The Grammar and Processing of Order and Dependency: a Categorial Approach

Mark Hepple

Ph.D.

University of Edinburgh
1990

Declaration

I declare that this thesis has been composed by me and that the research reported herein has been conducted by me unless otherwise indicated.

Mark Hepple

Edinburgh, 19th September 1990.

Acknowledgements

I would like to thank my supervisors, Elisabet Engdahl and Mark Steedman. Elisabet has been a continuing source of support. Her comments, criticisms and suggestions have been of great importance to any virtues the final thesis may have. It was through Mark's work that I first developed a passion for categorial grammar. Mark's enthusiasm and encouragement have helped me to maintain a constant momentum throughout the entire course of the thesis.

Special thanks are due to Guy Barry and Glyn Morrill who have influenced this work at almost every stage. Glyn and Guy have played central roles in the development of categorial grammar at the Centre for Cognitive Science in the past two years. Their ears provided the main testing ground in developing the accounts presented, and their comments on draft material have helped considerably in improving final presentation. Without our frequent discussions, categorial grammar would not have been half so much fun. Thanks also to Neil Leslie, whose contribution has been vital in the development of the framework within which the thesis is set.

The work has also benefited from discussions with many other people. I would like to mention: Inge Bethke, Martin Emms, Esther König, Michael McPartlin, Martin Pickering, Mike Reape, Antonio Sanfilippo, Anna Szabolcsi, Max Volino.

Special thanks to my mother and father, and to my wife's parents, Jim and Judith. Without their constant support and encouragement this thesis may never have come to pass.

My deepest gratitude goes to Isobel, my beloved wife, for her love, support, encouragement, humour and tolerance, and for just being there.

Abstract

This thesis presents accounts of a range of linguistic phenomena in an extended categorial framework, and develops proposals for processing grammars set within this framework. Linguistic phenomena whose treatment we address include word order, grammatical relations and obliqueness, extraction and island constraints, and binding. The work is set within a flexible categorial framework which is a version of the Lambek calculus (Lambek, 1958) extended by the inclusion of additional type-forming operators whose logical behaviour allows for the characterization of some aspect of linguistic phenomena.

We begin with the treatment of extraction phenomena and island constraints. An account is developed in which there are many interrelated notions of boundary, and where the sensitivity of any syntactic process to a particular class of boundaries can be addressed within the grammar.

We next present a new categorial treatment of word order which factors apart the specification of the order of a head's complements from the position of the head relative to them. This move has the advantage of allowing the incorporation of a treatment of grammatical relations and obliqueness, as well as providing for the treatment of Verb Second phenomena in Germanic languages.

A categorial treatment of binding is then presented which integrates the preceding proposals of the thesis, handling command constraints on binding in terms of relative obliqueness and locality constraints using the account of linguistic boundaries. Attention is given to the treatment of long distance reflexivization in Icelandic, a phenomenon of interest because of its unusual locality behaviour.

Finally, a method is developed for parsing Lambek calculus grammars which avoids the efficiency problems presented by the occurrence of multiple equivalent proofs. The method involves developing a notion of normal form proof and adapting the parsing method to ensure that only normal form proofs are constructed.

Contents

Chapter 1

Introduction

This thesis presents accounts of a range of linguistic phenomena in an extended Lambek calculus framework, and in addition develops proposals for processing grammars set within this framework. The essential goals of this work are threefold. Firstly, to develop the extended Lambek approach as a framework for linguistic description. Secondly, to create new accounts for linguistic phenomena within the extended Lambek framework. Thirdly, to develop methods for efficient processing of Lambek calculus grammars as a step towards the possibility of these grammars being used in real natural language processing applications. In this chapter, we begin by discussing these goals in slightly more detail, as well as some other considerations that bear on our choice of the extended Lambek calculus as a framework for linguistic research. After this discussion, we give an overview of the remainder of thesis.

1.1 General

Categorial grammar (CG) is an approach to language description which classifies linguistic expressions in terms of recursively defined types. Let us consider some important characteristics of CG approaches.

Firstly, CG approaches can be seen to fit in with a central trend in current linguistics, namely *lexicalism*, whereby a greater burden is placed on the lexicon for the encoding of syntactic information. CG approaches achieve this goal by an enrichment of category structure which enables lexical type assignments to encode most or all of the combinatoric possibilities for each word.

CGs are typically *monostratal*, i.e. invoking only a single level of syntactic description.

CGs typically exhibit a very close correspondence of syntax and semantics, so that syntactic structure can in general be seen as essentially just a refinement of 'semantic structure' (i.e. semantic function-argument structure). Thus, CG stands in opposition to the 'radical autonomy of syntax' position advocated by many in the transformational tradition (e.g. Chomsky, 1975) whereby syntactic structures are taken to arise under constraints that are fundamentally specific to the syntactic domain, and where the interpretation of such structures is an independent process.

A further characteristic of categorially based research is that, like some other modern monostratal frameworks, it has typically exhibited a strong commitment to formal rigour, with grammars often receiving a formal statement in terms of some logic. One respect in which such rigour is important is determining the empirical consequences of an account, and more specifically whether a given grammar does or does not admit a particular sentence. This contrasts most obviously with work in the Government and Binding (GB) framework (Chomksy, 1981), where, as has often been pointed out, the empirical consequences of accounts cannot in the general case be precisely determined, since 'grammars' typically receive no formal statement.

The particular CG that we adopt as the framework for this thesis is a version of extended Lambek calculus. This approach takes as its starting point the (unextended) calculus of Lambek (1958, 1962). The Lambek calculus is a highly lexicalist approach, with the combination of expressions being governed by a highly general type change calculus. The inference rules of the calculus govern the behaviour of individual type-forming operators, and should be constant across different languages. Hence, the burden for specifying language particular characteristics rests solely on the lexicon and lexical type assignment. The Lambek calculus is demonstrably decidable for reasons that are mentioned in the next chapter. This means that no problem arises in determining whether or not some particular sentence is admitted under a given set of lexical type assignments.

The extended Lambek calculus arises through augmenting the standard Lambek calculus by the inclusion of additional type forming operators whose logical behaviour allows for the characterization of some aspect of linguistic phenomena. Many of these augmentations are inspired by ideas from linear logic (Girard, 1987). Current research in the area can be seen as attempting to determine an adequate set of descriptive primitives. This task, quite obviously, can only be achieved through work that attempts to create accounts of a wide range of linguistic phenomena. During the course of the thesis, we introduce a number of new type-forming operators which we believe are required to handle some aspect of natural language phenomena.

The extended Lambek framework provides a new perspective for addressing linguistic problems, differing significantly from previous approaches. One characteristic we have already noted is that it is recognized that new primitives may *need* to be created, to deal with certain aspects of linguistic phenomena. This is not to say, however, that primitives having completely arbitrary behaviour should be admitted. The creation of new primitives is subject to the constraint that they should behave as logical operators, whose behaviour is determined by logical inference rules which may be included in a free calculus of type-change.

It is hoped that the extended Lambek calculus, in providing a novel perspective on linguistic phenomena, will lead to new insights in their explanation. The value of pursuing a new grammatical framework is often that it may encourage new approaches to handling familiar phenomena. In practice, solutions within one framework can frequently be 'reconstructed' within another framework, so that it is typically not the case that the essential intuition underlying a new explanation is fundamentally specific to that framework. The point, however, is that it may be the individual character of a new framework that tends to *suggest* the possibility of approaching a familiar problem in a particular novel way.

It is interesting to compare the extended Lambek approach with some alternative highly lexicalist approaches which place considerable importance on the use of *unification* formalisms. We feel that one problem of approaches which crucially depend on unification is that the power of the formalism is such as to allow what are essentially uninteresting solutions to linguistic problems to be encoded ('explanations' that basically just 'hack the data'). Of course, unification is *just* a formalism, and should not blamed for the details of solutions that are set within it. However, the impression is that unification formalisms, by their character, tend to foster poor solutions which are based on passing around arbitrarily large amounts of information as feature values. We take it to be an advantage of the extended Lambek approach that it does not allow for accounts of this kind, i.e. that the approach provides a certain *discipline* in the development of solutions.

It is important, however, to recognize the *limitations* of our framework. In particular, the approach as we formulate it in this thesis only allows for the characterization of *syntactic* constraints, it doesn't really allow for the characterization of constraints that properly belong in the *semantic* domain. One possibility here is to cheat a little and (occasionally) use a syntactic feature to make explicit some distinction that appears essentially semantic in character, so that we in effect 'grammaticalize' the relevant semantic property.[1] Note that we take the use of features to encode essentially semantic properties to be no more than a

[1] Such 'grammaticalization' of semantic characteristics as a 'trick' for overcoming limitations of the framework should not be confused with claims of genuine grammaticalization in linguistic accounts, i.e. claims to the effect that a language exhibits certain constraints that demonstrate a 'fossilization' of some originally semantic properties as stipulative syntactic requirements.

(perhaps necessary) trick in general. This criticism applies equally well to unification based approaches which use features to encode semantic properties. Embedding a feature within a structure labeled SEMANTICS does not avoid the criticism.

An alternative possibility is suggested by the type theoretic emphasis of the extended Lambek approach, which is that a further enrichment of the semantic type system be made to encode more subtle semantic characteristics than has been done so far, as a basis for encoding semantic constraints. Such a move might require us to modify the idea of syntactic types being simply a refinement of semantic types, as not all properties relevant at the semantic level would need to be 'visible' at the syntactic level. Syntactic and semantic type structure might perhaps arise as a refinement of a further level of *minimal* semantic type structure. We shall not attempt to explore this possibility in this thesis.

1.2 Overview

In Chapter 2, we begin by introducing categorial grammar and the Lambek calculus. Two alternative formulations of the Lambek calculus are presented. This is because the two formulations are suitable for different purposes. One formulation, in terms of *natural deduction*, is useful for presenting proofs in linguistic discussion. A second *sequent* formulation is valuable for parsing. We then consider some augmentations of the basic Lambek calculus and illustrate their use. We present only those extensions required for the rest of the thesis. Further extensions are proposed in context in the later chapters.

Chapter 3 addresses the treatment of extraction in categorial grammar. The discussion divides into two major subparts. Firstly, we discuss the problems that the treatment of *non-peripheral* extraction (i.e. extraction from non-peripheral positions) presents for CG approaches. In particular, we look at the difficulties that arise for treatments of non-peripheral extraction that are based on the inclusion of *disharmonic* rules of composition. These problems serve to indirectly motivate our treatment of extraction, which is based on the use of a *structural modality* operator. The second major topic addressed is the treatment of constraints on extraction, so-called *island constraints*. We consider problems that arise for some current accounts of island behaviour, and argue that these problems indicate the need for a treatment of subjacency that goes beyond a simple unitary notion of 'linguistic boundary'. Such an approach is developed in terms of a *polymodal* logic of necessity, which provides for an account in which there are many interrelated notions of boundary, and where the sensitivity of any syntactic process to a particular class of boundaries can be directly addressed within the grammar.

Chapter 4 is primarily concerned with the treatment of word order. A new CG treatment of word order is proposed which factors apart the specification of the order of a head's complements from the specification of the position of the head relative to them. This move has a number of advantages. Firstly, it allows for the incorporation of a theory of grammatical relations and obliqueness, providing a basis for handling a range of phenomena that have been argued to depend on these notions. Secondly, this move provides fairly straightforwardly for the treatment of Verb Second behaviour in Germanic languages. More generally, we argue that this approach provides a perspective within which cross-linguistic variation amongst the Germanic languages can be addressed, since it allows us to handle the obvious word order differences between the Germanic languages, but in a way that still recognizes underlying similarities between them.

Chapter 5 addresses the treatment of binding phenomena. We begin by considering some previous approaches and problems that arise for them. Then we present two alternative accounts of binding within the extended Lambek framework. The first of these is based on the use of a designated structural modality operator. We consider the treatment of both syntactic and discourse binding of anaphors. The account handles 'command' constraints on binding in terms of obliqueness (depending crucially for this on the treatment of word order developed in Chapter 4) and uses the polymodal treatment of linguistic boundaries developed in Chapter 3 to handle locality constraints on binding. Thus, the account provides a point of synthesis for the proposals of the preceding chapters. A second account is presented which originates with the proposals of Moortgat (1990b), and which is developed to incorporate the obliqueness based treatment of command and the polymodal treatment of linguistic boundaries. We then go on to consider long distance binding in Icelandic, where a reflexive may have an antecedent that occurs outside the tensed clause in which the reflexive occurs. Such binding, however, is not free of locality constraints, but appears instead to involve a constrained extension of binding domain. Our account of this phenomenon crucially depends on the polymodal treatment of linguistic boundaries.

Chapter 6 is concerned with the efficient parsing of Lambek calculus grammars. The sequent formulation is best suited for Lambek parsing, which may be treated as as instance of automated theorem proving. A problem arises for Lambek theorem proving, which is that the calculus typically allows many distinct proofs which assign the same meaning. Since search for proofs must be exhaustive to be sure of finding all different readings that may be assigned to a sequent, a naive theorem prover must do a lot of unnecessary work constructing proofs that assign the same meaning. This wasted effort radically reduces the efficiency of

Lambek theorem proving. We develop a solution to this problem which involves specifying a notion of normal form proof and adapting the theorem proving method so that only normal form proofs are returned. We demonstrate that the method is *safe* in the sense that the theorem prover will return every distinct reading that can be assigned to a given sequent, and *optimal* in the sense that only one proof will be returned which assigns each reading.

Chapter 2

The Categorial framework

The principal goal of this chapter is to introduce the categorial framework within which the rest of the thesis is set. This is a version of extended Lambek calculus, and falls within the family of *flexible* categorial grammars which are characterized by the property of allowing considerable flexibility in the assignment of types to strings, and, thereby, flexibility also in the assignment of syntactic structure. Note that our intention here is merely to prepare the reader for the ensuing presentation. The chapter is not intended to be an overview of the broad range of categorial grammars that have been proposed.

2.1 Categorial grammar

A categorial grammar (CG) consists of two components. Firstly, a categorial lexicon, which assigns to each word at least one syntactic type (plus associated interpretation). Secondly, a logic which determines the set of admitted type combinations and type transitions. This gives rise to a notion of *derivability*, whereby some type combination:

$$\mathrm{x}_1, \ldots, \mathrm{x}_n \Rightarrow \mathrm{y}$$

(where y, x_1, ..., x_n are types) is *derivable* if and only if there is a proof of that type combination under the given logic (i.e. the type combination is a *theorem*). This notion of derivability corresponds to that of *entailment* in standard logics such as the propositional calculus. The set of types is defined recursively in terms of a set of basic types and a set of operators. For standard *bidirectional* CG, the connectives are / and \, and the set of types is defined as follows:

(2.1) *The Set of Types (T)*

Given a set of basic types B, the set of types T is the smallest set such that:

a. *if* x $\in$ B, *then* x $\in$ T

b. *if* x,y $\in$ T, *then* x/y, x\y $\in$ T

Intuitively, the types assigned to words specify their subcategorization requirements, as well as information of constituent order. Thus, types X/Y and X\Y are functions which require an argument of type Y, which must occur to the right and left of the function, respectively.[1]

2.1.1 Applicative categorial grammars

The simplest categorial grammars, sometimes called AB grammars, provide only rules of application to allow for type combination. Adjukiewicz (1935), developing proposals of Lesniewski, proposed a categorial system based on a single directionless implicational connective. Bar-Hillel (1953) proposed a directed version of categorial grammar, using the type-forming operators / and \, and using types as defined in (2.1). This calculus employs the following *directed* versions of functional application:

(2.2) (>) X/Y, Y $\Rightarrow$ X
(<) Y, X\Y $\Rightarrow$ X

These rules allow the following derivation of the sentence *John loves Mary*, assuming the lexical type assignments shown:

(2.3)
```
John     loves      Mary
----   ---------    ----
 np    s\np/np       np
       ---------------->
             s\np
  ------------------<
          s
```

It is standard in CG approaches to assume that the operations for combining the interpretations of constituents are homomorphic to the syntactic operations. For classical AB grammars, the syntactic application rules are interpreted semantically as application also, i.e. so that the meaning of the result expression of some combination is given by applying the meaning of the functor expression to the meaning of the argument expression. Thus, the meaning assigned by the derivation in (2.3) is:

$(\text{loves}' \ \text{mary}')\ \text{john}'$

[1] In the notation we use, the argument symbol occurs to the right of the value symbol in the symbol for the function irrespective of the function's directionality. This contrasts with the notation used by Lambek (1958), where the order of value and argument symbols in the function notation depends on the directionality of the argument. Note that we assume a convention of *left association* for binary connectives, so that, for example, (s\np)/np may be written s\np/np.

2.1.2 Flexible categorial grammars

Various proposals have been made for augmenting the type combination component of applicative grammars by the addition of further type combination and type transition schemes. Suggestions for additional type combination schemes include rules of *type raising*, *division* (sometimes called the Geach rule), and *composition*, exemplified in (2.4a,b,c), respectively:

(2.4) a. $X \Rightarrow Y/(Y\backslash X)$

b. $X/Y \Rightarrow X/Z/(Y/Z)$

c. $X/Y,\ Y/Z \Rightarrow X/Z$

The addition of such rules has the effect of allowing greater flexibility in the assignment of types to strings, and, thereby, flexibility also in the assignment of syntactic structure.

For example, a system which includes the rules (2.4a,b), in addition to the application rules, allows the following derivation for *John loves Mary* as an alternative to that in (2.3):

(2.5)
```
      John        loves      Mary
      ----      --------     ----
       np       s\np/np       np
     -------
     s/(s\np)
  --------------
  s/np/(s\np/np)
  --------------------------->
             s/np
             ----------------------->
                        s
```

Observe in (2.5) that the string *John* receives a type s/(s\np), and the string *John loves* receives a type s/np. With the purely applicative grammar, neither of these types could be derived for these strings, which illustrates the point that the inclusion of additional type combination schemes allows greater flexibility in the assignment of types to strings. That the string *John loves* has constituent status in (2.5) illustrates the greater flexibility allowed in the assignment of syntactic structure.

