\rangle\mathsf{VP}\rightarrow\mathsf{VP}\big)$
beloven, *voorhouden*:	$(v)\big(\langle x\rangle\langle\mathcal{L}\rangle\mathsf{VP}\rightarrow\mathsf{TVP}\big)$
proberen, *menen*:	$(v)\big(\Diamond\mathsf{VP}\rightarrow\mathsf{VP}\big),(v)\big(\langle x\rangle\langle\mathcal{L}\rangle\mathsf{VP}\rightarrow\mathsf{VP}\big)$
verplichten:	$(v)\big(\Diamond\mathsf{VP}\rightarrow\mathsf{TVP}\big),(v)\big(\langle x\rangle\langle\mathcal{L}\rangle\mathsf{VP}\rightarrow\mathsf{TVP}\big)$

I end this section with some illustrative derivations of the system, broken up into pieces for clarity. First, consider again example (1)c from Section 7.1:

(*dat*) *Henk Cecilia de nijlpaarden helpt voeren*

Viewing the derivation bottom-up, we can start by licensing the ordering of *Henk* at the beginning of the sentence (the two steps at the top aren't strictly

necessary, their purpose is just to reintroduce the abbreviation $(\!(nom)\!)$ in order to enhance legibility):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\!(nom)\!)\mathsf{NP},\ (\mathit{Cecilia}\ (\mathit{de\ nijlpaarden})^{\Diamond}\ \mathit{helpt\ voeren}^{\Diamond})^{[acc_0,x]} \vdash [s]\mathsf{S}}{\langle nom\rangle[\mathcal{L}][nom]\mathsf{NP},\ (\mathit{Cecilia}\ (\mathit{de\ nijlpaarden})^{\Diamond}\ \mathit{helpt\ voeren}^{\Diamond})^{[acc_0,x]} \vdash [s]\mathsf{S}}\ [\mathrm{Def}]}{\langle nom_0\rangle[\mathcal{L}][nom]\mathsf{NP},\ (\mathit{Cecilia}\ (\mathit{de\ nijlpaarden})^{\Diamond}\ \mathit{helpt\ voeren}^{\Diamond})^{[acc_0,x]} \vdash [s]\mathsf{S}}\ [\Diamond^0]}{(\langle nom_0\rangle[\mathcal{L}][nom]\mathsf{NP})^{\{nom_0\}},\ (\mathit{Cecilia}\ (\mathit{de\ nijlpaarden})^{\Diamond}\ \mathit{helpt\ voeren}^{\Diamond})^{[acc_0,x]} \vdash [s]\mathsf{S}}\ [\in]}{(\langle nom\rangle[\mathcal{L}][nom]\mathsf{NP})^{\{nom_0\}},\ (\mathit{Cecilia}\ (\mathit{de\ nijlpaarden})^{\Diamond}\ \mathit{helpt\ voeren}^{\Diamond})^{[acc_0,x]} \vdash [s]\mathsf{S}}\ [\Diamond^0]}{((\!(nom)\!)\mathsf{NP})^{\{nom_0\}},\ (\mathit{Cecilia}\ (\mathit{de\ nijlpaarden})^{\Diamond}\ \mathit{helpt\ voeren}^{\Diamond})^{[acc_0,x]} \vdash [s]\mathsf{S}}\ [\mathrm{Def}]}{\mathit{Henk}^{\{nom_0\}},\ (\mathit{Cecilia}\ (\mathit{de\ nijlpaarden})^{\Diamond}\ \mathit{helpt\ voeren}^{\Diamond})^{[acc_0,x]} \vdash [s]\mathsf{S}}\ [\mathrm{Lex}]}{(\mathit{Henk\ Cecilia}\ (\mathit{de\ nijlpaarden})^{\Diamond}\ \mathit{helpt\ voeren}^{\Diamond})^{\mathcal{L}} \vdash [s]\mathsf{S}}\ [\mathrm{LP}']}{(\mathit{Henk\ Cecilia}\ (\mathit{de\ nijlpaarden})^{\Diamond}\ \mathit{helpt\ voeren}^{\Diamond})^{s} \vdash (\!(s)\!)\mathsf{S}}\ [(\!(s)\!)\mathrm{R}]$$

Next, a similar thing can be done for *Cecilia*, which yields:

$$(\!(nom)\!)\mathsf{NP},\ (\!(acc)\!)\mathsf{NP},\ ((\mathit{de\ nijlpaarden})^{\Diamond}\ \mathit{helpt\ voeren}^{\Diamond})^{[nom_1,x]} \vdash [s]\mathsf{S}$$

For *de nijlpaarden* the procedure is again similar, albeit that the fact that this NP is nested complicates matters somewhat:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\!(nom)\!)\mathsf{NP},\ (\!(acc)\!)\mathsf{NP},\ (\!(acc)\!)\mathsf{NP}^{\Diamond},\ (\mathit{helpt\ voeren}^{\Diamond})^{[nom_2,x]} \vdash [s]\mathsf{S}}{(\!(nom)\!)\mathsf{NP},\ (\!(acc)\!)\mathsf{NP},\ [\mathcal{L}][acc]\mathsf{NP}^{acc_1},\ (\mathit{helpt\ voeren}^{\Diamond})^{[nom_2,x]} \vdash [s]\mathsf{S}}\ [\mathrm{Def}]}{(\!(nom)\!)\mathsf{NP},\ (\!(acc)\!)\mathsf{NP},\ [\mathcal{L}][acc]\mathsf{NP}^{acc_1},\ (\mathit{helpt\ voeren}^{\Diamond})^{[nom_2,x]} \vdash [s]\mathsf{S}}\ [\Diamond^1]}{(\!(nom)\!)\mathsf{NP},\ (\!(acc)\!)\mathsf{NP},\ [\mathcal{L}][acc]\mathsf{NP}^{acc_1\{acc_1\}},\ (\mathit{helpt\ voeren}^{\Diamond})^{[nom_2,x]} \vdash [s]\mathsf{S}}\ [\in]}{(\!(nom)\!)\mathsf{NP},\ (\!(acc)\!)\mathsf{NP},\ \langle acc\rangle[\mathcal{L}][acc]\mathsf{NP}^{\Diamond\{acc_1\}},\ (\mathit{helpt\ voeren}^{\Diamond})^{[nom_2,x]} \vdash [s]\mathsf{S}}\ [\Diamond^1]}{(\!(nom)\!)\mathsf{NP},\ (\!(acc)\!)\mathsf{NP},\ (\!(acc)\!)\mathsf{NP}^{\Diamond\{acc_1\}},\ (\mathit{helpt\ voeren}^{\Diamond})^{[nom_2,x]} \vdash [s]\mathsf{S}}\ [\mathrm{Def}]}{(\!(nom)\!)\mathsf{NP},\ (\!(acc)\!)\mathsf{NP},\ (\mathit{de\ nijlpaarden})^{\Diamond\{acc_1\}},\ (\mathit{helpt\ voeren}^{\Diamond})^{[nom_2,x]} \vdash [s]\mathsf{S}}\ [\mathrm{Lex}]}{(\!(nom)\!)\mathsf{NP},\ (\!(acc)\!)\mathsf{NP},\ ((\mathit{de\ nijlpaarden})^{\Diamond}\ \mathit{helpt\ voeren}^{\Diamond})^{[nom_1,x]} \vdash [s]\mathsf{S}}\ [\mathrm{LP}']$$

After the position of the verbs has been checked in this way as well, we are left with the following sequent:

$$(\!(nom)\!)\mathsf{NP},\ (\!(acc)\!)\mathsf{NP},\ (\!(acc)\!)\mathsf{NP}^{\Diamond},\ (v)(\Diamond\mathsf{VP}\rightarrow\mathsf{TVP}),\ (v)\mathsf{TVP}^{\Diamond} \vdash [s]\mathsf{S}$$

The derivability of this sequent is demonstrated by the derivation below:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{VP}\vdash\mathsf{VP}}{(\!(acc)\!)\mathsf{NP},(\!(acc)\!)\mathsf{NP}\rightarrow\mathsf{VP}\vdash\mathsf{VP}}\ [\rightarrow\mathrm{L}^1]}{(\!(acc)\!)\mathsf{NP},\mathsf{TVP}\vdash\mathsf{VP}}\ [\mathrm{Def}]}{(\!(acc)\!)\mathsf{NP},(v)\mathsf{TVP}\vdash\mathsf{VP}}\ [(v)\mathrm{L}]}{((\!(acc)\!)\mathsf{NP},(v)\mathsf{TVP})^{\Diamond}\vdash\Diamond\mathsf{VP}}\ [\Diamond\mathrm{R}]}{(\!(acc)\!)\mathsf{NP}^{\Diamond},(v)\mathsf{TVP}^{\Diamond}\vdash\Diamond\mathsf{VP}}\ [\Diamond\mathrm{K}]$$

$$\dfrac{\dfrac{\dfrac{\dfrac{[s]\mathsf{S}\vdash[s]\mathsf{S}}{(\!(nom)\!)\mathsf{NP},(\!(nom)\!)\mathsf{NP}\rightarrow[s]\mathsf{S}\vdash[s]\mathsf{S}}\ [\rightarrow\mathrm{L}^1]}{(\!(nom)\!)\mathsf{NP},\mathsf{VP}\vdash[s]\mathsf{S}}\ [\mathrm{Def}]}{(\!(nom)\!)\mathsf{NP},(\!(acc)\!)\mathsf{NP},(\!(acc)\!)\mathsf{NP}\rightarrow\mathsf{VP}\vdash[s]\mathsf{S}}\ [\rightarrow\mathrm{L}^1]}{(\!(nom)\!)\mathsf{NP},(\!(acc)\!)\mathsf{NP},\mathsf{TVP}\vdash[s]\mathsf{S}}\ [\mathrm{Def}]$$