Various arguments have been advanced in favour of flexible constituency. These include arguments that relate to semantic considerations (e.g. Geach, 1972), the treatment of extraction (Ades & Steedman, 1982), the treatment of non-constituent coordination (Steedman, 1985; Dowty, 1988), and providing for the possibility of incremental interpretation in human sentence processing (Steedman, 1989). We shall not rehearse these arguments here. The reader is referred to the cited references, as well as other references that appear in the bibliography.

Speaking more generally, we can expect categorial logics which are suitable for the description of natural language, including facts of word order, to be of a certain character. Firstly, they will be *occurrence* logics, i.e. the number of occurrences of the types that participate in any type combination is of note. For example, a function X/Y requires for its complement precisely one occurrence of a type Y — zero or more than one occurrences will not do. Thus, type combinations such as the following are in general excluded:

$$X/Y, Y, Y \Rightarrow X$$

$$X/Y \Rightarrow X$$

Intuitively, the significance of this requirement is that, one the one hand, a head must be able to *require* the presence of some complement (e.g. so that *runs* may not be analysed as a clause), and on the other hand, that extraneous material may not appear in a clause (e.g. so that *John ran apple* is not a clause). Secondly, appropriate logics will be *non-commutative*. Thus, *commutative* logics contain rules (called *structural rules*) which in effect allow random permutation of the types from which some entailment follows, with the effect that the initial linear order of these types has no effect on whether the entailment holds or not. To claim that categorial logics for natural language should be non-commutative is essentially just to say that word order matters in natural language.

2.2 The Lambek calculus

We look next at a particular flexible CG framework, the Lambek calculus, proposed by Lambek (1958, 1961). This uses a third type forming operator * (in addition to / and \) referred to as the *product* operator. However, we make no use of the product operator in this thesis, and so we shall address only the *product-free* subsystem of the calculus here.[2]

The essential distinguishing characteristic of the Lambek calculus as a categorial framework is that, in addition to inference rules which govern the elimination of the type forming connectives, the Lambek calculus also provides inference rules which govern the *introduction* of these connectives. In the case of the implicational fragment, such inferences correspond to *conditionalization* (familiar from the propositional calculus) or, alternatively, *functional abstraction*. Consider again flexible CGs which are augmented by the addition of a finite number of type combination schemes (i.e. such as those discussed in the previous section). We can see that such additional type combination schemes make available limited possibilities for steps of functional abstraction but only in combination with other steps of application. In the Lambek calculus, the inference rules make the possibility of functional abstraction directly available as primitive steps.[3]

[2] This is alternatively known as the *implicational* subsystem of the Lambek calculus, given its dependence on only the connectives / and \, which are analogous to (directed variants of) *material implication* (→) in other logics.

[3] The correctness of this characterization is not obvious for the particular formulation of the Lambek calculus proposed in Zielonka (1981). However, Zielonka's formulation is demonstrably equivalent to alternative

Alternative formulations of the Lambek calculus are possible. We will focus on two in particular: the natural deduction formulation developed in Morrill, Leslie, Hepple and Barry (1990), and the Gentzen-style sequent formulation of Lambek (1958). The reason for using both formulations is that each is best suited for a different role. Natural deduction is convenient for the purpose of presenting proofs on the page, these proofs being (comparatively) easy to read. In contrast, the sequent formulation is best suited to the task of proof-search (i.e. parsing), enabling simple algorithms for this. In Chapters 3, 4 and 5, our presentation depends primarily on the natural deduction formulation, since these chapters emphasize linguistic discussion. Even so, wherever natural deduction rules are provided, corresponding sequent rules will also be given, since we attach considerable importance to the parsability of the grammars that are developed. Chapter 6 addresses the efficient parsing of Lambek calculus grammars, and so is based solely around the sequent approach.

2.2.1 Natural deduction and the Lambek calculus

We next consider the natural deduction formulation of the Lambek calculus presented in Morrill, Leslie, Hepple and Barry (1990). The sequent formulation will be considered in the next subsection. The present formulation develops from the natural deduction systems of Prawitz (1965). Natural deduction was invented by Gentzen (1936) to reflect the natural process of mathematical reasoning in which one uses a number of *inference rules* to justify a single *conclusion* on the basis of a number of propositions, called *assumptions*. In proving some entailment:

$$\mathrm{x}_1, \ldots, \mathrm{x}_n \Rightarrow \mathrm{y}$$

additional assumptions may be invoked, over and above the assumptions $\mathrm{x}_1, \ldots, \mathrm{x}_n$ on which the final entailment rests, provided that some inference in the proof licenses the withdrawal of each of the additional assumptions. The inference is said to *discharge* the assumption. The conclusion is said to *depend* on the undischarged assumptions, which are called the *hypotheses* of the proof.

We use the notation: $\begin{matrix}\vdots\\ \mathrm{A}\end{matrix}$

to designate a proof of A. Proofs proceed from a number of initial assumptions, some of which may be discharged during the course of the proof (with square brackets being used to indicate a discharged assumption). Each assumption is basically just some type. Note that, in constrast to standard natural deduction systems, the order of initial hypotheses is important here. We require the following natural deduction rules:

formulations of the Lambek calculus for which the characterization clearly is appropriate.

(2.6) Hypothesis rule: A

Rules for /:

$$\frac{\begin{array}{cc}\vdots & \vdots\\ A/B & B\end{array}}{A}\,/E \qquad \frac{\begin{array}{c}[B]^{i}\\ \vdots\\ A\end{array}}{A/B}\,/I^{i}$$

where B is the rightmost undischarged assumption in the proof of A

Rules for \\:

$$\frac{\begin{array}{cc}\vdots & \vdots\\ B & A\backslash B\end{array}}{A}\,\backslash E \qquad \frac{\begin{array}{c}[B]^{i}\\ \vdots\\ A\end{array}}{A\backslash B}\,\backslash I^{i}$$

where B is the leftmost undischarged assumption in the proof of A

These rules are sufficient for the product-free Lambek calculus. The hypothesis rule states that a type A is a proof of that type. The elimination rule [/E] states that given a proof of A/B and a proof of B, these proofs may be combined as shown to give a proof of A. The introduction rule [/I] states that given a proof of some type A, we may discharge some hypothesis B within the body of the proof and form a proof of A/B. This rule has a 'side condition' which requires that the assumption discharged by the rule must be the rightmost (undischarged) assumption in the proof to which the rule is applied. This condition is required to ensure that the creation of types under conditionalization respects the ordering requirements of the types combined. To indicate which assumption has been discharged in the application of an introduction rule, the assumption and inference are 'coindexed'. To demonstrate that some result type x_0 can be derived from sequence of types $x_1,..,x_n$ a proof of x_0 is given having hypotheses $x_1,..,x_n$ (in that order) with zero or more discharged assumptions interspersed amongst them.

The following proof of the combination a/b, b/c $\Rightarrow$ a/c ('simple forward composition') illustrates the natural deduction system:

(2.7)

$$\frac{\dfrac{a/b \qquad \dfrac{b/c \quad [c]^{i}}{b}\,/E}{a}\,/E}{a/c}\,/I^{i}$$

We may augment this system to provide the semantics of type combinations. For this purpose, each type in a proof is associated with a lambda expression, its *proof term*, corresponding to its meaning. The proof term for the conclusion of the proof expresses the conclusion's meaning as a combination of the meanings of the undischarged assumptions of the proof, and the inference rules specify how this lambda expression is constructed. Each initial assumption is assigned a (unique) variable for its semantics. The elimination and introduction rules correspond to semantic operations of functional application and abstraction, respectively.

(2.8) Hypothesis rule: $A{:}v$

Rules for /:

$$\frac{\begin{matrix}\vdots & \vdots\\ A/B{:}f & B{:}v\end{matrix}}{A{:}(fv)}\,/E \qquad \frac{\begin{matrix}[B{:}v]^{i}\\ \vdots\\ A{:}f\end{matrix}}{A/B{:}(\lambda v.f)}\,/I^{i}$$

where B is the rightmost undischarged assumption in the proof of A

Rules for \:

$$\frac{\begin{matrix}\vdots & \vdots\\ B{:}v & A\backslash B{:}f\end{matrix}}{A{:}(fv)}\,\backslash E \qquad \frac{\begin{matrix}[B{:}v]^{i}\\ \vdots\\ A{:}f\end{matrix}}{A\backslash B{:}(\lambda v.f)}\,\backslash I^{i}$$

where B is the leftmost undischarged assumption in the proof of A

Under this approach, each proof yields a lambda expression (i.e. the proof term of the conclusion type) which contains one free variable occurrence for each undischarged assumption and no other free variable occurrences. The following proof repeats that of (2.7), but with proof terms added:

(2.9)

$$\dfrac{\dfrac{a/b{:}x \qquad \dfrac{b/c{:}y \quad [c{:}z]^{i}}{b{:}(y\,z)}\,/E}{a{:}(x\,(y\,z))}\,/E}{a/c{:}\lambda z.(x\,(y\,z))}\,/I^{i}$$

In general, proof terms will be omitted to simplify presentation. When we are interested in some proof with regard to its use for a particular combination of lexical types assigned to given words, we will commonly present the proof with the words stated above the relevant hypotheses, and present the semantics for the combination with the lexical meanings for each word in place of the free variable occurrence corresponding to the relevant hypothesis.

It is probably worth emphasizing that the presence of discharged assumptions in natural deduction proofs is essentially an artefact of the proof method. This should *not* be construed as some significant aspect of linguistic structure, akin to traces or empty categories in the representations of, say, GB theory. Note also that the coindexing of introduction inferences with the assumption that these discharge is purely to help the reader comprehend the structure of the proof. No inference rules will have side conditions that refer to the pattern of coindexing within a proof. Thus, again, such coindexing is merely an aspect of the proof notation and should not be misconstrued as an aspect of linguistic structure.

We might include the following inference rule, called *cut*, in addition to those stated above:

(2.10) $$\begin{array}{c} \vdots \\ B \end{array} \quad \begin{array}{c} B \\ \vdots \\ A \end{array} \quad \Longrightarrow \quad \begin{array}{c} \vdots \\ B \\ \vdots \\ A \end{array}$$

The rules given earlier state only the form of the proof which results from the application of the rule, the form of the proof(s) to which the rule is applied remains implicit. In this case, however, we explicitly state the form of both the input proofs and the resulting proof to avoid any confusion. The cut rule simply states that given a proof of some type B and another proof which rests on an (undischarged) assumption B, the two proofs may be combined, with the first proof being substituted in place of the assumption B in the second proof. This rule expresses a fundamental property of derivability: that it is transitive.

The inclusion of this rule would allow, for example, the elimination rules to be given a simpler statement than above, i.e. specifying only the combination of assumptions rather than proofs — the cut rule allowing proofs for the assumptions to be inserted at a later stage. However, it is more convenient, and also more in keeping with the general natural deduction perspective, to have the elimination rules combine proofs.

2.2.2 The sequent formulation of the Lambek calculus

We shall next consider a second formulation of the Lambek calculus which is based on the *sequent calculus* notation of Gentzen (1936). Sequents are objects of the form:

$$\Gamma \Rightarrow x$$

where $\Rightarrow$, the derivability relation, indicates that the *succedent* x can be derived from the *antecedents* Γ. Thus, each sequent corresponds to an entire proof of the natural deduction system. For the Lambek calculus, we require that each sequent must have precisely one succedent type and at least one antecedent type.

The following inference rules and axiom scheme, from Lambek (1958), provide a sequent formulation for the (product-free) Lambek calculus:

(2.11) Axiom: $$x \Rightarrow x$$

Right rules: $$\frac{\Gamma, y \Rightarrow x}{\Gamma \Rightarrow x/y}[/R] \qquad \frac{y, \Gamma \Rightarrow x}{\Gamma \Rightarrow x\backslash y}[\backslash R]$$

Left rules: $$\frac{\Delta \Rightarrow y \qquad \Gamma, x, \Lambda \Rightarrow z}{\Gamma, x/y, \Delta, \Lambda \Rightarrow z}[/L] \qquad \frac{\Delta \Rightarrow y \qquad \Gamma, x, \Lambda \Rightarrow z}{\Gamma, \Delta, x\backslash y, \Lambda \Rightarrow z}[\backslash L]$$

Cut rule: $$\frac{\Delta \Rightarrow x \qquad \Gamma, x, \Lambda \Rightarrow y}{\Gamma, \Delta, \Lambda \Rightarrow y}[cut]$$

The axiom scheme states the form of sequents whose correctness is not questioned, and essentially expresses the notion that any type derives itself. The inference rules specify the correctness of the sequent that appears below the line (the conclusion) given the correctness of the sequent(s) appearing above the line (the premises). The *left* and *right* rules (marked L and R, respectively) are *logical*, governing the behaviour of a single connective which is the principal connective of some antecedent or succedent type (referred to as the *active type*). The naming of these rules as 'left' or 'right' refers to which side of the derivability relation $\Rightarrow$ that the active type for the rule appears. The left rules correspond to the elimination rules of the natural deduction formulation, the right rules to the introduction rules. As with the natural deduction formulation, a cut rule is provided, expressing the transitivity of derivability.

Lambek (1958) demonstrates that this particular sequent formulation of the calculus has the property of *cut elimination*, i.e. that every sequent proof based on these rules can be transformed into one having the same end-sequent but containing no uses of the cut rule. It follows that any sequent which is a theorem under the formulation in (2.11) may be proven without the cut rule. Top-down search for proofs of any sequent using just the remaining rules is guaranteed to terminate. This is because, firstly, the premise(s) of each rule are (in sum) simpler than the conclusion (under a metric which counts the number of connectives), and secondly, premises are constructed using only subformulas of the types occurring in the conclusion. Thus, theoremhood is *decidable*. It is for this reason that the sequent formulation is of value for parsing, which may handled as an instance of automated theorem proving.

The proof (2.12), again for simple composition, illustrates the sequent method. Comparing this proof with that in (2.7), the advantages of the natural deduction formulation over the sequent formulation for general presentation should be obvious to the reader.

(2.12)
$$\cfrac{\cfrac{\mathrm{c} \Rightarrow \mathrm{c} \qquad \cfrac{\mathrm{b} \Rightarrow \mathrm{b} \quad \mathrm{a} \Rightarrow \mathrm{a}}{\mathrm{a/b,\ b} \Rightarrow \mathrm{a}}[/L]}{\mathrm{a/b,\ b/c,\ c} \Rightarrow \mathrm{a}}[/L]}{\mathrm{a/b,\ b/c} \Rightarrow \mathrm{a/c}}[/R]$$

Moortgat (1988) shows how the sequent formulation may be extended to provide the semantics for valid type combinations. As before, this involves associating each type with a lambda expression, its *proof term*. This gives rise to the following formulation:

(2.13) Axiom: $\mathrm{x}{:}f \Rightarrow \mathrm{x}{:}f$

Right rules:

$$\frac{\Gamma,\ \mathrm{y}{:}i \Rightarrow \mathrm{x}{:}f}{\Gamma \Rightarrow \mathrm{x/y}{:}\lambda i.f}[/\mathrm{R}] \qquad \frac{\mathrm{y}{:}i,\ \Gamma \Rightarrow \mathrm{x}{:}f}{\Gamma \Rightarrow \mathrm{x\backslash y}{:}\lambda i.f}[\backslash \mathrm{R}]$$

Left rules:

$$\frac{\Delta \Rightarrow \mathrm{y}{:}g \qquad \Gamma,\ \mathrm{x}{:}fg,\ \Lambda \Rightarrow \mathrm{z}{:}h}{\Gamma,\ \mathrm{x/y}{:}f,\ \Delta,\ \Lambda \Rightarrow \mathrm{z}{:}h}[/\mathrm{L}] \qquad \frac{\Delta \Rightarrow \mathrm{y}{:}g \qquad \Gamma,\ \mathrm{x}{:}fg,\ \Lambda \Rightarrow \mathrm{z}{:}h}{\Gamma,\ \Delta,\ \mathrm{x\backslash y}{:}f,\ \Lambda \Rightarrow \mathrm{z}{:}h}[\backslash \mathrm{L}]$$

Cut rule:

$$\frac{\Delta \Rightarrow \mathrm{x}{:}f \qquad \Gamma,\ \mathrm{x}{:}f,\ \Lambda \Rightarrow \mathrm{y}{:}g}{\Gamma,\ \Delta,\ \Lambda \Rightarrow \mathrm{y}{:}g}[\mathrm{cut}]$$

2.3 Extraction and flexible categorial grammars

We next consider a general approach to handling extraction that is widely used in flexible CG work, which depends crucially on flexible type assignment. Treatments of extraction along these lines derive with greater or lesser degree of modification from the proposals of Ades and Steedman (1982).

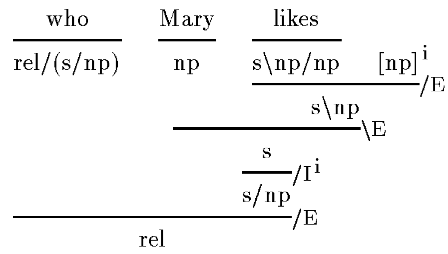
Flexible type assignment in many cases allows that, where some subconstituent is missing (i.e. extracted) from some domain (the 'extraction domain'), the remainder of the material that goes to make up that domain can be combined to give a single constituent. More specifically, if the missing item is of type B and the extraction domain is one which would in general be assigned a type A, then in the absence of the missing constituent, the remainder of the domain may be combined to give a type which is a function from B to A. Given this, a fronted item may be assigned a type which subcategorizes for such a function. Such a type for the fronted item can be seen to, in effect, 'abstract over' the missing element in the extraction domain.

Consider, for example, the case of relativization. In a relative clause such as *who Mary likes*, the substring *Mary likes* corresponds to a clause from which a NP has been extracted. As we shall see, this substring can be analysed of type s/np, and so we can allow for such cases by assigning the relative pronoun a lexical type such as:

rel/(s/np)

where rel is used to abbreviate the type of the relative clause (usually taken to be a noun modifier). This type allows a derivation for the relative clause *who Mary likes* as follows:

(2.14)(a man)



This proof assigns the following meaning:

who′ (λx.likes′ x mary′)

Observe that the semantic argument position corresponding to the missing NP is abstracted over in the semantics assigned for its extraction domain (i.e. the embedded clause).

2.4 Extended Lambek calculus

Despite its appealing elegance and simplicity, the standard Lambek calculus is clearly not adequate as a linguistic framework since it is not powerful enough to describe a range of natural language phenomena. Examples of this arise in respect of linguistic phenomena involving word order variation, and phenomena involving optionality and iteration of complementation. Other problems for the Lambek calculus arise in respect of phenomena which exhibit *locality constraints* such as, for example, *island constraints*.

Recently, various proposals have been made for dealing with these shortcomings of the standard Lambek calculus. These proposals involve augmenting the basic calculus with a range of further operators whose logical behaviour is such as to allow description of some aspect of the linguistic data. For example, Morrill, Leslie, Hepple and Barry (1990) suggest how the use of operators called *structural modalities*, borrowed from linear logic (Girard, 1987), may provide a basis for handling phenomena involving word order variation, optionality and iteration. Related proposals in respect of some of these possibilities are also made by Moortgat (1990c). Morrill (1989) suggests an operator with the logical behaviour of necessity, and discusses how it may allow for the treatment of phenomena involving linguistic boundaries. Some further possibilities include first-order quantification over features (Morrill, 1990a), second-order quantification over types (van Benthem, 1989; Emms, 1990) and the use of boolean operators of union and intersection (Moortgat, 1990c; Morrill 1990a).

We term the general approach that arises given the availability of these augmentations the *extended Lambek calculus framework*. In what follows, we outline some additions to the standard framework which are required for the presentation of the ensuing chapters.

2.4.1 Linguistic boundaries as modality

Morrill (1989,1990b) suggests the use of a (unary) operator $\Box$, corresponding to *necessity* in modal logics, whose utility is twofold.[4] Firstly, Morrill takes $\Box$ to be an *intensional* type-forming operator, of use in handling intensionality in natural language semantics. Secondly, Morrill suggests that $\Box$ may provide a basis for handling locality effects on extraction and reflexivization. We shall not pursue Morrill's proposals with regard to intensionality in this thesis.[5] Instead, we assume that the role of this operator is simply in specifying linguistic boundaries, with $\Box$ having no semantic import, and its inference rules having no semantic effect. In what follows we illustrate how $\Box$ may allow the specification of linguistic boundaries by discussing its use in respect of the boundedness of extraction. Note, however, that this discussion is purely for illustration and is not intended to correctly represent the details of Morrill's account (which is discussed in the next chapter), or our own view of how $\Box$ should be used in the linguistic treatments.

We begin by considering the inference rules of $\Box$ and the behaviour they give rise to. In (2.15) we see the elimination rule for $\Box$, together with a provisional version of its introduction rule (to be revised shortly).

(2.15) Rules for $\Box$:

$$\frac{\begin{array}{c}\vdots\\ \Box A{:}x\end{array}}{A{:}x}\Box E \qquad \frac{\begin{array}{c}\vdots\\ A{:}x\end{array}}{\Box A{:}x}\Box I$$

where every undischarged assumption in the proof of A is a $\Box$-type

Let us consider why an operator which has these inference rules is appropriately viewed as a modal operator. The elimination rule allows a non-modal type (A) to be derived from its modal variant ($\Box$A). This realizes the intuitively reasonable idea about necessity that 'necessarily X' implies X (an implication which is taken as an axiom in many modal logics, the so-called *axiom of Necessity*). The introduction rule realizes a general principle that whatever follows from a necessary truth must itself be necessarily true, i.e. allowing that any conclusion which can be drawn on the basis of necessary assumptions may itself be taken as necessary. In practice, it is convenient to generalize the $\Box$ introduction rule with respect to the cut rule, so that instead of requiring necessary *assumptions,* $\Box$ introduction may apply provided that the proof is based on a number of *independent subproofs* of $\Box$ types. We can state such a version of the rule as follows:

[4] Note that the inclusion of this operator requires that we must extend the definition of the set of types so that for all types X, $\Box$X is also a type. Similar extensions of the definition of the set of types are required in respect of the other operators used in the remainder of this chapter and subsequent chapters, although we will not generally state the (obvious) extension of the definition of types that they require.

[5] In the next chapter, in fact, we will note some arguments *against* assuming such a semantic role for the $\Box$ operator.

(2.16) $$\dfrac{\begin{matrix}\vdots\\ \text{A:x}\end{matrix}}{\Box\text{A:x}}\Box\text{I}$$ where every path to an undischarged assumption in the proof of A leads to an independent subproof of a □-type

The sequent rules for □ are as in (2.17) (where $\Box x_1, \Box x_2, \ldots, \Box x_n$ indicates a sequence of boxed types):

(2.17) $$\dfrac{\Gamma,\ x,\ \Lambda \Rightarrow y}{\Gamma,\ \Box x,\ \Lambda \Rightarrow y}[\Box L] \qquad \dfrac{\Box x_1, \Box x_2, \ldots, \Box x_n \Rightarrow x_0}{\Box x_1, \Box x_2, \ldots, \Box x_n \Rightarrow \Box x_0}[\Box R]$$

The inference rules for □ give rise to a system in which □ behaves similarly to the necessity operator in the modal logic S4. This similarity is revealed by the fact that the following type transitions are allowed:

(2.18) a. $\Box X \Rightarrow X$

b. $\Box(X/Y) \Rightarrow \Box X/\Box Y$

c. $\Box X \Rightarrow \Box\Box X$

These type transitions correspond to the characteristic axioms of the modal logic S4. Note, however, that the correspondence of this system to S4 is limited, not least since our system lacks negation. This prevents the interdefinability of necessity and possibility, an important characteristic of most standard modal logics, including S4.

To be able to prove modal result types from combinations of lexical types, we must allow that lexical types are themselves modal. For present purposes, we adopt the following scheme for marking □ operators on lexical types. Firstly, we assume that each lexical type is marked with an overall □ operator, i.e. is of the form □X, for some X. For example, a lexical noun phrase has type □np, and a lexical intransitive verb has type □(s\np). Secondly, some functional types may in addition seek a modal type for their argument, which serves the purpose of specifying that this argument constitutes a bounded domain (in a manner we will come to shortly). For example, a sentence embedding verb such as *believe* would have type □(s\np/□s). The following proof illustrates the use of the □-inference rules:

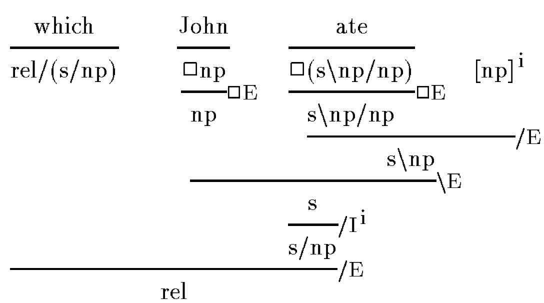
(2.19) $$\dfrac{\dfrac{\dfrac{\text{Mary}}{\Box np}}{np}\Box E \quad \dfrac{\dfrac{\dfrac{\text{believes}}{\Box(s\backslash np/\Box s)}}{s\backslash np/\Box s}\Box E \quad \dfrac{\dfrac{\dfrac{\dfrac{\text{John}}{\Box np}}{np}\Box E \quad \dfrac{\dfrac{\text{smells}}{\Box(s\backslash np)}}{s\backslash np}\Box E}{s}\backslash E}{\Box s}\Box I}{s\backslash np}/E}{s}\backslash E$$

Note that the use of [□I] in (2.19) depends on the fact that all (i.e. both) of the hypotheses for the proof of the embedded sentence are □-types.

The treatment of boundedness that this apparatus allows depends on the fact that, in general, we cannot abstract out of a modal domain by discharging a non-modal assumption. This is because the subproof for a modal domain must end with a [□I] inference, but the presence of an undischarged non-modal assumption in the proof will usually mean that the side condition on the rule is not met. In fact this is not necessarily the case since the condition in (2.16) refers to independent subproofs of □-types rather than □-type assumptions. However, if the marking of □ operators on lexical types is restricted to the pattern described above, the side condition on [□I] will only be satisfied when all the undischarged *assumptions* in the subproof are □-types.[6]

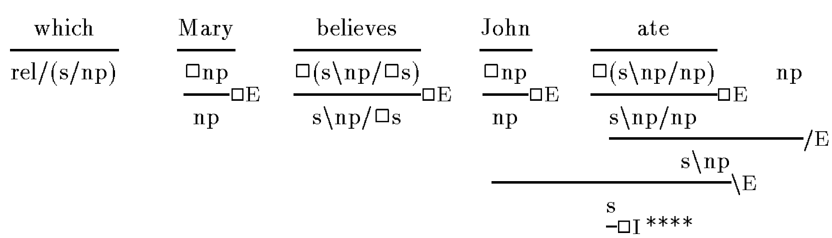
We can illustrate this approach to boundedness by considering extraction using a relative pronoun type rel/(s/np). This type seeks a complement which is a clause 'lacking' a *non-modal* NP. To prove such a type, we must assume an additional hypothesis of (non-modal) type np, as in the following proof:

(2.20)



However, in the case of extracting out of a (modal) embedded sentence, the presence of this *undischarged* non-modal assumption blocks the application of the □-introduction rule (i.e. the starred inference in (2.21)). Since the matrix verb requires a modal type for its argument, the proof cannot be completed:

(2.21)

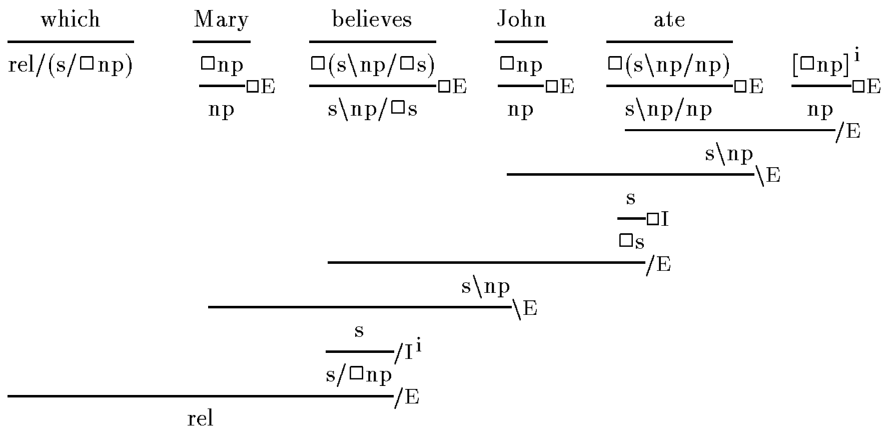


In contrast, with the relative pronoun type rel/(s/□np), extraction out of an embedded modal sentence *is* possible because then the additional assumption must be of type □np,

[6] In later chapters, we consider cases where diverging from the pattern of □-marking described above might be of use.

and so its presence does not block [□I]:

(2.22)



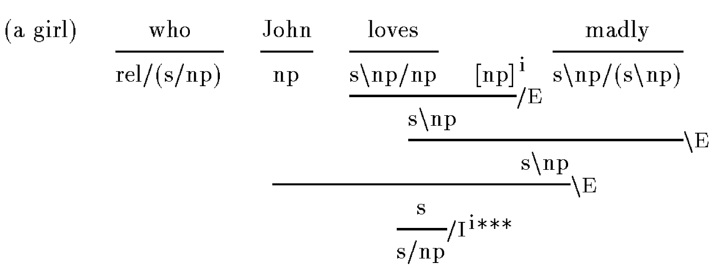
Hopefully this discussion has illustrated how □ may provide a basis for handling locality constraints. Again, we must emphasize that this discussion does not present either our own or Morrill's proposals of how island constraints should be handled. That topic is discussed extensively in the next chapter.

2.4.2 Permutation and non-peripheral extraction

A problem arises for a number of flexible CG frameworks, including the present one, for handling what has been termed *non-peripheral* extraction, i.e. extraction from anywhere other than the leftmost or rightmost position within an extraction domain. In some other frameworks, solutions for this problem have been proposed which are based around the inclusion of so-called *disharmonic* rules of composition (e.g. Steedman, 1987; Moortgat, 1988). Such proposals will be critically discussed in the next chapter. In this section, we discuss the problem that non-peripheral extraction presents for the Lambek calculus, and we consider a solution to this problem proposed in Morrill, Leslie, Hepple & Barry (1990), which is based around the use of *structural modalities.*

Consider the following attempt to prove a relative clause involving non-peripheral extraction (we ignore for the moment the involvement of boundary modalities):

(2.23)



Note that this proof invokes an additional assumption corresponding to the (extracted) object of the verb. For the proof to be completed, this assumption would need to be discharged in a [/I] inference. However, this inference (starred in (2.23)) is not properly licensed, because the rule has a side condition which requires that the assumption which is discharged must be the rightmost undischarged assumption in the proof. In general, the side conditions on the slash introduction rules only allow for abstraction over left and right peripheral elements in a proof, and so our approach as its stands is unable to handle non-peripheral extraction.

Morrill, Leslie, Hepple & Barry (1990) suggest a solution to this problem which is based around the use of what are termed *structural modalities*, an idea borrowed from linear logic (Girard, 1987). Various logics contain *structural rules*, which manipulate the order or number of premises in a proof rather than introduce or eliminate some logical connective. One possibility here is rules of *permutation* which reorder the premises of a proof. (2.24a) shows a permutation rule as this might be stated for a natural deduction formulation, whilst (2.24b) shows a sequent permutation rule.

(2.24)(a) $\dfrac{X \quad Y}{Y \quad X}P$ (b) $\dfrac{\Gamma, y, x, \Delta \Rightarrow Z}{\Gamma, x, y, \Delta \Rightarrow Z}P$

The inclusion of such rules gives rise to a *commutative* system, i.e. one in which the relative order of premises is not important. Such a system is clearly not appropriate for linguistic description. Proposals within linear logic (Girard, 1987) suggests how the effects of structural rules may be introduced in a controlled fashion. Linear logic makes use of operators called *exponentials* whose behaviour resembles that of the modalities necessity and possibility in modal logics, but which have associated inference rules corresponding to the structural rules of *weakening* and *contraction* (which allow premises to be ignored or utilized more than once).

Morrill, Leslie, Hepple & Barry (1990) suggest a number of *structural modality* operators for use in a linguistic calculus (both necessity and possibility operators) which provide for controlled involvement of structural rules. These structural modalities are of use in relation to phenomena involving reordering, iteration and optionality.[7] Only one of these possibilities is of relevance for the present work, a permutation operator $\triangle$ having the modal behaviour of necessity. This has the following natural deduction inference rules:

[7] See Moortgat (1990c) for related proposals in respect of some of these possibilities.

(2.25) Rules for $\triangle$:

$$\frac{\triangle A{:}x \quad B{:}y}{B{:}y \quad \triangle A{:}x}\triangle P \qquad \frac{A{:}y \quad \triangle B{:}x}{\triangle B{:}x \quad A{:}y}\triangle P$$

$$\frac{\begin{array}{c}\vdots\\ \triangle A{:}x\end{array}}{A{:}x}\triangle E \qquad \frac{\begin{array}{c}\vdots\\ A{:}x\end{array}}{\triangle A{:}x}\triangle I$$

where every path to an undischarged assumption in the proof of A leads to an independent subproof of a $\triangle$-type

Note that the introduction and elimination rules for $\triangle$ are essentially the same as those for $\Box$, and that $\triangle$ is for this reason another necessity modality with S4-like behaviour. The two permutation rules allow that types marked with this modality may permute both to the left and right, and so this modality is a *bidirectional* permutor.[8] These permutation rules in effect override the significance of the linear order of the initial premises of a proof in so far as these are $\triangle$-types. Thus, a sequence of types Γ is essentially equivalent to any alternative sequence Γ' of the same types which differs only in the position of any $\triangle$-types. Note that for such equivalent sequences Γ and Γ', all the elements which are not $\triangle$-types must occur in the same relative order. This operator provides for *controlled* overriding of ordering information without undermining ordering altogether.

Note one difference of the permutation rules in (2.25) to the rules we have seen so far that these are *multiply conclusioned.* It is standard in Lambek work to assume that in all permissible derivations, a *single* conclusion type is derived from one or more premises (although this contrasts with the situation in many other logics). Since we wish to maintain this restriction, we stipulate that any *complete* proof must be single-conclusioned. It follows that a permutation inference may not be the final inference in any complete proof.

The sequent rules for $\triangle$ are as follows:

(2.26)

$$\frac{\Gamma,\ x,\ \Lambda \Rightarrow y}{\Gamma,\ \triangle x,\ \Lambda \Rightarrow y}[\triangle L] \qquad \frac{\triangle x_1,\ \triangle x_2, \ldots,\ \triangle x_n \Rightarrow x_0}{\triangle x_1,\ \triangle x_2, \ldots,\ \triangle x_n \Rightarrow \triangle x_0}[\triangle R]$$

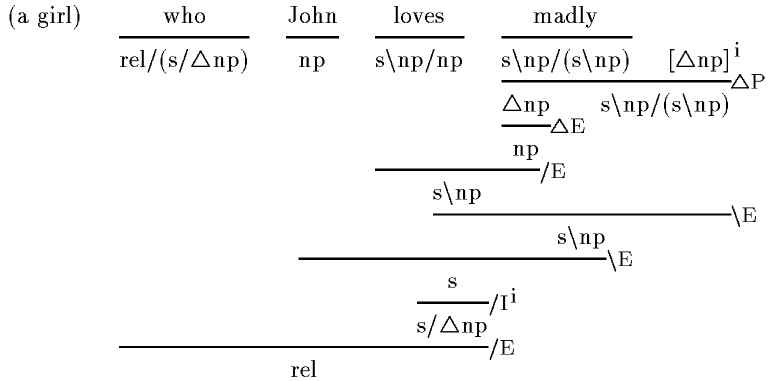
$$\frac{\Gamma,\ \triangle x,\ y,\ \Lambda \Rightarrow z}{\Gamma,\ y,\ \triangle x,\ \Lambda \Rightarrow z}[\triangle P] \qquad \frac{\Gamma,\ x,\ \triangle y,\ \Lambda \Rightarrow z}{\Gamma,\ \triangle y,\ x,\ \Lambda \Rightarrow z}[\triangle P]$$

The value of this operator is perhaps best seen by looking at an example. Consider the relative pronoun type rel/(s/$\triangle$np). This allows the following proof for the example considered earlier:

---

[8] The presentation in Morrill, Leslie, Hepple & Barry (1990) differs slightly in that two *unidirectional* permutors $\rhd$ and $\lhd$ are defined, each of which can permute in only one direction.