$$\dfrac{(\!(nom)\!)\mathsf{NP},(\!(acc)\!)\mathsf{NP},(\!(acc)\!)\mathsf{NP}^{\Diamond},\Diamond\mathsf{VP}\rightarrow\mathsf{TVP},(v)\mathsf{TVP}^{\Diamond}\vdash[s]\mathsf{S}}{(\!(nom)\!)\mathsf{NP},(\!(acc)\!)\mathsf{NP},(\!(acc)\!)\mathsf{NP}^{\Diamond},(v)(\Diamond\mathsf{VP}\rightarrow\mathsf{TVP}),(v)\mathsf{TVP}^{\Diamond}\vdash[s]\mathsf{S}}\ [(v)\mathrm{L}]$$

The second example derivation is for a sentence in which extraposition occurs:

$$\dfrac{\dfrac{\dfrac{\vdots}{(\textit{de nijlpaarden te voeren})^{\mathcal{L}} \vdash \mathsf{VP}}}{\textit{de nijlpaarden te voeren} \vdash [\mathcal{L}]\mathsf{VP}}\,[[\mathcal{L}]\mathrm{R}]}{(\textit{de nijlpaarden te voeren})^{x} \vdash \langle x\rangle[\mathcal{L}]\mathsf{VP}}\,[\langle x\rangle\mathrm{R}] \qquad \dfrac{\vdots}{(\!(nom)\!)\mathsf{NP}, \mathsf{VP} \vdash [s]\mathsf{S}}$$

$$\dfrac{\cdots \qquad \cdots}{(\!(nom)\!)\mathsf{NP},\ \langle x\rangle[\mathcal{L}]\mathsf{VP} \to \mathsf{VP},\ (\textit{de nijlpaarden te voeren})^{x} \vdash [s]\mathsf{S}}\,[\to\mathrm{L}]$$

$$\dfrac{}{(\!(nom)\!)\mathsf{NP},\ (v)(\langle x\rangle[\mathcal{L}]\mathsf{VP} \to \mathsf{VP}),\ (\textit{de nijlpaarden te voeren})^{x} \vdash [s]\mathsf{S}}\,[(v)\mathrm{L}]$$

$$\dfrac{}{(\!(nom)\!)\mathsf{NP},\ (v)(\langle x\rangle[\mathcal{L}]\mathsf{VP} \to \mathsf{VP})^{\{v\}},\ (\textit{de nijlpaarden te voeren})^{x} \vdash [s]\mathsf{S}}\,[\in]$$

$$\dfrac{}{(\!(nom)\!)\mathsf{NP},\ \textit{afspreekt}^{\{v\}},\ (\textit{de nijlpaarden te voeren})^{x} \vdash [s]\mathsf{S}}\,[\mathrm{Lex}]$$

$$\dfrac{}{(\!(nom)\!)\mathsf{NP},\ \textit{afspreekt}^{\{v\}},\ (\textit{de nijlpaarden te voeren})^{x\{x\}} \vdash [s]\mathsf{S}}\,[\in]$$

$$\dfrac{}{(\!(nom)\!)\mathsf{NP},\ (\textit{afspreekt}\ (\textit{de nijlpaarden te voeren})^{x})^{[v_0,x]} \vdash [s]\mathsf{S}}\,[\mathrm{LP}']$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{(\!(nom)\!)\mathsf{NP}^{\{nom_0\}},\ (\textit{afspreekt}\ (\textit{de nijlpaarden te voeren})^{x})^{[v_0,x]} \vdash [s]\mathsf{S}}}{\textit{Cecilia}^{\{nom_0\}},\ (\textit{afspreekt}\ (\textit{de nijlpaarden te voeren})^{x})^{[v_0,x]} \vdash [s]\mathsf{S}}\,[\mathrm{Lex}]}{(\textit{Cecilia afspreekt}\ (\textit{de nijlpaarden te voeren})^{x})^{\mathcal{L}} \vdash [s]\mathsf{S}}\,[\mathrm{LP}']}{(\textit{Cecilia afspreekt}\ (\textit{de nijlpaarden te voeren})^{x})^{s} \vdash (\!(s)\!)\mathsf{S}}\,[(\!(s)\!)\mathrm{R}]$$

8.7 Adjuncts

Bouma and van Noord (1994a) illustrate the interaction of adjuncts with verb complementation by means of the following examples.

(5)a. *(dat) Frits Marie volgens mij lijkt te ontwijken*
that Frits Marie to me seems to avoid
'It seems to me that Frits avoids Marie'

b. *(dat) Frits Marie opzettelijk lijkt te ontwijken*
that Frits Marie deliberately seems to avoid
'It seems that Frits deliberately avoids Marie'

c. *(dat) Frits Marie de laatste tijd lijkt te ontwijken*
that Frits Marie lately seems to avoid
'Lately it seems as if Frits avoids Marie'
'It seems as if lately Frits avoids Marie'

Some adjuncts take scope over either the matrix verb or the governed verb, as (5)a and (5)b show, respectively. Bouma and Van Noord argue that in both cases only one of the two possible scopes is available due to semantic factors. This implies that what really needs to be accounted for are those cases, illustrated in (5)c, in which both possibilities exist.

This phenomenon is problematic for most categorial approaches to verb raising, which in some way or other employ restricted versions of non **L**-derivable

rules such as disharmonic composition and division. Bouma and Van Noord present a somewhat artificial solution which requires that verbs lexically subcategorize for the adjuncts modifying them. This means that they need to be subjected to a lexical rule which creates infinitely many lexical entries.

In the present set up there is no problem whatsoever, since in an **LP** setting both readings are generated. All we need to do is to specify that adjuncts are positioned after noun phrases, but before verbs. In other words, we add a modality $\langle adj \rangle$ and the LP constraints now become:

$$nom_0 \prec acc_0 \prec nom_1 \prec \ldots \prec adj \prec pre \prec v_0 \prec v_1 \prec \ldots \prec x$$

The type assignments for the sentence in question are as follows:

Frits, *Marie*:	$(\!(nom)\!)\mathsf{NP}, (\!(acc)\!)\mathsf{NP}$
de laatste tijd:	$(adj)(\mathsf{VP} \to \mathsf{VP})$
lijkt:	$(v)(\Diamond\mathsf{VP} \to \mathsf{VP})$
te ontwijken:	$(v)\mathsf{TVP}$

For the sentence (*dat*) *Frits Marie de laatste tijd lijkt te ontwijken* we can now obtain two derivations that each give one of the required scopings.

The conclusion of the first derivation, which accounts for the reading in which *de laatste tijd* has scope over the matrix verb, is the following sequent:

$$(\mathit{Frits}^{nom_0}\ \mathit{Marie}^{acc_1}\ \mathit{de\ laatste\ tijd\ lijkt}\ (\mathit{te\ ontwijken})^{\Diamond})^{s} \vdash (\!(s)\!)\mathsf{S}$$

After LP constraint checking, this sequent is reduced to:

$$(\!(nom)\!)\mathsf{NP}, (\!(acc)\!)\mathsf{NP}^{\Diamond}, (adj)(\mathsf{VP} \to \mathsf{VP}), (v)(\Diamond\mathsf{VP} \to \mathsf{VP}), (v)\mathsf{TVP}^{\Diamond} \vdash [s]\mathsf{S}$$

The derivability of this sequent follows from the derivation below:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{VP} \vdash \mathsf{VP}}{(\!(acc)\!)\mathsf{NP}, (\!(acc)\!)\mathsf{NP} \to \mathsf{VP} \vdash \mathsf{VP}}\,[\to\mathrm{L}^1]}{(\!(acc)\!)\mathsf{NP}, \mathsf{TVP} \vdash \mathsf{VP}}\,[\mathrm{Def}]}{(\!(acc)\!)\mathsf{NP}, (v)\mathsf{TVP} \vdash \mathsf{VP}}\,[(v)\mathrm{L}]}{((\!(acc)\!)\mathsf{NP}, (v)\mathsf{TVP})^{\Diamond} \vdash \Diamond\mathsf{VP}}\,[\Diamond\mathrm{R}]}{(\!(acc)\!)\mathsf{NP}^{\Diamond}, (v)\mathsf{TVP}^{\Diamond} \vdash \Diamond\mathsf{VP}}\,[\Diamond\mathrm{K}] \qquad \dfrac{\dfrac{\dfrac{\dfrac{[s]\mathsf{S} \vdash [s]\mathsf{S}}{(\!(nom)\!)\mathsf{NP}, (\!(nom)\!)\mathsf{NP} \to [s]\mathsf{S} \vdash [s]\mathsf{S}}\,[\to\mathrm{L}^1]}{(\!(nom)\!)\mathsf{NP}, \mathsf{VP} \vdash [s]\mathsf{S}}\,[\mathrm{Def}]}{(\!(nom)\!)\mathsf{NP}, \mathsf{VP} \to \mathsf{VP}, \mathsf{VP} \vdash [s]\mathsf{S}}\,[\to\mathrm{L}^1]}{(\!(nom)\!)\mathsf{NP}, (adj)(\mathsf{VP} \to \mathsf{VP}), \mathsf{VP} \vdash [s]\mathsf{S}}\,[(adj)\mathrm{L}]}{(\!(nom)\!)\mathsf{NP}, (\!(acc)\!)\mathsf{NP}^{\Diamond}, (adj)(\mathsf{VP} \to \mathsf{VP}), \Diamond\mathsf{VP} \to \mathsf{VP}, (v)\mathsf{TVP}^{\Diamond} \vdash [s]\mathsf{S}}\,[\to\mathrm{L}]}{(\!(nom)\!)\mathsf{NP}, (\!(acc)\!)\mathsf{NP}^{\Diamond}, (adj)(\mathsf{VP} \to \mathsf{VP}), (v)(\Diamond\mathsf{VP} \to \mathsf{VP}), (v)\mathsf{TVP}^{\Diamond} \vdash [s]\mathsf{S}}\,[(v)\mathrm{L}]$$