(2.27)



Since the relative pronoun requires an argument type s/△np, the proof is made using an additional assumption of type △np. This additional assumption must initially appear right peripherally so that the [/I] inference is properly licensed. However, because the type is marked with a △ operator, it may permute along through the sequence of assumptions until it arrives at the appropriate position for the missing element. Since the verb requires a bare np type for its argument (rather than △np), an elimination inference is made which strips off the △ operator, enabling the combination of verb and assumption type. Note that the initial position of the △-marked assumption in (2.27) is selected only in terms of the requirements of the [/I] rule, and should not be taken to be of any significance in terms of linguistic order. The position that *is* of note with regard to standard word order requirements is the position at which the [△E] inference is made, since it corresponds to the usual position of the extracted item.

We use the △ operator exclusively in the treatment of extraction. In practice, any assumption type of the form △X that appears in a sequence of premises is one used in the service of some extraction analysis, and will usually be discharged in the overall analysis. A function type of the form X/△Y or X\△Y is one whose argument is undergoing extraction. Whereas a function X/Y might be loosely characterized as an "X lacking a Y at its rightmost edge", a function X/△Y could be described as an "X lacking a Y *at some position*". In general, the appearance of a △-type in a proof of some combination of lexical types is licensed by the presence of some lexical extractor having a 'higher order' type such as e.g. X/(Y/△Z).

Note the type transition in (2.28a), which corresponds to a step of establishing that the argument of a function is one undergoing extraction (loosely corresponding to the slash-termination step of Generalized Phrase Structure Grammar (GPSG, Gazdar *et al*, 1985)). Note that the reverse transition, whereby a △-marked argument loses its △ operator, is *not* possible, with the effect that an argument that is identified as one undergoing extraction may not lose this status. The type transitions (2.28b,c) illustrate the inheritance of △-marked arguments onto the result of a combination (loosely akin to GPSG slash-percolation). The type transitions (2.28d,e) show that the slash directionality of a △-marked argument is freely changeable.

(2.28) a. $X/Y \Rightarrow X/\triangle Y$

b. $X/Y/\triangle Z,\ Y \Rightarrow X/\triangle Z$

c. $X/Y,\ Y\backslash\triangle Z \Rightarrow X\backslash\triangle Z$

d. $X/\triangle Y \Rightarrow X\backslash\triangle Y$

e. $X\backslash\triangle Y \Rightarrow X/\triangle Y$

f. $X/\triangle(Y/(Y\backslash Z)) \Rightarrow X/\triangle Z$

For the creation and inheritance of gap-marked arguments, only the permutation and elimination rules are required. The inclusion of the introduction rule, however, is not superfluous; its presence allows for certain transitions that would otherwise not be available. Although we won't dwell on the value of this additional flexibility here, we note as a single example that the type transition (2.28f), corresponding to argument lowering, is possible only given the introduction rule's inclusion.

2.4.3 Boolean union

The use of boolean type-forming connectives of *intersection* and *union* has been suggested (Moortgat, 1990c; Morrill 1990a). We require only the union connective ∪, which allows construction of types such as X∪Y, i.e. the union of the types X and Y. This has the twin introduction rules shown in (2.29a):

(2.29)(a) $\dfrac{\begin{matrix}\vdots\\ X\end{matrix}}{X\cup Y}\cup I \qquad \dfrac{\begin{matrix}\vdots\\ Y\end{matrix}}{X\cup Y}\cup I$ (b) $\dfrac{\begin{matrix}\vdots\\ X{:}f\end{matrix}}{X\cup Y{:}f}\cup I \qquad \dfrac{\begin{matrix}\vdots\\ Y{:}f\end{matrix}}{X\cup Y{:}f}\cup I$

i.e. given a proof of a type X (or Y), we can construct a proof of a type X∪Y. The corresponding elimination rule is hard to state in the natural deduction system (though it can be straightforwardly stated in the sequent system). However, the introduction rule is all we require for the accounts to be given. The rules are shown with semantics in (2.29b). We take these rules to have no associated semantic transformation (i.e. each rule has 'identity' semantics), and this is sufficient for our purposes (although note that more complicated semantics schemes have been suggested (e.g. Morrill, 1990a)).

The corresponding sequent rules for ∪ are shown (2.30). The type transitions (2.31) illustrate the behaviour of the ∪ operator.

(2.30) $\dfrac{\Gamma \Rightarrow \mathsf{x}{:}f}{\Gamma \Rightarrow \mathsf{x}{\cup}\mathsf{y}{:}f}[\cup R] \qquad \dfrac{\Gamma \Rightarrow \mathsf{y}{:}f}{\Gamma \Rightarrow \mathsf{x}{\cup}\mathsf{y}{:}f}[\cup R]$

(2.31) a. X ⇒ X∪Y

b. Y ⇒ X∪Y

c. X/(Y∪Z), Y ⇒ X

d. X/(Y∪Z), Z ⇒ X

2.5 Conclusion

We have considered natural deduction and sequent formulations of the Lambek calculus, and some possible augmentations of the Lambek calculus. This list of augmentations should not be taken to be exhaustive. We have considered only a few possibilities that we require for the accounts to be offered in the rest of the thesis. Some further possibilities will be introduced in context during the presentation of these accounts. The development of the extended Lambek framework is a project that has only recently begun. The precise form of the extended calculus that is required to provide for the task of natural language analysis is something that can only be determined during considerable further linguistic research.

Chapter 3

Islands and Extraction

This chapter focuses on the treatment of extraction. There are two major aspects to the discussion. Firstly, the treatment of extraction from non-peripheral positions (i.e. from positions that are not peripheral in the overall constituent from which the extracted element is missing), which we call *non-peripheral extraction*. Secondly, the treatment of constraints on extraction, and in particular *subjacency* effects which result from the presence of linguistic boundaries, giving rise to so-called *islands*.

The treatment of non-peripheral extraction is of interest here because it has presented a serious (though essentially just technical) problem for categorial grammars. The apparatus that we use for non-peripheral extraction was introduced in Chapter 2, the essential ingredient being the permutation structural modality $\triangle$. What we discuss in this chapter is the problems that arise for alternative approaches to handling the problem of non-peripheral extraction. In particular, we look at treatments of non-peripheral extraction based on the inclusion of *disharmonic* (or *slash-mixing*) rules of composition. The problems that arise for such approaches serve to motivate indirectly the approach we take for non-peripheral extraction.

The larger part of the chapter is addressed to island constraints on extraction. We consider some previous attempts to deal with islands in categorial frameworks. Such attempts include what we term *restrictional* accounts of island constraints, which are based around the stipulative exclusion of certain (generally valid) type combination or type transition possibilities. A second approach we consider is based on the use of the $\Box$ operator, which was described in Chapter 2. As we shall see, both of these approaches run into serious problems. We take these problems to indicate the need for a treatment of subjacency that goes beyond a simple unitary notion of 'linguistic boundary', and such an approach is developed in terms of a polymodal logic of necessity.

3.1 The treatment of non-peripheral extraction

We saw in the preceding chapter how flexible CGs allow a treatment of extraction whereby a constituent C from which some element E is missing (i.e. has been extracted) can be assigned a type that is a function from E to C. The semantics of this constituent will involve functional abstraction over the missing element. An 'extracted item' can then be assigned a type which is function over such a constituent, a type which in effect 'abstracts over' the missing element in the extraction domain. Such accounts derive from the proposals of Ades and Steedman (1982) with greater or lesser extents of modification.

The standard Lambek calculus only allows for the treatment of *peripheral* extraction (i.e. extraction of a leftmost or rightmost subconstituent within some domain) given the side condition on the introduction rules. This point was illustrated in Chapter 2, where the structural modality required to overcome this problem was presented. The reader is referred to that chapter for illustration of the method. Such a solution gives an approach in which extracted arguments have a 'distinguished' status. Thus, a function of the form $x/\triangle y$ is one whose argument is being extracted, and extracted items are assigned types of the form $z/(x/\triangle y)$.

3.1.1 Disharmonic composition and non-peripheral extraction

Alternative treatments for non-peripheral extraction within FCG frameworks have been proposed. We shall consider some proposals which embody an assumption that the processes that give rise to extraction and other non-canonical structures are essentially the same as those that give rise to canonical structures. This assumption is in part realized as a limitation to using the standard categorial connectives / and \. We shall consider some accounts of non-peripheral extraction which embody this assumption that are based on the use of what have been called *slash-mixing* or *disharmonic* rules of composition. Proposals for the use of disharmonic composition have been made by Morrill (1987), Steedman (1987) and Moortgat (1988), amongst others.

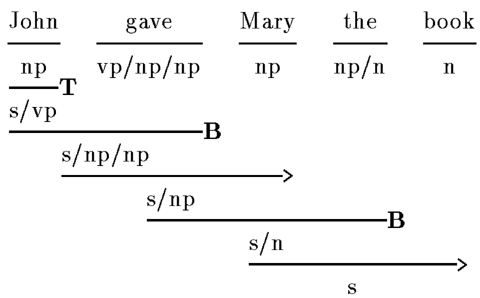
The proposals of Steedman (1987) are set within the Combinatory Categorial Grammar (CCG) framework. We shall use a simplified version of CCG as a basis for presenting proposals for the use of disharmonic composition of Steedman and others. A CCG formulation involves a set of rules, including rules of functional application (corresponding to our slash elimination rules), together with some further type transition/combination schemas.[1]

[1] In CCG, these additional type combination schemas are related in terms of their associated semantic operations to *combinators* which are primitive operations for combining functional expressions. See Curry and Feys (1958) and Hindley and Seldin (1986) for discussion of combinators and combinatory logics, and

For example, the following rules of type-raising, 'simple' composition and 'second order' composition are typically included:

(3.1)	(>)	$X/Y,\ Y \Rightarrow X$	
	(<)	$Y,\ X\backslash Y \Rightarrow X$	
	(**T**)	$X \Rightarrow Y/(Y\backslash X)$	$X \Rightarrow Y\backslash(Y/X)$
	(**B**)	$X/Y,\ Y/Z \Rightarrow X/Z$	$Y\backslash Z,\ X\backslash Y \Rightarrow X\backslash Z$
	($\mathbf{B}^2$)	$X/Y,\ Y/Z/W \Rightarrow X/Z/W$	$Y\backslash Z\backslash W,\ X\backslash Y \Rightarrow X\backslash Z\backslash W$

Note that these type combination schemas are all valid within the Lambek calculus. (The interested reader may derive the associated semantic recipes for these rules via their Lambek natural deductions.) The use of these rules is illustrated in the following proof:[2]

(3.2)


The following shows a case of non-peripheral extraction:

(3.3) (the cake) which John ate ∅ quickly

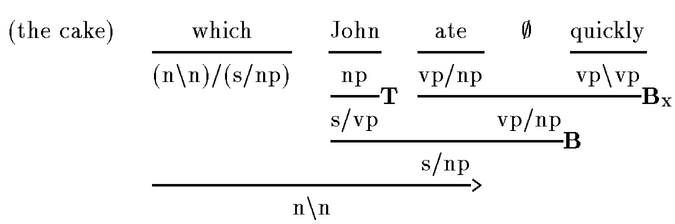
(the cake)	which	John	ate	∅	quickly	
	(n\n)/(s/np)	np	vp/np		vp\vp	⇒ (n\n)

The indicated type combination is not valid under the Lambek calculus, and receives no proof under the natural deduction formulation. Furthermore, the type combination cannot be made using the (Lambek valid) CCG combination rules in (3.1).

It has been proposed that examples such as (3.3) can be handled by inclusion of the following disharmonic composition rule:

(3.4) ($\mathbf{B}_\mathrm{x}$) $Y/Z,\ X\backslash Y \Rightarrow X/Z$

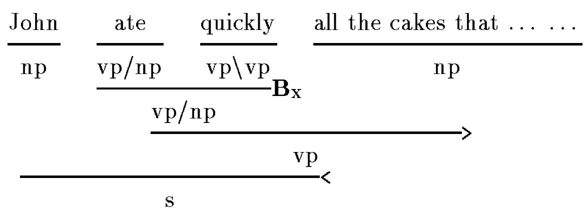
The rule is termed 'disharmonic' since the directionalities of the main and subordinate functors are not 'in harmony' (in contrast with the Lambek valid composition rules shown in (3.1)). This rule allows the type combination (3.3) to be made as follows:

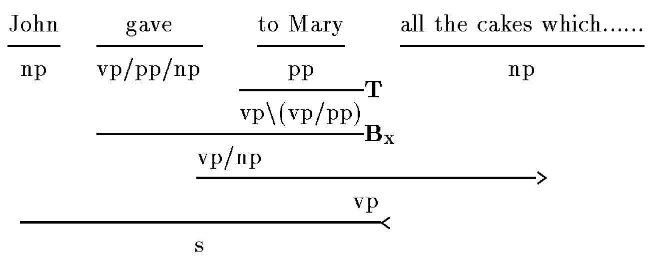
(3.5)


Steedman (1988) for discussion of combinators in CCG.

[2] We use vp to abbreviate the type of VPs, which for CCG is (s\np).

This rule is also suggested for the analysis of right extraction or extraposition. The following two derivations illustrate the treatment of heavy NP shift. In (3.6), the NP moves past an adverb, and in (3.6) past a prepositional object:

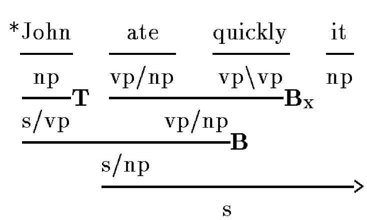
(3.6) 

(3.7) 

3.1.2 Problems for disharmonic composition based accounts

We next consider some problems that arise for accounts that depend on the use of disharmonic composition. Moortgat (1988) shows that adding a rule of disharmonic composition to the Lambek calculus results in permutation closure for all sequences of at least three types, i.e. that for any theorem $x_1,\ldots,x_n \Rightarrow x_0$, where $n \geq 3$, the type x_0 can also be derived for any permutation of $x_1,\ldots,x_n$. This result obviously rules out the inclusion of an unrestricted rule of disharmonic composition for the Lambek calculus. However, this result does not carry over to systems such as CCG where a finite set of type combination schemas is provided and where restrictions upon what types may instantiate any combination schema are allowed. Even so, a number of problems do arise.

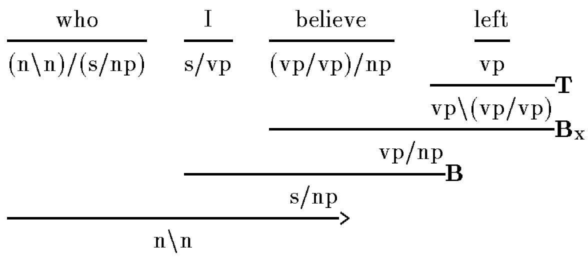
Bouma (1987) points out that that under this account of heavy NP shift there is no way of ensuring that the right extracted NP actually is 'heavy'. The grammar fails to distinguish the acceptable case in (3.6) from the following clearly unacceptable one:

(3.8) 

It is a central assumption of categorial approaches that constituents express their combinatory potential explicitly in their type. We expect that any two constituents that bear the same type will behave alike. The type vp/np allowed for the string *ate quickly* is the same as that of a lexical transitive verb such as *liked*, and there seems no way that we may restrict one but not the other to combine with only heavy NPs without violating this central assumption.

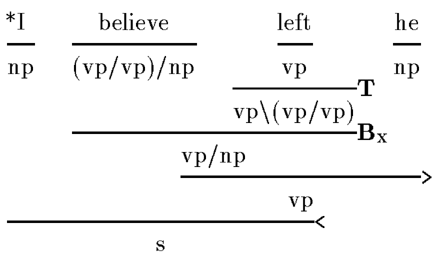
A similar problem arises for the account of embedded subject extraction presented in Steedman (1987).[3] Steedman suggests that sentence complement verbs such as *believe* be given (amongst others) a type $(vp/vp)/np_{[nom]}$, i.e. with the verb separately seeking the subject and VP of its embedded complement. This allows that embedded subject extraction can be analysed in line with the extraction of first objects in double object constructions (as illustrated in (3.7)), i.e:

(3.9)



However, as Steedman himself notes, this treatment also allows right extraction of an embedded subject across its verb as in:

(3.10)



Such examples are grossly unacceptable and even the use of a 'heavy' subject NP does little to soften this impression.[4]

The problems we have described all involve inappropriate 'filling' for an extracted argument. For a flexible CG approach which uses just the standard slash connectives, an argument may correspond equally to either ordinary subcategorization or to a complement that is undergoing extraction. The failure to distinguish extracted and non-extracted arguments makes it impossible to avoid inappropriate 'filler introduction'.

[3] In this paper, Steedman discusses the possibility of allowing for embedded subject extraction using forward slash-mixing composition but rejects this option because it threatens to result in unacceptable overgeneration. For example, it would allow leftward drift of subjects as in **I he think left.*

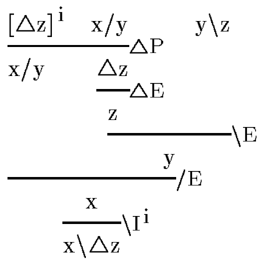
[4] Steedman offers sentence (i) as a not too bad case of 'subject extraposition' admitted by his grammar, and contrasts it with the worse non-embedded case (ii) which his grammar does not admit:
(i) ?Brutus implied was no good each Caesar whom he ostensibly praised.
(ii) *Was no good each Caesar whom he ostensibly praised.

3.1.3 Distinguishing extracted arguments

For our account, the structural modality $\triangle$ plays the role of distinguishing extracted from non-extracted arguments. This operator allows for combinations which clearly correspond to disharmonic composition (such as (3.11), proven in (3.12)), but avoid the same problems of overgeneration since the 'reordered' argument is marked as extracted, preventing inappropriate filler introduction.

(3.11) $\quad x/y,\ y\backslash z \Rightarrow x\backslash\triangle z$

(3.12)



One interesting question is the relation of this treatment of extraction to that offered in Head-driven Phrase structure Grammar (HPSG - Pollard and Sag, 1987). HPSG is a unification-based grammar which combines ideas from phrase structure grammar (principally GPSG) and categorial grammar. The subcategorization requirement of words and phrases is explicitly stated by means of a list-valued subcategorization feature. Constituents are combined using 'phrase structure rules' (which access the subcategorization information of the head element), and the linear order of the daughters of any projection is determined using linear precedence statements. The most obvious relation of this approach to CG is the way that HPSG heads specify explicit and *ordered* subcategorization information. HPSG extraction is handled in a GPSG-style gap-feature passing method, using a list valued (or in some work, set valued) feature. Gap 'introduction' involves a step of moving subcategorization information from the subcategorization list to the 'gap' list. A similarity to our approach then is that subcategorization information of extracted complements is in some way 'distinguished' from other subcategorization information. Despite this, our approach is far more closely akin to other flexible CG treatments of extraction than that of HPSG. Unlike HPSG, an argument is marked as extracted using the $\triangle$ operator, *not* by separating it off from other (non-extracted) subcategorization information. Such 'marked' arguments participate in the *same* ordering over subcategorization as non-extracted arguments, in contrast to HPSG where identifying some complement as extracted depends on taking it out of the ordering over general subcategorization. Furthermore, the 'inheritance' of 'gap' information onto the result of some combination proceeds under the general type-combination system provided by the calculus, whereas HPSG requires feature inheritance conventions for this purpose.

3.2 Island phenomena

A number of problems that arise for the treatment of extraction relate to *boundedness* of extraction. The relevant phenomena are often referred to as *island phenomena*, a term introduced by Ross (1967). An *island* is a constituent from which the extraction of subconstituents is impossible or highly constrained (Ross's metaphor being that subconstituents of an island are marooned there). Island constraints have typically been explained by the postulation of *bounding nodes,* i.e. constituent nodes that present barriers to extraction, as, for example, in *Subjacency* theory, which originates with Chomsky (1973) and is still important in current GB work. We shall briefly consider some island phenomena as a basis for further discussion, but note that this list is not intended to be exhaustive. Some further island phenomena will be mentioned later in relation to the account we develop.

3.2.1 Bridge verbs and unbounded extraction

We consider firstly extraction from clauses embedded under verbs. In English it is possible to extract from such clauses, this forming the paradigm case of unbounded dependency under movement, exemplified in the following sentences:

(3.13) a. (John is the man) who Mary loves.

b. (John is the man) who Bill believes that Mary loves.

c. (John is the man) who Tom said that Bill believes that Mary loves.

The possibility of such unbounded extraction is a language particular characteristic of English. There are many languages which exhibit much more restricted possibilities for extraction, for example allowing left extraction to occur unboundedly within a tensed clause but not out of a tensed clause. Such variation within and between languages has been explained in terms of a notion of *bridge verb*, i.e. sentential complement verbs which license extraction from their complement, contrasting with non-bridge verbs, whose sentential complements is an island.

3.2.2 *Wh*-islands

There are well known phenomena which suggest that unbounded extraction in English is itself highly constrained. An important case is constraints on multiple extraction, giving rise to so-called *Wh-island* phenomena, illustrated in (3.14) and (3.15).

(3.14) a. John wondered who bought the book

b. *This is the book that John wondered who bought

c. *Which book did John wonder who bought

(3.15) a. John wondered which book Mary bought

b. *This is the woman who John wondered which book bought

c. *Who did John wonder which book bought

In transformational theory, island phenomena have been explained in terms of theories such as Subjacency, which postulate the existence of designated *bounding* nodes (generally held to be S and NP for English). According to Subjacency, no more than one bounding node may be crossed in any instance of movement. The landing site for Wh-movement is taken to be COMP (or in some accounts SPEC of CP). The effect of Subjacency is to prevent an item moving directly from its extraction site to the COMP of the matrix clause. Instead, unbounded extractions such as that in (3.11) are taken to involve a series of movements, i.e. firstly, to the nearest COMP and then to the next highest COMP, where each of these movements is permitted by Subjacency. Such movement is termed *successive cyclic movement*.

Alternative accounts of boundedness phenomena exist within transformational theory. For example, according to Koster (1978, 1987) movement is subject to a one-node condition, the Bounding Condition, which rules out movements that cross even a single bounding node unless particular conditions are met. According to Koster, $\overline{S}$, rather than S, is the relevant clausal bounding node for English (and other languages). Again, this condition forces long distance extraction to occur by successive cyclic movement.

The transformational theory explanation of *wh*-island phenomena involves an assumption that two *wh*-phrases may not occupy the same COMP node. Since two *wh*-phrases extracted from an embedded clause may therefore not move to the same COMP, one would have to move directly to a higher COMP, but this movement is ruled out by Subjacency. Under the assumptions of this framework, a *trace* is left whenever a constituent moves from a position. The further assumption that traces must be governed allows an explanation of what it is to be a bridge verb. These are taken to be verbs that are capable of governing a trace in the COMP of an embedded clause, government without which an ill-formed structure would result. Thus, the possibility of extraction from an embedded clause is claimed to depend upon the lexical properties of the governing verb.

3.2.3 Extraction of adjuncts

The possibility of long distance extraction appears to be restricted to constituents of certain categories. For example, although adverbial adjuncts may undergo left extraction, extraction of them from an embedded clause is in general ungrammatical:

(3.16) a. John fixed the car so that he could get to work in the morning.

b. Why did John fix the car?

c. *Why do you remember that John fixed the car?
(* under the intended reading)

Engdahl (1985), discussing adverb extraction in Swedish and English, points out that there are limited possibilities for adverb extraction from embedded clauses, but only with a small number of sentence embedding verbs. Most of Engdahl's discussion relates to Swedish, but it is clear that this is also possible for English verbs such as *say* and *think* (which correspond to Swedish verbs that also allow adverb extraction). This is illustrated in (3.17), where the acceptability of (3.17b) as an answer to (3.17a) indicates extraction of the adverb from the embedded clause. This contrasts with (3.18).

(3.17) a. Why do you think that John fixed the car?

b. Because he needs it to get to work.

c. Because I didn't see it outside his house today.

(3.18) a. Why do you remember that John fixed the car?

b. *Because he needs it to get to work.

c. Because he spent a whole night in the pub telling me about it.

(Here, of course, * indicates unacceptable continuation of discourse, rather than an ungrammatical sentence.) Engdahl suggests that the possibility of the extraction in (3.17) is related to the involvement of semantic factors. Thus, pairs of grammatical and ungrammatical sentences can be found which differ only in terms of the aspect and mood of the matrix verb. In later discussion of adjunct extraction, we shall call verbs which allow it *super-bridges*, referring to their exceptional behaviour in acting as bridges to the extraction of adjuncts.

3.2.4 The Complex Noun Phrase Constraint and adjunct islands

It is in general impossible to extract from relative clauses and sentential complements of nouns, as illustrated in (3.19) and (3.20):

(3.19) a. John met a man who loves the works of Shakespeare

b. *What did John meet a man who loves?

(3.20) a. John denied the claim that he stole the book

b. *What did John deny the claim that he stole?

This island behaviour was first noted by Ross (1967) who formulated a condition on movement called the Complex NP Constraint (CNPC) to explain it.

Adjuncts are also islands to extraction, as (3.21) illustrates for VP modifiers. Note however that subcategorized adverbs typically do permit extraction, as in (3.22).

(3.21) a. John filed the articles without telling Bill

b. *Who did John file the articles without telling

(3.22) a. John put the book on the table

b. Which table did John put the book on?

We take relative clauses to be adnominal modifiers. This allows an alternative grouping of phenomena to that suggested by the CNPC, with the behaviour of relative clauses being addressed under a general rubric of the island status of adjuncts.

3.2.5 The Right Roof Constraint

Ross (1967) provides evidence that right extraction is clause bounded. He refers to this as the *Upward Boundedness Constraint* on right extraction, commonly known as the *Right Roof Constraint* (RRC). The following examples illustrate this constraint:

(3.23) a. I had hoped that a serious debate on this topic could develop for quite some time.

b. I had hoped that a serious debate could develop on this topic for quite some time.

c. *I had hoped that a serious debate could develop for quite some time on this topic.

There is some controversy as to the reality of the RRC. For example, Gazdar (1981) draws on work by a number of authors (Grosu, 1972; Witten, 1972; Postal, 1974; Andrews, 1975) in arguing against the the RRC. Gazdar presents examples of RRC violations involving rightward movement of NPs and clauses. We agree that these examples throw the status of the RRC into question for the case of NPs and clauses (although we take the matter to be far from settled). However, Gazdar does not address the case of relative clause extraposition, illustrated in (3.23), which appears to show RRC effects far more robustly. All the cases of right extraction considered in the remainder of this chapter involve relative clause extraposition, and we will assume that in this case at least upward boundedness does apply.

3.3 Previous categorial accounts of island phenomena

We next look at some previous attempts to deal with locality constraints in categorial treatments of extraction, and consider some problems that arise for these accounts.

3.3.1 Restrictional accounts of island phenomena

In the flexible CG treatments of extraction we have discussed, extraction depends on the construction of a single constituent from the material which forms the extraction domain. If this constituent cannot be constructed, extraction cannot occur. This allows for the approach to island constraints that we consider in this section, whereby restrictions are placed on the rules of the grammar to block construction of a constituent for the extraction domain lacking the extracted element in those cases where extraction would involve an island violation. This is in general achieved by blocking what corresponds to an island constituent lacking some extracted subconstituent being incorporated into any larger constituent as a step toward completing an overall extraction analysis. We shall call accounts of island constraints which work in this way *restrictional* accounts.

Bouma (1987) suggests how the island status of embedded questions might be handled in a flexible CG (though he does so only for the purpose of arguing against such an account). This involves using a feature [±wh] to distinguish embedded questions. A verb such as *to wonder* which takes an embedded question complement might have a category vp/s[+wh], and a complementizer of embedded questions such as *whether* might have category s[+wh]/s. Constraining the forward composition rule as in (3.24), to disallow composition into embedded questions, blocks analysis of *wh*-island violations such as that in (3.25) (where the starred reduction indicates the point at which the analysis blocks).

(3.24) $X/Y,\ Y/Z \Rightarrow X/Z \qquad \text{where } Y \neq S[+wh]$

(3.25) *who have I been wondering whether bill likes

	wondering	whether	bill likes
	vp/s[+wh]	s[+wh]/s	s/np
		s[+wh]/np (B)	
	*** (B***)		

Bouma suggests a problem that arises for restrictional accounts of island constraints (Bouma, 1987) which concerns the relationship predicted between the possibilities of Right-Node Raising (RNR) and extraction. In particular, where rule constraints prohibit the construction of some constituent to prevent left extraction from it, RNR from it will also be blocked because precisely the same constituent is required for the RNR analysis.

For example, consider the following example which involves RNR out of a *wh*-island (i.e. the first conjunct):

(3.26) I have been wondering whether, but wouldn't positively want to state that, your theory is correct. (Bresnan, 1974)

Let us assume the restrictional account of *wh*-island behaviour discussed above. It should be clear that the constraint on the composition rule which blocks the *wh*-island violation analysis in (3.25) will also prevent the left hand conjunct of (3.26) being analysed as a constituent, as is required for the overall RNR analysis. Hence, the restrictional account, in attempting to exclude (3.25), incorrectly also excludes the related RNR example in (3.26). This prediction applies more broadly, for example giving similar predictions for the relation of RNR and the CNPC. As Bouma points out, this prediction is in conflict with the claims of a number of authors who contend that RNR out of a complex NP is grammatical (e.g. Wexler and Culicover, 1980; McCawley, 1982; and Levine, 1985).

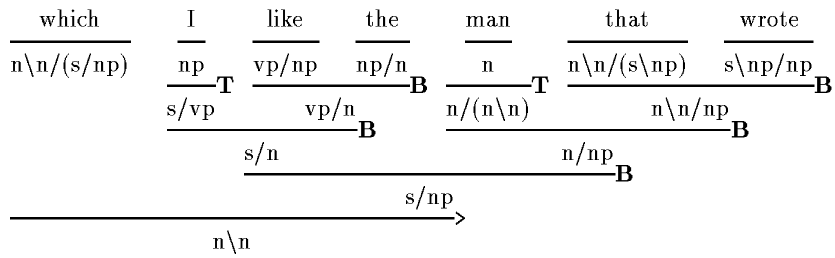
Steedman (1987) suggests an account of the island behaviour of adjuncts, specifically noun modifiers (including relative clauses) and adverbial adjuncts. This is not strictly a restrictional account as we have defined them above, being based on 'pragmatic' factors. However, Steedman suggests a particular point in terms of the grammar at which the pragmatic factors might come into play, determining the appropriateness of the application of syntactic rules. This gives rise to an approach which has something of the character of a restrictional account.

Adjuncts are taken to have endocentric modifier types, such as (n\n) and (vp\vp), which apply to the phrase that they modify. Incomplete adjuncts (including those that lack some extracted subconstituent) are functions into the modifier type, i.e. functions requiring one or more arguments *before* they can apply to the constituent that they modify. If functional application is the only way available for them to combine under the particular grammar, an extraction analysis cannot be completed and so adjuncts behave as islands. The following derivation illustrates this for extraction from a relative clause (the starred reduction indicates the point at which the analysis blocks):

(3.27)	*(a book)	which	I	like	the	man	that	wrote
		n\n/(s/np)	np	vp/np	np/n	n	n\n/(s\np)	s\np/np
							n\n/np (B)	
						(man … that wrote) ***		

Note however that it is perfectly possible to provide combination rules which allow for extraction from the relative in this case. Steedman (1987) discusses the possibility of type raising the modified constituent over the adjunct. For example, forward type raising of the noun over its postmodifier allows the above partial analysis to be completed as follows:

(3.28) *(a book)



Clearly, if such type raising were freely allowed, the approach would fail to characterize the island behaviour of adjunct phrases. Steedman observes, however, that allowing some extent of such type raising might be appropriate given the relative acceptability of certain cases of extraction from adjuncts.[5] Steedman proposes an approach which links the treatment of adjunct island behaviour to certain 'pragmatic' factors. In particular, he argues that the adjunct island constraint reflects a 'cost' associated with the use of the relevant type raising rules, a cost which is semantic in origin, and which reflects the 'reasonableness' of the (incomplete) complex concepts that the ensuing combinations give rise to.

This proposal, however, gives rise to a 'paradox'. The information required to evaluate the correctness of applying the type raising rule is *not available* at the stage of applying the rule. The information only becomes available at the later stage, when the output of the rule is incorporated into larger constituents whose 'reasonableness' can be evaluated. It seems that the relevant type raising rule, far from having only limited applicability, would need to be allowed to apply freely, so that a *post hoc* filtering process (evaluating semantic 'reasonableness') could apply to the constituents into which the raised types were incorporated.

This problem might be to some extent alleviated if further combination rules were available which (potentially) allowed the construction of constituents enabling island-violating extractions, but where the step of 'semantic evaluation' could apply directly to each rule's output (rather than to constituents into which that output is later incorporated). For example, the combination schema:

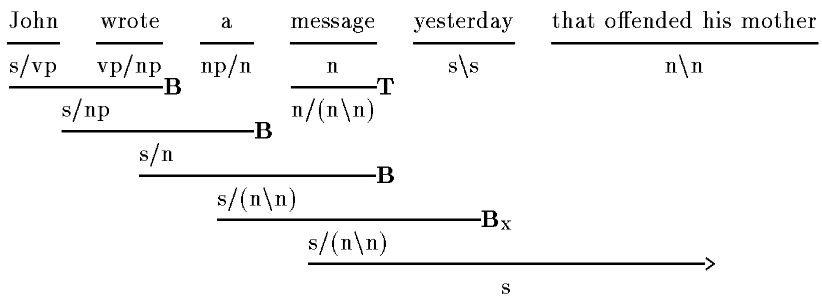
$$\mathrm{Y},\ \mathrm{X\backslash Y/Z} \Rightarrow \mathrm{X/Z}$$

would allow the starred reduction in (3.27), and the semantic evaluation step could apply directly to its output in deciding the correctness of the rule's application. Even this modification of Steedman's proposal seems unsatisfactory in requiring a 'generate and test' pattern of processing, with combinations being made whose result must be discarded.

[5] Consider, for example, the following island violation, due to Belletti, via Chomsky (1982): *... books which I will go to London without reading.*

A further problem for Steedman's account of the island status of relative clauses arises if we wish to use type-raising over relatives in the treatment of relative clause extraposition, as has been suggested by Moortgat (1988).[6] This possibility is illustrated in (3.29), where such type-raising allows extraposition of the relative *that offended his mother* past a sentential adverb:

(3.29)



Some such manoeuvre appears necessary for a CCG-style treatment of relative clause extraposition since, in general, for some constituent to undergo extraction, it must be an argument of some other constituent. The problem that arises is that it then appears that we can no longer restrict the type-raising of a noun over its adjunct to special conditions of semantic 'reasonableness,' since relative clause extraposition is not conditioned by the semantic factors that appear to license the relevant CNPC violations. This is most obviously seen from the fact that there is no correspondence between grammatical sentences which involve right extraction of a relative clause and related grammatical sentences involving CNPC violations:

(3.30) a. John wrote a message yesterday that offended his mother

b. *Who did John write a message yesterday that offended?

c. *Who did John write a message that offended yesterday?

Under Moortgat's proposal, the relative clause extraposition in (3.30a) suggests that *message* is type-raised over its adjunct. But the availability of that raised type does not seem to allow for extraction from the adjunct as (3.30b) and (3.30c) illustrate. A similar problem arises in respect of the island status of adverbs if we wish to use type-raising in handling their left extraction.

3.3.2 Unimodal treatments of island constraints

In this section, we consider accounts of islands based on the □ operator. This approach originates with the proposals of Morrill (1989), although our use of the method differs in

[6] Note, however, that Steedman has not adopted this proposal.

some minor details.[7] The essential ideas of this approach were discussed in the preceding chapter, and we shall not work through the details of this again here. Instead, we are concerned in this section with problems that arise for treatments based on this apparatus. These problems motivate the treatment to be developed subsequently in this chapter, which is based on a *polymodal* system. By contrast to this, we refer to treatments which stem from Morrill's proposals as *unimodal* treatments of linguistic boundaries.