The reading where *de laatste tijd* takes scope over the nested verb is expressed by this sequent:

$$(\mathit{Frits}^{nom_0}\ \mathit{Marie}^{acc_1}\ (\mathit{de\ laatste\ tijd})^{\Diamond}\ \mathit{lijkt}\ (\mathit{te\ ontwijken})^{\Diamond})^{s} \vdash (\!(s)\!)\mathsf{S}$$

After LP constraint checking, this sequent is reduced to:

$$(\!(nom)\!)\mathsf{NP}, (\!(acc)\!)\mathsf{NP}^{\Diamond}, (adj)(\mathsf{VP} \to \mathsf{VP})^{\Diamond}, (v)(\Diamond\mathsf{VP} \to \mathsf{VP}), (v)\mathsf{TVP}^{\Diamond} \vdash [s]\mathsf{S}$$

The following derivation shows that this sequent is derivable:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{VP} \vdash \mathsf{VP}}{\mathsf{VP} \to \mathsf{VP}, \mathsf{VP} \vdash \mathsf{VP}}\,[{\to}\mathrm{L}^1]}{(\!(acc)\!)\mathsf{NP}, \mathsf{VP} \to \mathsf{VP}, (\!(acc)\!)\mathsf{NP} \to \mathsf{VP} \vdash \mathsf{VP}}\,[{\to}\mathrm{L}^1]}{(\!(acc)\!)\mathsf{NP}, \mathsf{VP} \to \mathsf{VP}, \mathsf{TVP} \vdash \mathsf{VP}}\,[\mathrm{Def}]}{(\!(acc)\!)\mathsf{NP}, (adj)(\mathsf{VP} \to \mathsf{VP}), (v)\mathsf{TVP} \vdash \mathsf{VP}}\,[(v)\mathrm{L},\ (adj)\mathrm{L}]}{((\!(acc)\!)\mathsf{NP}, (adj)(\mathsf{VP} \to \mathsf{VP}), (v)\mathsf{TVP})^{\diamond} \vdash \diamond\mathsf{VP}}\,[\diamond\mathrm{R}]}{(\!(acc)\!)\mathsf{NP}^{\diamond}, (adj)(\mathsf{VP} \to \mathsf{VP})^{\diamond}, (v)\mathsf{TVP}^{\diamond} \vdash \diamond\mathsf{VP}}\,[\diamond\mathrm{K}\ (2\mathrm{x})] \qquad \dfrac{\dfrac{[s]\mathsf{S} \vdash [s]\mathsf{S}}{(\!(nom)\!)\mathsf{NP}, (\!(nom)\!)\mathsf{NP} \to [s]\mathsf{S} \vdash [s]\mathsf{S}}\,[{\to}\mathrm{L}^1]}{(\!(nom)\!)\mathsf{NP}, \mathsf{VP} \vdash [s]\mathsf{S}}\,[\mathrm{Def}]}{(\!(nom)\!)\mathsf{NP}, (\!(acc)\!)\mathsf{NP}^{\diamond}, (adj)(\mathsf{VP} \to \mathsf{VP})^{\diamond}, \diamond\mathsf{VP} \to \mathsf{VP}, (v)\mathsf{TVP}^{\diamond} \vdash [s]\mathsf{S}}\,[{\to}\mathrm{L}]}{(\!(nom)\!)\mathsf{NP}, (\!(acc)\!)\mathsf{NP}^{\diamond}, (adj)(\mathsf{VP} \to \mathsf{VP})^{\diamond}, (v)(\diamond\mathsf{VP} \to \mathsf{VP}), (v)\mathsf{TVP}^{\diamond} \vdash [s]\mathsf{S}}\,[(v)\mathrm{L}]$$

8.8 Separable prefixes

Some Dutch verbs consist of a verbal head and a so-called *separable prefix*, a kind of particle which directly precedes either its governor or the verb cluster. This is illustrated in (6)–(8), where the verb *opbellen* is made up of the head *bellen* and the separable prefix *op*.