Morrill's proposal

In Morrill (1989), where the use of the $\Box$ operator is originally proposed, the method is motivated in relation to handling locality constraints on reflexivization. Szabolcsi (1987a) suggests an account of reflexivization in which reflexives are given types corresponding to those of noun phrases type-raised over verbs (i.e. types of the form x/(x\np) or x\(x/np) where x is some verbal type). Reflexives are assigned meanings which when applied to the semantics of a verb, causes the outermost argument position of this semantics to become semantically identified with some (functionally) later argument position. Hence, reflexives have meanings such as $\lambda f.\lambda y.[f\,y\,y]$ and $\lambda f.\lambda y.\lambda z.[f\,y\,z\,y]$, etc.

For example, Szabolcsi assigns the reflexive *himself* a type (s\np)\(s\np/np) with meaning $\lambda f.\lambda y.[f\,y\,y]$ to allow for subject/object reflexivization as in *John saw himself.* A derivation for this sentence is shown in (3.31). The meaning assigned is given by the expression in (3.32a), equivalent to that in (3.32b).

(3.31)
$$\dfrac{\dfrac{\text{John}}{\text{np}} \qquad \dfrac{\dfrac{\text{saw}}{\text{s}\backslash\text{np/np}} \quad \dfrac{\text{himself}}{(\text{s}\backslash\text{np})\backslash(\text{s}\backslash\text{np/np})}}{\text{s}\backslash\text{np}}<}{\text{s}}<$$

(3.32) a. $(\lambda f.\lambda y.[f\,y\,y])\ \text{saw}'\ \text{john}'$

b. $\text{saw}'\ \text{john}'\ \text{john}'$

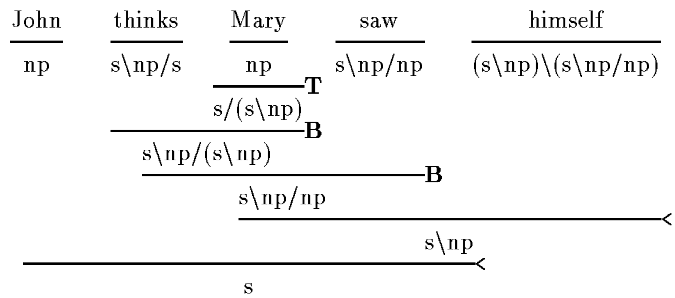
However, as Szabolcsi herself observes, without some further complication this account will allow violations of the locality constraints on reflexivization.[8] For example, the following ungrammatical example, where the reflexive is bound by a NP outside the embedded clause containing the reflexive is not blocked, as the proof in (3.34) shows.

(3.33) *John$_i$ thinks Mary saw himself$_i$

[7] One significant difference is that Morrill takes the $\Box$ operator to be an *intensional* type-constructor, its associated inference rules having semantic operations of intensionalization and extensionalization. In contrast, we take $\Box$ to be of essentially only syntactic import, its associated inference rules not affecting the semantics of the types in the proofs to which they are applied.

[8] Szabolcsi (1987a) hints at the possibility of handling locality effects in terms of a lexicality property, as we shall discuss in Chapter 5.

(3.34)



Morrill shows that this overgeneration can be avoided using the □ apparatus. For example, under the lexical type assignments shown, (3.35) cannot be derived:

(3.35)

John	thinks	Mary	saw	himself		
□np	□(s\np/□s)	□np	□(s\np/np)	□((s\np)\(s\np/np))	⇒	s

To analyse the string *thinks Mary saw* as a single constituent, we must hypothesize an np assumption corresponding to the verb's object. Since this occurs within an embedded domain, it must be □np (i.e. to satisfy the condition on [□I]), giving an overall type (s\np)/□np. However, the reflexive requires a type (s\np)/np for its argument, which cannot be derived when the object belongs in an embedded domain.

Using □ to handle locality effects on reflexivization sets certain requirements on the marking of □ on lexical types in general. For example, embedded clauses must be marked as modal domains to block long binding of reflexives. It follows that, to enable long distance extraction, the types assigned to extracted items must be able to extract from within modal domains. We might then expect a unimodal approach to have problems addressing island constraints since, following the discussion of □ in Chapter 2, it would seem that such extractor types will allow extraction irrespective of the number or origin of any boundaries that intervene, apparently leaving no room for extraction to be conditioned by the linguistic material that intervenes between the extracted element and its extraction site. This would indeed be the case if we were limited to the pattern of □-marking described in Chapter 2.

Morrill (1990a) suggests an account of island constraints which depends upon diverging from this suggested pattern of □-marking. Firstly, note that Morrill assumes that, for verbs taking a complementized sentential complement, the modal boundary is specified by the complementizer, as for example in (3.36) (ignoring permutation operators to simplify presentation):

(3.36)

who	John	knows	that	Mary	likes
□(rel/□(s/□np))	□np	□(s\np/sp)	□(sp/□s)	□np	□(s\np/np)

Morrill suggests that some lexical types might not be marked with an 'all encompassing' box, but rather be of the form $\Box X/\Box Y$ or $\Box X\backslash\Box Y$ so that the type has an implicational connective for its principal connective. In particular, a word such as *whether* might be assigned a type such as $\Box sp/\Box s$, as a basis for handling certain cases of the *wh*-island constraint. This type does not allow extraction from the sentential complement. Thus, the following combination is *not* possible:

(3.37)
$$\frac{\text{*who}}{\Box(rel/\Box(s/\Box np))} \quad \frac{\text{John}}{\Box np} \quad \frac{\text{knows}}{\Box(s\backslash np/sp)} \quad \frac{\text{whether}}{\Box sp/\Box s} \quad \frac{\text{Mary}}{\Box np} \quad \frac{\text{likes}}{\Box(s\backslash np/np)} \quad \Rightarrow \quad rel$$

Note the relative pronoun type that Morrill uses here. This requires that the relative pronoun's complement be analysed of type $\Box(s/\Box np)$, i.e. with an encompassing $\Box$. However, such an analysis is not possible because the complementizer type is, as a whole, non-modal (i.e. having / for its principal connective).

Let us look at this account in slightly more detail. Consider the following possible relative pronoun type (ignoring irrelevant modalities):

$$rel/\Box_i(s/\Box_j np)$$

The two $\Box$'s have been distinguished by subscripts. It should be clear that it is $\Box_j$ which determines whether extraction is bounded or not, i.e. if it is absent only bounded extraction is possible. For Morrill's account of islands, the presence or absence of the other box $\Box_i$ has the effect of determining whether extraction is sensitive to island constraints, such as the island status of embedded questions specified by either verb or complementizer type as just described. This gives rise to the following possibilities:

(3.38) a. $rel/(s/np)$

b. $rel/\Box_i(s/np)$

c. $rel/(s/\Box_j np)$

d. $rel/\Box_i(s/\Box_j np)$

The types in (3.38a) and (3.38b) would only allow bounded extraction. (3.38c) would give 'island insensitive' unbounded extraction, and (3.38d) 'island sensitive' unbounded extraction.

Morrill's proposals introduce an additional degree of freedom which is sufficient to handle the case discussed, but does not generalize to allow for more complex cases. Consider, for example, extraction from subjects. These appear to be islands (giving the Subject Condition on extraction):

(3.39) a. ... the man who Mary bought a picture of.

b. *... the man who a picture of was on sale.

Consider how Morrill's islands approach might be applied to this case. For an intransitive verb we could assign a type such as:

$\Box s \backslash \Box np$

The absence of an all encompassing $\Box$ on this type would block extraction as before. However, a problem arises for verbs of any greater arity. Consider what type we might assign for a transitive. The type must not have an encompassing $\Box$, leaving the following two possibilities:

$\Box s \backslash \Box np / np \qquad \Box s \backslash \Box np / \Box np$

Neither of these types would allow extraction of or from the object NP, and hence are incorrect.

As another case, consider asymmetries in NP and PP extraction. Both kinds of extraction are potentially unbounded, and show similar sensitivity to the island status of embedded questions, and so both would receive types similar to that in (3.38d). However, as originally noted by Belletti (quoted in Chomsky 1982), there is a clear distinction between the acceptability of NP and PP extractions from adjuncts: whilst the former is judged relatively poor, the latter is strongly ungrammatical:

(3.40) a. ? Who did you leave the meeting without speaking to?

b. * To whom did you leave the meeting without speaking?

We might attempt to handle the island status of adverbial adjuncts in line with Morrill's proposal by assigning them a type such as:

$\Box(vp \backslash vp)/\Box vp$

but this approach predicts no distinction for NP vs. PP extraction, and provides no basis for addressing such a distinction.

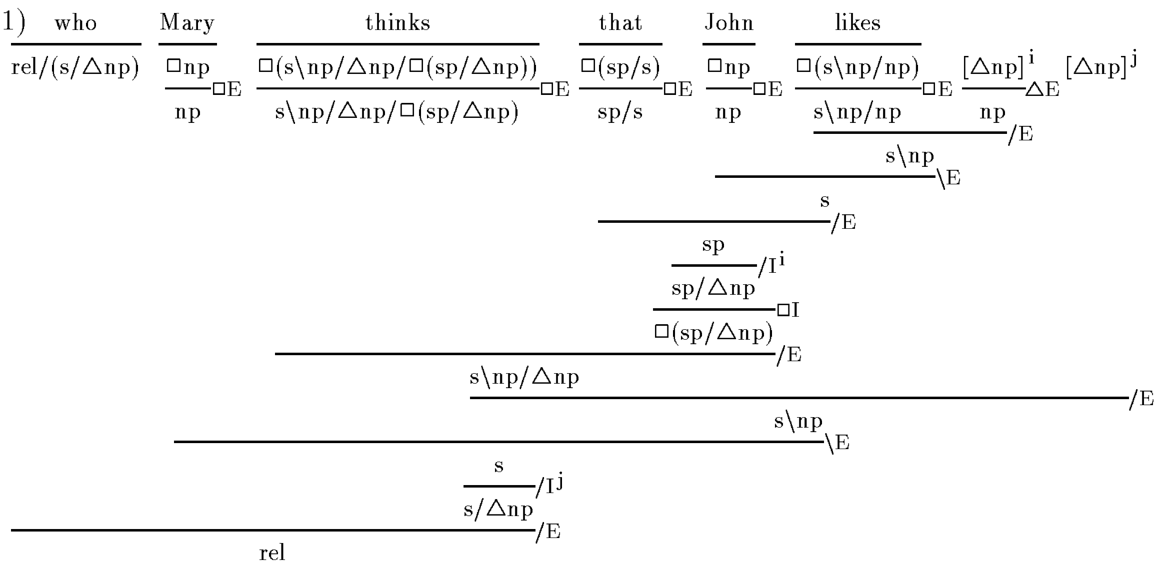
Hepple's proposal

An alternative treatment of island constraints which uses the unimodal apparatus is described in Hepple (1989), where it is suggested that extracted items should be assigned types which do not allow for extraction across modal boundaries (such as, for example, with the relative pronoun type $rel/(s/\triangle np)$, which abstracts over a NP that is not box-modal). Then, special provision must be made to allow for long-distance extraction. This is done by giving heads which specify modal boundaries additional types that specifically license the escape of extracted elements. For example, in addition to a basic type $\Box(s \backslash np/\Box sp)$, the verb *thinks* might have a further lexical type:

$\Box(s\backslash np/\triangle np/\Box(sp/\triangle np))$

which directly licenses the escape of an extracted NP from the modal domain of its sentential complement, as in the following proof:

(3.41)



Such additional types might be generated by a lexical rule such as the following (which is assumed to apply prior to the addition of each lexical type's encompassing □):

(3.42) a. $s\backslash np/\Box sp \Rightarrow s\backslash np/\triangle np/\Box(sp/\triangle np)$

b. $\lambda f \lambda g \lambda x. f(gx)$

(The operation performed by this rule essentially corresponds to *division*, though not quite given the presence of the □ operator.) The possession of such additional 'escape hatch' types constitutes what it is in this account for some word to be a *bridge*. Island constraints would be handled in terms of the assignment of bridge types to words in the lexicon. For example, the island status of embedded *wh*-questions in English might be handled by not assigning bridge forms for verbs which take embedded question complements. Thus, the rule (3.42) might be stated to require that the verb sought a [−wh] complement.

There are a number of reasons, however, why such an approach seems unsatisfactory. The approach seems somewhat cumbersome in requiring us to explicitly encode in the lexicon pretty much all possibilities for boundary crossing extraction. For example, since both NPs and PPs can undergo long-distance extraction in English, additional lexical types for bridge verbs would be needed to act as escape hatches for both of these cases. For languages which allow multiple extraction (for example, Swedish), multiply divided bridge forms would be needed to allow for different numbers and types of extracted elements. As another case, consider the Subject Condition, which will require us to specify subjects as bounded domains, so that a transitive verb might have a type such as:

$\Box(s\backslash\Box np/np)$

However, right extraposition from subjects of nominal adjuncts such as PPs and relative clauses is possible as in: *A man arrived who was wearing a false beard.* Such cases would require the provision of 'escape hatch' types for all verbs to license right extraposition of nominal adjuncts, such as the following transitive verb type:

$\Box(s/\triangle(n\backslash n)\backslash\Box(np/\triangle(n\backslash n))/np)$

This would give an enormous expansion of the lexicon. Thus, this approach to island constraints seems unduly cumbersome, and seems also to lack explanatory potential since it essentially depends upon case by case lexical encoding of the observed behaviour.

3.4 A polymodal system for linguistic boundaries

In this section, we argue that we can give a better account of island constraints, as well as other phenomena involving linguistic boundaries, if we move from a system in which there is only a single unitary notion of 'linguistic boundary' to one in which there are many related notions of boundary, and in which the sensitivity of any process to a particular boundary can be directly addressed. We show that such a system can be constructed in terms of a polymodal logic of necessity in which there are many *interrelated* necessity operators. To introduce the system we require, it is first necessary to discuss the *interpretation* of types within the Lambek calculus.

3.4.1 Lambek interpretation and polymodality

The standard approach to interpretation for the Lambek calculus is in terms of *sets of strings*. These sets are subsets of an overall 'language' of strings $\mathcal{L}$, which is obtained by closing the vocabulary (formally taken to be a set of arbitrary atoms or 'words') under concatenation. Atomic types are assigned some subset of $\mathcal{L}$ for their interpretation. The interpretation of complex types is fixed in terms of definitions for the type-forming operators. For example, the interpretation of implicational types is specified by the following definitions for the type-forming connectives / and \ (where the notation $[\![X]\!]$ stands for the interpretation of the type X):

(3.43) $$[\![A/B]\!] = \{x \in \mathcal{L} \mid \forall y(y \in [\![B]\!] \rightarrow xy \in [\![A]\!])\}$$
$$[\![A\backslash B]\!] = \{x \in \mathcal{L} \mid \forall y(y \in [\![B]\!] \rightarrow yx \in [\![A]\!])\}$$

Thus, the interpretation of A/B (i.e. $[\![A/B]\!]$) corresponds to that set of strings x such that for any string y belonging to the type B, the concatenation xy is a string belonging to the type A.

This approach to interpretation gives rise to notion of *valid* type combination. Thus, any type combination

$X_1, \ldots, X_n \Rightarrow X_0$

is *valid* just in case every string $x_1 \ldots x_n$ that can be formed by concatenating members of the given antecedent types in the given order is a string of the type X_0, i.e.

(3.44) $X_1, \ldots, X_n \Rightarrow X_0$ is valid *iff*
$\forall x_1, \ldots, x_n ((x_1 \in [\![X_1]\!], \ldots, x_n \in [\![X_n]\!]) \rightarrow x_1 \ldots x_n \in [\![X_0]\!])$

This has the special case where there is a single antecedent type, when the type transition is valid just in case the interpretation of the single antecedent type is a subset of the interpretation of the succedent type:

(3.45) $X \Rightarrow Y$ *iff* $[\![X]\!] \subseteq [\![Y]\!]$

Following standard ideas about the interpretation of S4 modal logics, we can give an interpretation for the $\Box$ operator as follows:

(3.46) $[\![\Box X]\!] = [\![X]\!] \cap \mathcal{L}'$
where $\mathcal{L}'$ is some sublanguage (i.e. $\mathcal{L}' \subseteq \mathcal{L}$) that is closed under concatenation

This interpretation gives the appropriate S4-like behaviour. Thus, the following type-transitions are valid as we will see:

(3.47) a. $\Box X \Rightarrow X$

b. $\Box X \Rightarrow \Box\Box X$

c. $\Box(X/Y) \Rightarrow \Box X/\Box Y$

Consider the introduction and elimination rules for $\Box$, repeated here, in relation to this interpretation.

(3.48) $\dfrac{\begin{matrix}\vdots \\ \Box X\end{matrix}}{X}\Box E \qquad \dfrac{\begin{matrix}\vdots \\ X\end{matrix}}{\Box X}\Box I$ where each path to an undischarged assumption includes an independent subproof of a $\Box$-type

The elimination rule is fine since obviously:

$[\![X]\!] \cap \mathcal{L}' \subseteq [\![X]\!]$

For the introduction rule, let us begin by considering the special case where each undischarged assumption is a $\Box$ type. Let S be the set of all strings that result from concatenating strings in the antecedent interpretations in the given order. The proof to which the inference rule is to be applied indicates that $S \subseteq [\![X]\!]$. Since each undischarged assumption is assumed to be a $\Box$ type, its interpretation must be a subset of $\mathcal{L}'$, and since $\mathcal{L}'$ is closed under concatenation it follows that $S \subseteq \mathcal{L}'$. Hence, $S \subseteq [\![X]\!] \cap \mathcal{L}'$, showing that the proof that results from the application of the inference rule is valid. This result generalizes to the standard statement of the introduction rule, where the condition on the rule is for independent subproofs of modal types rather than the stronger requirement of modal hypotheses, since (as we noted in Chapter 2) the one form of the rule simply corresponds to the other form generalized with respect to the transitivity of the derivability relation (expressed by the cut rule). The type transitions shown in (3.47) are clearly valid since they are theorems of any system which has the rules in (3.48).