(6) *dat Jan Marie op belt*
that Jan Marie up calls
'that Jan calls up Marie'

(7)a. *dat Jan Marie wil op bellen*
that Jan Marie wants up call
'that Jan wants to call up Marie'

b. *dat Jan Marie op wil bellen*

(8)a. *dat Jan Marie zou willen op bellen*
that Jan Marie would want up call
'that Jan would like to call up Mary'

b. *dat Jan Marie zou willen op bellen*

To account for these data, a new modality $\langle sprf \rangle$ is introduced which labels separable prefixes. The two word orders displayed above can then be accounted for as follows:

1. The case in which the separable prefix comes at the beginning is achieved by simply ordering *sprf* behind all np_is, but before *pre*. The category of a separable prefix like *op* above then becomes $\langle sprf \rangle \mathsf{PRT}$, with its lexical head subcategorizing for it, e.g. *bellen* would be assigned the type $\mathsf{PRT} \to \mathsf{TVP}$;

2. The case where the separable prefix comes immediately in front of the verbal head can be explained in terms related Dowty's (to appear) notion of

attachment: the separable prefix attaches itself to the verb, which means that the combination becomes impenetrable for other material. In the present set up, this can be implemented in a very straightforward manner: just assign the verb an additional directed type, e.g. $\mathsf{PRT}\backslash^{(v)}\mathsf{TVP}$ for *bellen*.

With these type assignments and LP constraints, the grammaticality of (7)a and (7)b above follows from the (abridged) derivations below, which show that both *Marie op wil bellen* and *Marie wil op bellen* are well-ordered VPs:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{VP} \vdash \mathsf{VP}}{(\!(acc)\!)\mathsf{NP}, (\!(acc)\!)\mathsf{NP} \to \mathsf{VP} \vdash \mathsf{VP}}\,[\to\mathrm{L}^1]}{(\!(acc)\!)\mathsf{NP}, \mathsf{TVP} \vdash \mathsf{VP}}\,[\mathrm{Def}]}{(\!(acc)\!)\mathsf{NP}, \mathsf{PRT}, \mathsf{PRT} \to \mathsf{TVP} \vdash \mathsf{VP}}\,[\to\mathrm{L}^1]}{(\,(\!(acc)\!)\mathsf{NP}, \mathsf{PRT}, \mathsf{PRT} \to \mathsf{TVP}\,)^{\Diamond} \vdash \Diamond\mathsf{VP}}\,[\Diamond\mathrm{R}]}{(\,(\!(acc)\!)\mathsf{NP}, \mathsf{PRT}, \mathsf{PRT} \to \mathsf{TVP}\,)^{\Diamond}, \Diamond\mathsf{VP} \to \mathsf{VP} \vdash \mathsf{VP}}\,[\to\mathrm{L}^1]}{(\!(acc)\!)\mathsf{NP}^{\Diamond}, \mathsf{PRT}^{\Diamond}, \Diamond\mathsf{VP} \to \mathsf{VP}, \mathsf{PRT} \to \mathsf{TVP}^{\Diamond} \vdash \mathsf{VP}}\,[\Diamond\mathrm{K}\ (2\mathsf{x})]}{\vdots}}{\mathit{Marie}^{\Diamond\{acc_1\}}, \mathit{op}^{\Diamond\{sprf\}}, \mathit{wil}^{\{v_0\}}, \mathit{bellen}^{\Diamond\{v_1\}} \vdash \mathsf{VP}}\,[\mathrm{LP}'\ (3\mathsf{x})]}{(\mathit{Marie}^{\Diamond}\ \mathit{op}^{\Diamond}\ \mathit{wil}\ \mathit{bellen}^{\Diamond})^{[acc_0, x]} \vdash \mathsf{VP}}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{(\!(acc)\!)\mathsf{NP}^{\Diamond}, \Diamond\mathsf{VP} \to \mathsf{VP}, \mathsf{TVP} \vdash \mathsf{VP}}\,[(v)\mathrm{L}]}{(\!(acc)\!)\mathsf{NP}^{\Diamond}, \Diamond\mathsf{VP} \to \mathsf{VP}, (v)\mathsf{TVP} \vdash \mathsf{VP}}\,[\Diamond_i]}{(\!(acc)\!)\mathsf{NP}^{\Diamond}, \Diamond\mathsf{VP} \to \mathsf{VP}, [v]\mathsf{TVP}^{v_1} \vdash \mathsf{VP}}\,[\in]}{(\!(acc)\!)\mathsf{NP}^{\Diamond}, \Diamond\mathsf{VP} \to \mathsf{VP}, [v]\mathsf{TVP}^{v_1\{v_1\}} \vdash \mathsf{VP}}\,[\Diamond_i]}{(\!(acc)\!)\mathsf{NP}^{\Diamond}, \Diamond\mathsf{VP} \to \mathsf{VP}, (v)\mathsf{TVP}^{\Diamond\{v_1\}} \vdash \mathsf{VP}}\,[\backslash\mathrm{L}]}{(\!(acc)\!)\mathsf{NP}^{\Diamond}, \Diamond\mathsf{VP} \to \mathsf{VP}, (\mathsf{PRTPRT}\backslash^{(v)}\mathsf{TVP})^{\Diamond\{v_1\}} \vdash \mathsf{VP}}\,[sprf\mathrm{L}]}{(\!(acc)\!)\mathsf{NP}^{\Diamond}, \Diamond\mathsf{VP} \to \mathsf{VP}, (\mathit{sprf})\mathsf{PRTPRT}\backslash^{(v)}\mathsf{TVP})^{\Diamond\{v_1\}} \vdash \mathsf{VP}}\,[\mathrm{Lex}]}{(\!(acc)\!)\mathsf{NP}^{\Diamond}, \Diamond\mathsf{VP} \to \mathsf{VP}, (\mathit{op}\ \mathit{bellen})^{\Diamond\{v_1\}} \vdash \mathsf{VP}}}{\vdots}}{\mathit{Marie}^{\Diamond\{acc_1\}}, \mathit{wil}^{\{v_0\}}, (\mathit{op}\ \mathit{bellen})^{\Diamond\{v_1\}} \vdash \mathsf{VP}}\,[\mathrm{LP}'\ (2\mathsf{x})]}{(\mathit{Marie}^{\Diamond}\ \mathit{wil}\ (\mathit{op}\ \mathit{bellen}^{)}\Diamond)^{[acc_0, x]} \vdash \mathsf{VP}}$$