In the polymodal system we require, there are multiple notions of necessity, some of which can be derived from others.[9] For this system, we allow an indefinite number of necessity operators to be defined, which differ in terms of the sublanguages used in their definition. These operators are notated using labelled boxes (i.e. $\boxed{a}$, $\boxed{b}$, ..., etc), where the label names the language used in defining the modality. The interpretation of these operators is defined as follows:

(3.49) $[\![\boxed{\alpha}X]\!] = [\![X]\!] \cap \mathcal{L}_\alpha$
where $\mathcal{L}_\alpha$ is some sublanguage (i.e. $\mathcal{L}_\alpha \subseteq \mathcal{L}$) that is closed under concatenation

We call the language $\mathcal{L}_\alpha$ used in defining a modal operator its *defining language*.

The derivability of one modality from another becomes possible if we place constraints on admissible interpretive models. For example, if we set the constraint that $\mathcal{L}_a \subseteq \mathcal{L}_b$, it follows that the type transition $\boxed{a}X \Rightarrow \boxed{b}X$ is valid. We know that $\boxed{a}X \Rightarrow \boxed{b}X$ is valid just in case:

$$[\![\boxed{a}X]\!] \subseteq [\![\boxed{b}X]\!]$$

which, from the definition of the boxes, is true if:

$$\mathcal{L}_a \cap [\![X]\!] \subseteq \mathcal{L}_b \cap [\![X]\!]$$

This is indeed the case given that $\mathcal{L}_a \subseteq \mathcal{L}_b$.

To be able to address the possibilities for valid type combinations and transitions in the proof system, the inference rules (and in particular the [□I] rule) must have access to the constraints on models that have been stated for the defining sublanguages. Assuming this (which is standard), we can state the inference rules for the polymodal system as follows:

(3.50)
$$\dfrac{\begin{array}{c}\vdots\\ \boxed{\alpha}A{:}x\end{array}}{A{:}x}\,\Box E \qquad \dfrac{\begin{array}{c}\vdots\\ A{:}x\end{array}}{\boxed{\beta}A{:}x}\,\Box I$$
where each path to an undischarged assumption includes an independent subproof of some modal type $\boxed{\alpha}X$ such that $\mathcal{L}_\alpha \subseteq \mathcal{L}_\beta$

[9] In the way that the standard Lambek calculus parallels propositional logic (or even more appropriately, a subset of linear logic), the Lambek logic that results from this line of development parallels systems such as dynamic predicate logic.

The elimination rule is as before except that it can remove any kind of box. The introduction rule again requires that each path to an undischarged assumption leads to an independent subproof of a modal type, but additionally requires that the defining language is a subset of that for the box introduced. The arguments above for the validity of the unimodal □ inference rules are easily adapted for the polymodal rules. The sequent rules for the polymodal apparatus are as follows:

(3.51)
$$\frac{\Gamma,\ \mathrm{x},\ \Lambda \Rightarrow \mathrm{y}}{\Gamma,\ \boxed{\alpha}\mathrm{x},\ \Lambda \Rightarrow \mathrm{y}}[\Box L] \qquad \frac{\boxed{\alpha_1}\mathrm{x}_1,\ \boxed{\alpha_2}\mathrm{x}_2, \ldots,\ \boxed{\alpha_n}\mathrm{x}_n \Rightarrow \mathrm{x}_0}{\boxed{\alpha_1}\mathrm{x}_1,\ \boxed{\alpha_2}\mathrm{x}_2, \ldots,\ \boxed{\alpha_n}\mathrm{x}_n \Rightarrow \boxed{\alpha_0}\mathrm{x}_0}[\Box R] \qquad \text{where for } 1 \leq i \leq n,\ \mathcal{L}_{\alpha_i} \subseteq \mathcal{L}_{\alpha_0}$$

Some relations amongst modalities may arise not as a direct consequence of explicitly stated constraints on models, but because of relations amongst defining languages that follow from the standard properties of sets. The most obvious (and, for us, significant) case is set inclusion relations that follow from the explicitly stated relations under transitivity. For example, if we state $\mathcal{L}_{\mathrm{a}} \subseteq \mathcal{L}_{\mathrm{b}}$ and $\mathcal{L}_{\mathrm{b}} \subseteq \mathcal{L}_{\mathrm{c}}$ as constraints on admissible models, it obviously follows that $\mathcal{L}_{\mathrm{a}} \subseteq \mathcal{L}_{\mathrm{c}}$, with the consequence that $\boxed{\mathrm{a}}\mathrm{X} \Rightarrow \boxed{\mathrm{c}}\mathrm{X}$ is a valid type transition. The transitivity of the subset relation as it bears on the interrelation of defining languages has significant ramifications for the account of island constraints to be proposed in this chapter.[10]

3.4.2 Polymodality and islands: the basic idea

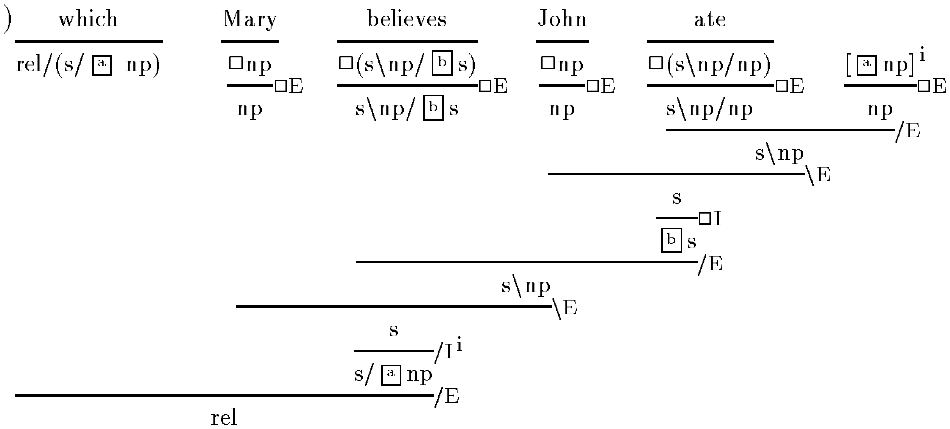
We shall next consider how the polymodal system might be used in some linguistic treatments. For convenience, we assume a sublanguage $\mathcal{L}_{\emptyset}$ for which the modality it defines is notated by the empty box □. We use this for the box marked on each lexical item having scope over all of it. Since we want lexical types to be able to appear embedded under any modal domain, this defining language is taken to be a subset of all the other defining languages we use.[11] The basic idea for the use of this system in handling island constraints is very simple, and is based around the interrelation of the modalities used in specifying modal boundaries and the types for extracted items. This is illustrated by the following example

[10] Further possibilities for relations amongst modalities arise if we allow complex 'names' for the defining languages of modalities, formed using set-theoretic operations. For example, we might allow modalities such as $\boxed{\mathrm{a}\cap\mathrm{b}}$, whose defining language is the intersection of two other sublanguages, i.e. $\mathcal{L}_{\mathrm{a}} \cap \mathcal{L}_{\mathrm{b}}$. Since it is necessarily the case that $(\mathcal{L}_{\mathrm{a}} \cap \mathcal{L}_{\mathrm{b}}) \subseteq \mathcal{L}_{\mathrm{a}}$ it follows automatically that $\boxed{\mathrm{a}\cap\mathrm{b}}\mathrm{X} \Rightarrow \boxed{\mathrm{a}}\mathrm{X}$. The case of set union is less straightforward since the union of two languages that are closed under concatenation will not in general also be a language that is closed under concatenation. However, if we assume an operation ∂ which returns the closure of a set of strings under concatenation, we can see that since necessarily $\mathcal{L}_{\mathrm{a}} \subseteq \partial(\mathcal{L}_{\mathrm{a}} \cup \mathcal{L}_{\mathrm{b}})$, it follows automatically that $\boxed{\mathrm{a}}\mathrm{X} \Rightarrow \boxed{\partial(\mathrm{a}\cup\mathrm{b})}\mathrm{X}$.

[11] Of course the possibility exists that we could use the box marked on a lexical item, which serves the purpose of licensing its appearance within modal domains, to limit its distribution. We shall not, however, be exploiting this possibility in any of the accounts to be offered here.

involving relativization from an embedded clause (which omits the △ operator to simplify):

(3.52)



Note firstly that, in line with the remarks just made, we assume $\mathcal{L}_\emptyset \subseteq \mathcal{L}_{\mathrm{a}}$ and $\mathcal{L}_\emptyset \subseteq \mathcal{L}_{\mathrm{b}}$. The proof has been drawn as running through to completion. However, whether this proof is in fact correct depends on whether the [□I] rule has been correctly applied, which is so just in case $\mathcal{L}_{\mathrm{a}} \subseteq \mathcal{L}_{\mathrm{b}}$. This is the basic idea for the treatment of island constraints: the possibilities for extraction from embedded domains are determined by the relation between modalities specified as constraints on admissible models. As we shall see, this approach allows us to state our observations about a given language's possibilities for extraction in simple and direct terms in the grammar.

3.4.3 Asymmetries in the extraction of different categories

Note that in (3.52), the modality ⊡b marked on the argument of the verb has the effect of specifying that this complement is a modal domain, thereby creating a boundary to any extraction that is not appropriately licensed. We refer to such modalities as *boundary* modalities, for obvious reasons. Consider the modality ⊡a marked on the relative pronoun type in (3.52). This specifies the modal properties of the extracted element, over which the relative pronoun type abstracts. We shall refer to such modalities marked on extractor types as *penetrative* modalities, since they determine an extractor type's capacity to 'penetrate' modal boundaries specified by verbs and other heads.[12] This approach provides a basis for addressing asymmetries observed in the extraction possibilities of different categories, since we can assign distinct penetrative modalities for the different extractor types.

If we assume that the types for extracted NPs and adverbs specify distinct penetrative modalities, this provides a basis for addressing the differences they show for extraction from

[12] Note that this statement that penetrative modalities determine extraction possibilities is not strictly correct, since they are in fact determined by the relations *between* defining languages. However, it is convenient to speak in this 'one-sided' manner when we address extraction from the 'point of view' of extractor types.

different domains. For example, we noted above that whereas extraction of adverbials is in general bounded, NPs show unbounded extraction, although both of these are sensitive to the island status of adjuncts. Let us assume that the penetrative modalities for extracted NPs[13] and adverbs are $\boxed{\text{a}}$ and $\boxed{\text{b}}$, respectively, and that the modal boundaries of adjuncts[14] and (non-*wh*) sentence embedding verbs are specified with the modalities $\boxed{\text{e}}$ and $\boxed{\text{f}}$. (Thus, an adverbial head such as *without* might have a type $\Box(\text{vp}\backslash\text{vp}/\boxed{\text{e}}\text{vp})$.) Then, the extraction behaviour we have described would result given the following constraint, since it licenses the sole possibility of extracting NPs from embedded clauses.

(3.53) $\mathcal{L}_{\text{a}} \subseteq \mathcal{L}_{\text{f}}$

For this account, it is the constraint in (3.53) which give rise to behaviour of (non-*wh*) sentence embedding verbs as *bridges* to extraction.

We noted earlier that, in contrast to the general case, a restricted subclass verbs of English (and Swedish), such as *say* and *think,* do allow extraction of an adverbial phrase from their sentential complement, as observed by Engdahl (1985). We refer to such verbs as *super-bridges*, given their exceptional behaviour in allowing such adjunct extraction. Engdahl suggests that the possibilities for extraction in these cases are related to the involvement of semantic factors. Thus, pairs of grammatical and ungrammatical sentences can be found which differ only in terms of the aspect and mood of the matrix verb. Although allowing that factors of aspect and mood can condition how much alternative interpretations are favoured or disfavoured, we here explore the hypothesis that the limitation of adverbial 'long extraction' to just a few sentence embedding verbs is a fact that should be encoded syntactically.

Assuming that the boundary of super-bridges is specified using a further modality, say $\boxed{\text{g}}$, we might encode the relevant possibilities for extraction using the following constraints:

(3.54) a. $\left.\begin{array}{l}\mathcal{L}_{\text{a}} \subseteq \mathcal{L}_{\text{f}}\\ \mathcal{L}_{\text{a}} \subseteq \mathcal{L}_{\text{g}}\end{array}\right\}$ NPs can extract from the complements of bridge verbs and super-bridges.

b. $\left.\mathcal{L}_{\text{b}} \subseteq \mathcal{L}_{\text{g}}\right\}$ Adverbs can extract from the complements of super-bridges.

These statements encode the relevant possibilities in direct pairwise fashion, i.e. each condition explicitly states that extraction is possible for a particular extracted category across a particular type of boundary. Each statement has no other ramification.

[13] Of course, it is not necessary that we assume a consistent penetrative modality across all NP extractor types. If, for example, a language exhibited clear asymmetries in NP extraction between relativization and *wh*-question formation, then different penetrative modalities could be assigned to address these differences.

[14] We assume for present purposes that different kinds of adjunct are similar in these terms.

3.4.4 Interrelation of boundary and penetrative modalities

It might be seen as an unsatisfactory aspect of an account of island constraints based on the statements in (3.54) that it treats bridges and super-bridges as wholly distinct, with it being necessary to state for NPs that they can extract from the complements of both bridges and super-bridges. We might, for example, take the view that extraction from bridges and super-bridges is fundamentally linked so that it could never be the case that something could extract from the complement of a bridge verb but be unable to extract from the complement of a super-bridge. Since, the notion of super-bridge is essentially a refinement of the notion of bridge, such an interdependence seems appropriate. There is nothing in the statements in (3.54) which encodes this idea.

The possibility arises that we might give more structure to the relations amongst modalities. For example, the interdependence of bridges and super-bridges could be encoded using the statement:

$$\mathcal{L}_{\mathrm{f}} \subseteq \mathcal{L}_{\mathrm{g}}$$

Then, any penetrative modality allowing extraction across the boundary of a bridge verb would also be able to extract across the boundary of a super-bridge. This follows given the transitivity of the subset relation. Thus, the extraction of, say, NPs across bridge verbs is licensed by the constraint $\mathcal{L}_{\mathrm{a}} \subseteq \mathcal{L}_{\mathrm{f}}$. Given this and $\mathcal{L}_{\mathrm{f}} \subseteq \mathcal{L}_{\mathrm{g}}$, it follows that $\mathcal{L}_{\mathrm{a}} \subseteq \mathcal{L}_{\mathrm{g}}$, and hence that NPs can extract across super-bridges. This removes the need to explicitly license extraction across super-bridges for anything that can extract across bridges, to that the set of conditions in (3.54) could be amended to be:

(3.55) a. $\mathcal{L}_{\mathrm{a}} \subseteq \mathcal{L}_{\mathrm{f}}$

b. $\mathcal{L}_{\mathrm{b}} \subseteq \mathcal{L}_{\mathrm{g}}$

c. $\mathcal{L}_{\mathrm{f}} \subseteq \mathcal{L}_{\mathrm{g}}$

As further illustration, let us assume for the sake of discussion that we wanted to add the claim that there is a dependence between possibilities for extraction from adjuncts and extraction from bridge verbs, such that the former cannot be possible when the latter is not. This would place the boundary modalities of adjuncts, bridges and super-bridges into an ordering based on the following relations of their defining languages:

(3.56) $\mathcal{L}_{\mathrm{e}} \subseteq \mathcal{L}_{\mathrm{f}} \subseteq \mathcal{L}_{\mathrm{g}}$

Then, the possibilities for extracting some item would follow from the point at which its penetrative modality linked into this ordering (i.e. by a subset constraint relating the defining language for this penetrative modality to a defining language in the ordering).

A similar linking of extraction possibilities could be made for penetrative modalities as for boundary modalities. Assume, again for the sake of discussion, that we wanted to claim a link between the extraction possibilities of NPs and adverbs, such that whenever adverbs could extract across some boundary, then so too could NPs. Assuming, as before, penetrative modalities $\boxed{\scriptstyle a}$ and $\boxed{\scriptstyle b}$ for NPs and adverbs, this claim could be encoded with the constraint:

$$\mathcal{L}_{\mathrm{a}} \subseteq \mathcal{L}_{\mathrm{b}}$$

The linking of boundary and penetrative modalities in this way provides a means for stating directly within the grammar claims about interdependencies amongst types of boundaries and types of extractors.

3.4.5 The problem of partial acceptability

One problem that arises in characterizing island constraints (and other linguistic phenomena) is partial acceptability, i.e. where speakers judge that certain examples, although not 'good', are also not so bad as to be clearly ungrammatical. Partial acceptability presents a problem for a calculus based approach such as the present one (and indeed for most grammatical frameworks) since given a particular set of lexical assignments and a particular statement of the calculus, any given sentence is either designated 'in' or 'out', since any particular combination of types either is or is not a theorem.

One position we might take on this problem is to simply ignore it, under the justification that a certain extent of idealization of the data is of necessity required in formal linguistics. We might set some (hard to state) level of relative acceptability as a point of demarcation between cases to be counted as 'good' and 'bad' (where 'good/bad' is two-valued formal notion). Here, however, we shall consider how partial acceptability might addressed rather than ignored.

A number of possibilities present themselves for addressing partial acceptability, though these are essentially laid on top of the logical framework, rather than fitting comfortably with it. We might view it as an *uncertainty* on the part of the speaker concerning some aspect of the grammar, perhaps uncertainty as to which of two possible lexical type assignments is appropriate for a given word. In respect of island constraints, uncertainty might concern which penetrative or boundary modality should be marked on a lexical type.[15]

Let us consider this idea in relation to Belletti's observation (noted earlier) that there is a difference in the relative acceptability of NP and PP extractions from adjuncts, such

[15] An alternative possibility is that speaker uncertainty might concern whether or not some constraint on admissible models is part of the grammar.

that whereas the former is in general judged relatively poor, the latter is in general judged strongly ungrammatical. This is illustrated by the following examples (repeated from above):

(3.57) a. ?Who did you leave the meeting without speaking to?

b. *To whom did you leave the meeting without speaking?

Assume as before that adjuncts have boundary modality $\boxed{\scriptstyle e}$, and the two modalities $\boxed{\scriptstyle a}$ and $\boxed{\scriptstyle c}$ such that $\mathcal{L}_a \subseteq \mathcal{L}_e$ holds, but $\mathcal{L}_c \subseteq \mathcal{L}_e$ does not. Assigning $\boxed{\scriptstyle c}$ as the penetrative modality for PPs will mean that cases such as (3.57b) are not allowed. For NPs, however, we might assume speaker uncertainty between having penetrative modality $\boxed{\scriptstyle a}$ and $\boxed{\scriptstyle c}$. Since with one of these modalities the extraction would be allowed, and with the other it would not, we might then expect a 'net effect' of semi-acceptability. Note the characteristic of this view of partial acceptability that it requires that type assignments must be possible for both the cases of admitting and excluding the relevant class of examples.