8.9 Conclusion

With his theory of word order domains, Mike Reape presents an alternative to a problematic basic assumption of most contemporary, which is that word order is derived from the terminal yield of phrase structure trees. This alternative has been worked out further in a HPSG setting by Kathol (1995), who calls it "Linearization-based Syntax".

It is important to realize that although linearization-based syntax was illustrated here through verb-raising, it is not just intended to account for exceptional constructions that are hard to deal with in a more traditional setting based on phrase-structure trees. For instance, Kathol extends a modified version of Reape's proposal to account for a whole range of word order phenomena in German. The accounts of Hoeksema and Dowty mentioned in Section 8.3 deal with basic constructions in relatively fixed word order languages such as Dutch and English.

Implicit in Reape's approach is a categorial approach to syntactic combination. In this chapter, I have shown how this implicit component can be made explicit by reformulating Reape's proposals in a categorial setting. This strongly suggests the viability of a linearization-based categorial approach to the syntax of natural languages. However, as a pilot study, the proposal put forth in this chapter still suffers from a number of defects, whose resolution will depend upon further research. Two of the more important problems are these:

- If sizeable grammar fragments are to be built based on the ideas presented here, a fairly elaborate feature system is unavoidable. It is doubtful whether the approach sketched here would be easily extendible to such sophisticated feature systems, and most likely a thorough investigation of the relation between feature systems and feature modalities would be needed.

- In order to allow rigorous checking of multimodal grammars such as the ones presented here, an implementation is needed which allows reasonably efficient parsing. As they are currently formulated, there is quite a bit of indeterminism in the logics, for instance in the antecedents of conclusion sequents, the structure of which needs to be guessed by the parser.

Samenvatting

Taalkunde is de wetenschap die zich bezighoudt met de structuur van menselijke taal. De belangstelling gaat daarbij niet zozeer uit naar specifieke talen, maar vooral naar de structuur en eigenschappen die alle talen met elkaar gemeen hebben, hoe verschillend ze op het eerste gezicht ook mogen lijken.

Er zijn verschillende taalkundige theorieën, en een daarvan vormt het onderwerp van dit proefschrift: *categoriale grammatica*. Het belangrijkste kenmerk dat de categoriale grammatica onderscheidt van andere theorieën is de grote rol die zij toekent aan de *logica*. Een categoriale grammatica is in feite een speciaal soort logica, en aantonen dat een bepaalde zin grammaticaal is, komt neer op het bewijzen van een stelling in deze logica.

Het proefschrift bestaat uit vier delen, die elk twee hoofdstukken omvatten.

Het eerste deel heeft een inleidend karakter. Hoofdstuk 1 begint met een beknopte historische inleiding, die wordt gevolgd door een beschrijving van de *Lambek calculus*, een systeem dat in de categoriale grammatica een centrale plaats inneemt. In hoofdstuk 2 komen een aantal uitbreidingen van de Lambek calculus aan de orde die de afgelopen tien jaar zijn voorgesteld in de literatuur. Dit overzicht eindigt met *multimodale* categoriale grammatica's, waarin verschillende soorten structuur gelijktijdig beschreven kunnen worden. Het zijn systemen van dit type die het onderwerp van studie vormen in de overige drie delen.

In een nette logica wordt de betekenis van wat er zich in het bewijssysteem afspeelt nauwkeurig omschreven. In deel twee wordt dit gedaan voor een specifieke soort multimodale grammatica's, bestaande uit traditionele systemen waaraan *modaliteiten* zijn toegevoegd, operatoren die kenmerkende eigenschappen van woorden of zinsdelen kunnen beschrijven. In hoofdstuk 3 wordt het eenvoudigste dergelijke systeem bekeken, en *volledigheid* aangetoond, wat betekent dat alle stellingen die volgens de aan het systeem gehechte betekenis waar zijn ook inderdaad kunnen worden bewezen. De hierbij opgedane inzichten komen vervolgens goed van pas in hoofdstuk 4, waar dit bewijs wordt uitgebreid tot een grote groep van vergelijkbare systemen.

Het derde deel gaat over logica's die weliswaar verwant zijn aan die uit deel twee, maar die ten gevolge van een andere opvatting van de betekenis ook andere regels kennen. In hoofdstuk 5 worden deze op *residuatie* gebaseerde systemen geïntroduceerd, en wordt voor een specifiek paar van dergelijke logica's getoond hoe het gedrag van de ene in de andere kan worden beschreven zonder dat hiervoor essentiële toevoeg-

ingen nodig zijn. Het onderwerp van hoofdstuk 6 is de *conjoinability*-relatie, waarvan voor een aantal belangrijke systemen wordt bewezen dat zij *beslisbaar* is, wat wil zeggen dat van ieder willekeurig tweetal uitdrukkingen kan worden bepaald of ze in deze relatie tot elkaar staan.

Het vierde en laatste deel behelst een meer concrete taalkundige toepassing van de in deel drie beschreven systemen. In hoofdstuk 7 wordt de *verb-raising* constructie uit het Nederlands besproken, gevolgd door een overzicht van een aantal verklaringen die er door verschillende theorieën voor gegeven zijn. Eén van deze verklaringen, Reape's theorie van *woordvolgordedomeinen*, vormt de inspiratie voor het laatste hoofdstuk. Daarin worden de belangrijkste verb-raising voorbeelden beschreven door middel van een multimodale categoriale grammatica waarin de onderlinge volgorde van de woorden in een uitdrukking gedeeltelijk is losgekoppeld van de structuur die deze uitdrukking heeft.

Curriculum Vitae

My parents are Cor Versmissen and Han van der Linden, and I was born in the southern Dutch village of Waalre on xx yy zzzz. Little did I realize when my mother first took me to kindergarten a few years later that I was embarking on a quarter century of well-nigh continuous study. From kindergarten, this educational path led me first to primary school, De Wilderen in Waalre. This was followed by six years of secondary school at the Hertog Jan College in Valkenswaard, where I obtained my gymnasium-β degree in 1984.

I started my academic years as a student of mathematics at Utrecht University, graduating in June 1989 on a *doctoraal* thesis about generalized polygons which I wrote under the supervision of Arjeh Cohen. Not really knowing where I wanted to go from there, and having recently gotten interested in linguistics, I decided to do a second degree: computer science. Thanks to my background I would be able to complete this degree fairly rapidly, after all, and it allowed me to do a linguistics minor. Being by no means narrow-minded, the computer science department allowed me to write a *doctoraal* thesis on categorial grammar. This thesis, which I completed in October 1991, was supervised by Doaitse Swierstra, Nico Verwer and Michael Moortgat.

From January 1992 to December 1995, I was employed by Utrecht University's OTS (the Research Institute for Language and Speech) as a Ph.D. student, or *AiO*. My task was to do research on the broadly formulated subject of "categorial grammar and substructural logics", again under the supervision of Michael Moortgat. This Ph.D. thesis presents the results of my investigations.

On 1 April 1996, I returned to the department of computer science, this time as a part-time employee: coordinator of SIKS, the Dutch research school for knowledge and information systems.

In my spare time I enjoy playing bridge and chess. I am particularly interested in chess problems, and I have been the president of the *Nederlandse Bond van Schaakprobleemvrienden* (Dutch Chess Problem Society) since March 1995.