3.4.6 The island status of embedded questions

We shall next consider some further island phenomena. A simple treatment of the island status of embedded questions could be given by assuming that the distinct types which are required for verbs which take [+wh] and [−wh] complements mark their complement with different boundary modalities, that do and do not allow extraction from their complement, respectively. However, there is considerable variation in the unacceptability of different cases of extraction from embedded questions. For example, extraction from infinitival VP *wh*-complements is generally found better than extraction from tensed S *wh*-complements. This is illustrated by the examples in (3.58), from Chomsky (1986). Many speakers find the sentences in (3.58a,b) less acceptable that those in (3.58c,d).[16]

(3.58) a. *What did you wonder to whom John gave?

b. *To whom did you wonder what John gave?

c. *?What did you wonder to whom to give?

d. *?To whom did you wonder what to give?

Extraction from an embedded question formed using the [+wh] complementizer *whether* is typically found better than that from embedded questions formed using *wh*-movement:

(3.59) a. *? Which book did you wonder who had read

b. ? Which book did you wonder whether Mary had read?

[16] The acceptability judgments marked in (3.58) are the author's.

One approach to handling these distinctions depends on the fact that in the different cases, the [+wh] complements of *wonder* are distinct in type. For the cases in (3.59), the two verbs seek complements s[+wh,−comp] and s[+wh,+comp], respectively (with [±comp] a feature distinguishing complementized and uncomplementized sentences). In (3.58c,d), the complement is a [+wh] VP (i.e. s[+wh]\np). Different types would be assigned to handle these distinct complementation possibilities, and these distinct types may specify distinct boundary modalities for their complement. Then, as discussed earlier, speaker uncertainty as to the boundary modalities marked on the various verb types and/or the penetrative modality of the extracted item would give rise to the pattern of acceptability judgments observed.

An alternative possibility for the distinction in (3.59) is that the lower acceptability of (3.59a) results from some boundary modality introduced by the type of the (embedded) *wh*-item *who*, that is not similarly introduced by the complementizer *whether.* Thus, we might have an extractor type for *who* such as:

$$\Box(\mathrm{s}_{wh}/\boxed{\mathrm{v}}(\mathrm{s}/\triangle\boxed{\mathrm{a}}\mathrm{np}))$$

whose additional boundary modality would affect extraction from the (residue of) the embedded question.

3.4.7 The CNPC and adjunct islands

We assume that the phenomena grouped under the CNPC should in fact be subdivided, with the island behaviour of sentential complements of nouns being dealt with separately from the island behaviour of relative clauses. In the discussion above, it was assumed that the island behaviour of different classes of adjuncts was to be seen as a single problem. This is by no means necessary, and we might want to subdivide the different kinds of modifier to address any variations observed. However, we here follow the earlier simplifying assumption. Some example lexical type assignments to some heads of adjunct phrases are as follows:

(3.60)	(Adverbial)	*without*	$\Box(\mathrm{s}\backslash\mathrm{np}\backslash(\mathrm{s}\backslash\mathrm{np})/\boxed{\mathrm{e}}(\mathrm{s}\backslash\mathrm{np}))$
	(PP adnominal)	*by*	$\Box(\mathrm{n}\backslash\mathrm{n}/\boxed{\mathrm{e}}\mathrm{np})$
	(Relative)	*which*	$\Box(\mathrm{n}\backslash\mathrm{n}/\boxed{\mathrm{e}}(\mathrm{s}/\triangle\boxed{\mathrm{a}}\mathrm{np}))$

Concerning sentential complements of nouns, we assume that the relevant nouns have types which subcategorize for a sentential argument. For example, *belief* might be:

$$\Box(\mathrm{n}/\boxed{\mathrm{h}}\mathrm{sp})$$

where the boundary modality $\boxed{\mathrm{h}}$ would not be so related to any penetrative modality as to allow extraction from the sentential complement. Obviously, the boundary modality for *belief* must be distinct from that of the related verb *believe* given the clear differences the two items show for extraction from their complement.

3.4.8 The Subject Condition

A slightly more interesting case is the Subject Condition, whose effects require us to mark subject arguments with a boundary modality, as in the transitive verb type:

$$\Box(s\backslash\boxed{k}\,np/np)$$

Right extraction of noun modifiers such as restrictive relatives and PPs from subjects is possible, although it is blocked by ordinary sentence boundaries. We can allow right extraposition of relatives to clause-final position by giving relative pronouns a type such as:

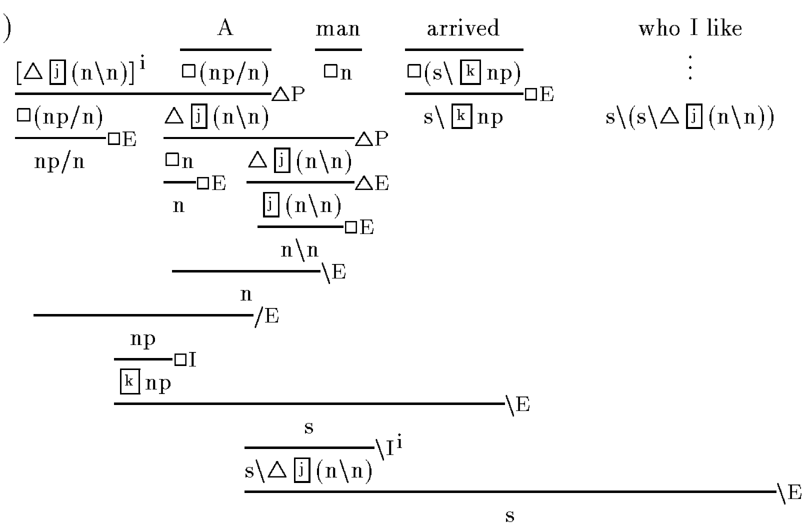
$$\Box(s\backslash(s\backslash\triangle\boxed{j}\,(n\backslash n))/\boxed{e}\,(s/\triangle\boxed{a}\,np))$$

provided we include the constraint $\mathcal{L}_j \subseteq \mathcal{L}_k$. This relative pronoun type gives rise to relative clauses of type:

$$s\backslash(s\backslash\triangle\boxed{j}\,(n\backslash n))$$

which is a raised extractor type for adnominals that appear in clause-final position. This relative pronoun type can be generated from the standard relative pronoun type by lexical rule, and allows the following derivation for *A man arrived who I like* (ignoring the subproof for the relative clause):

(3.61)



In the absence of any further constraints for the penetrative modality $\mathcal{L}_j$, right extraposition of relatives will remain bounded, other than the stated exception of 'subject islands'.

3.4.9 Subject extraction and the **that-t* filter

The move of marking subjects with a modality to make them islands to left extraction has a further significant consequence. This is that the boundary modality not only blocks extraction *from* the subject, it also blocks extraction *of* the subject. A left extractor such as a relative pronoun requires the sentence from which the NP is missing to be analysed as type $s/\triangle\boxed{a}np$. This cannot be derived from the VP type $s\backslash\boxed{k}np$. This would require the validity of the transition $\boxed{a}np \Rightarrow \boxed{k}np$, which is not allowed since there is no constraint $\mathcal{L}_a \subseteq \mathcal{L}_k$. A significant factor here is that the type (i.e. np) for which we want to block extraction from the subject is the same as the type of the subject. If we had unrelated penetrative modalities for, say, NPs and PPs, it would be quite possible to block extraction of NPs from some PP complement of a head without blocking extraction of that PP itself. Thus there is a sort of 'A-over-A' property to the phenomenon in question, i.e. the behaviour only arises because we want to block extraction of NPs *from* NPs.

An obvious side effect of disallowing subject extraction is that it immediately excludes ungrammatical examples involving **that-t* filter violations, such as the following:

(3.62) a. *Who does John believe that loves Mary?

b. *This is the man who I think that arrived yesterday

The problem then arises for how we readmit the grammatical cases of subject extraction. Before we consider the answer to this, a brief comment about exceptional case-marking (ECM) constructions, illustrated in (3.63), is in order.

(3.63) Mary believes John to be an idiot

We assume for ECM constructions, that the 'embedded subject' NP is in fact an argument of the matrix verb. For example, the type of *believes* for the example in (3.63) is:

$$\Box(s\backslash\boxed{k}np/\boxed{g}(s\backslash\boxed{k}np)/np)$$

i.e. a type which looks first for an object NP and then for a VP. The NP object corresponds semantically to the subject of the VP (a fact which follows from the lexical semantics of the matrix verb, i.e. as a consequence of the lexical entailments of this semantics.) This view (i.e. that ECM 'subjects' should be treated as objects of the matrix verb) is also taken, in some form or other, by various non-transformational approaches, for example GPSG, HPSG, and some CGs.

For subject extraction with ordinary cases of sentential complementation we distinguish two cases of interest. Firstly, where extraction is 'string vacuous', i.e. where the *wh*-item appears essentially in the position where the subject would have appeared, as in (3.64), and secondly, where the subject is extracted out of an embedded clause.

(3.64) Who loves Bill?

(3.65) a. Who does John think will win?

b. Who does John think really wants to win?

The second case only arises in the absence of a complementizer (an observation encoded in the **that-t* filter). The absence of a complementizer means that what remains of the embedded clause given extraction of the subject is indistinguishable from a VP. The same is true for the 'string vacuous' subject extraction cases just mentioned.

The solution we adopt for subject extraction is to allow the clause missing a subject to be directly subcategorized for as a VP by the governing item. For the case of 'string vacuous' movement this means that extracted *wh*-items must have types that require a VP argument. For example, the *wh*-word *who* must have a type such as:

$$\Box(\mathrm{n}\backslash\mathrm{n}/\boxed{\mathrm{e}}(\mathrm{s}\backslash\boxed{\mathrm{k}}\mathrm{np}))$$

to allow for relativization examples such as *This is the man who sang*, and a type such as:

$$\Box(\mathrm{s}[+\mathrm{wh}]/(\mathrm{s}\backslash\boxed{\mathrm{k}}\mathrm{np}))$$

to allow for constituent question examples such as *I wonder who sang.* For subject extraction from embedded clauses, we allow additional types for sentence embedding verbs which require a VP argument in place of their sentential argument, and which specify an additional argument, already marked as extracted (with the $\triangle$ operator), which corresponds semantically to the missing subject. For example, *believes* has an additional type:

$$\Box(\mathrm{s}\backslash\boxed{\mathrm{k}}\mathrm{np}/\triangle\mathrm{np}/\boxed{\mathrm{g}}(\mathrm{s}\backslash\boxed{\mathrm{k}}\mathrm{np}))$$

allowing for examples such as *This is the man who I believe will win.* Such additional types can be derived by lexical rule from the types for the same verbs which allow for the corresponding non-extraction examples, i.e. the above type for *believes* would be derived from the type $\Box(\mathrm{s}\backslash\boxed{\mathrm{k}}\mathrm{np}/\boxed{\mathrm{g}}\mathrm{s})$ already assigned to this verb, by a rule such as the following (which will only apply to verbs which specify their complement to be finite and [−wh]):

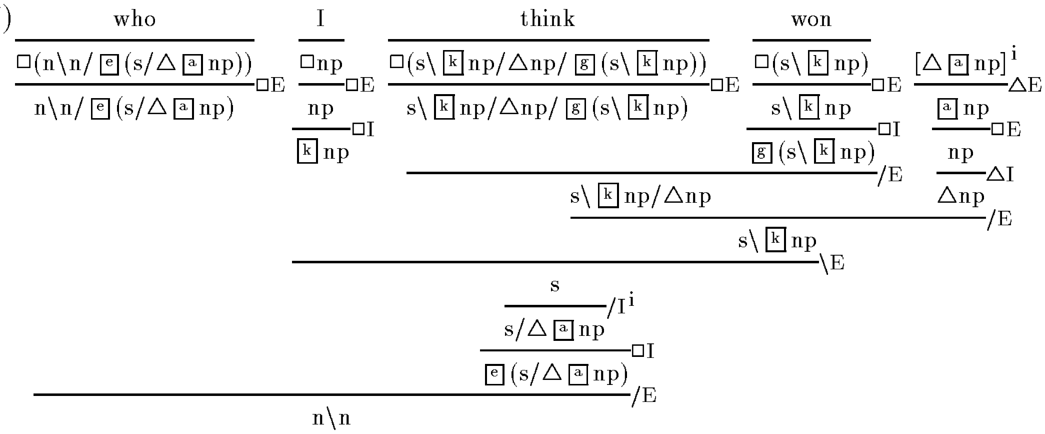
(3.66) a. $\Box(\mathrm{s}\backslash\boxed{\mathrm{k}}\mathrm{np}/\boxed{\mathrm{g}}\mathrm{s}) \Rightarrow \Box(\mathrm{s}\backslash\boxed{\mathrm{k}}\mathrm{np}/\triangle\mathrm{np}/\boxed{\mathrm{g}}(\mathrm{s}\backslash\boxed{\mathrm{k}}\mathrm{np}))$

b. $\lambda f \lambda g \lambda x.f(gx)$

Given our assumption that box modalities have no realization in the lambda semantics, the associated semantics for this rule corresponds to unary composition.

The following proof of the relative clause *who I think won* illustrates this treatment of embedded subject extraction:

(3.67)



Let us note some parallels between this account of embedded subject extraction and that of GPSG (Gazdar *et al.*, 1985). GPSG handles slash termination (corresponding to the 'gap' for extraction) using metarules. Slash termination for all extraction except subject extraction is allowed for by a metarule (STM1) whose operation depends on the presence of the lexical head which subcategorizes for the missing item. Since subjects are introduced as sisters to VP rather than the finite verb, STM1 cannot perform slash termination for them. Instead, a second slash termination rule (STM2) is required which operates at the 'next level up', i.e. at the level where the verb governing the embedded clause appears. This as follows:

(3.68) STM2:

$$\begin{array}{c} X \rightarrow W,\ V^2[+\textsc{subj},\ \textsc{fin}] \\ \Downarrow \\ X/NP \rightarrow W,\ V^2[-\textsc{subj}] \end{array}$$

The metarule can be paraphrased as follows: for any rule introducing a finite sentential complement, there is a corresponding rule which instead introduces a finite VP and where [SLASH NP] is marked on the mother node. The correspondence of this treatment to the present one is obvious, and becomes even stronger when we bear in mind that metarules are restricted to only creating new lexical ID rules from old, a restriction which embodies a claim that:

> "... metarules serve solely to express generalizations about the subcategorization possibilities of lexical heads." (Gazdar *et al.*, 1985, p59)

Given this, we can state a single interpretation which applies equally to the two accounts as follows: where a head subcategorizes for a finite sentential complement, an alternative relationship is possible where the head combines with a finite VP and transmits the subcategorization for the subject of that VP *marked as extracted.* Of course the two accounts differ in that the 'alternative relationship' is in one case realized by an additional phrase-structure rule and in the other by an additional modified type for the head. A further similarity is that the two accounts get the appropriate characterization of the **that-t* filter data by having a general set up which does not allow subject extraction, and then making special provision in the grammar to admit only the grammatical subject extraction cases.

This latter point also gives a similarity to the subject extraction treatment of Steedman (1987), discussed earlier in the chapter. Extraction of embedded subjects is not allowed by the general mechanisms of Steedman's grammar for English, i.e. the lexical type assignments and combinatory rule component (which most notably does not include disharmonic forward composition). Special provision is then made to allow for grammatical cases of subject extraction, by providing additional types for sentence embedding verbs, such as the following type for *believes*:

$$(s\backslash np)/(s\backslash np)/np$$

Using this type, the treatment of embedded subject extraction can be assimilated to the general treatment of non-peripheral argument extraction, as discussed earlier.[17] The correspondence between the two categorial accounts, however, is limited since the rule required to produce the additional CCG types does not have unary composition for its semantics.[18]

3.4.10 Subject/object extraction asymmetries

The account of embedded subject extraction we have presented provides a basis for explaining some well-known asymmetries between the extraction of subjects and objects (where 'object' is here used to mean general 'VP-internal' argument, rather than a particular grammatical relation), illustrated by the following examples:

(3.69) a. ?What do you wonder who read?

b. *Who do you wonder what read?

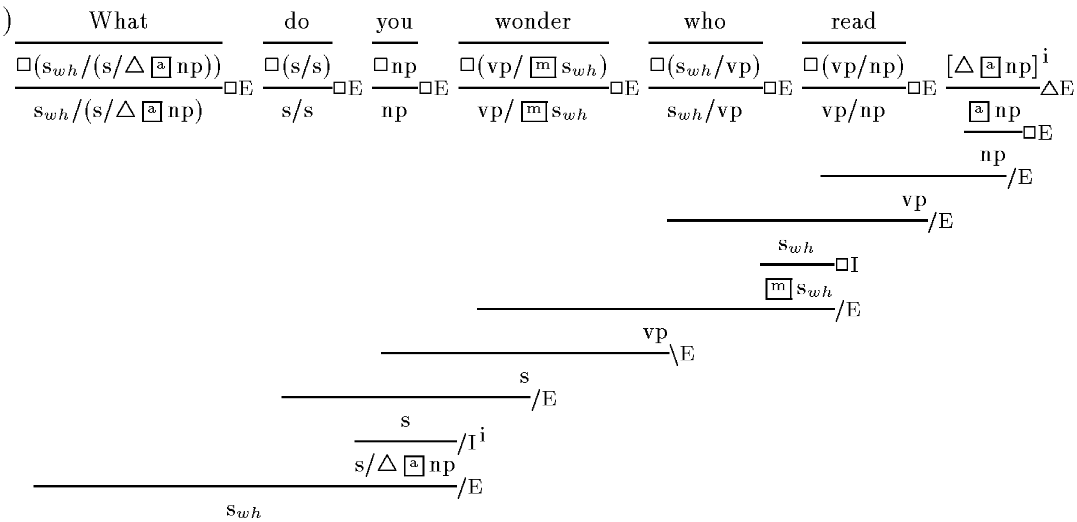
Both of these sentences involve *wh*-island constraint violations, because of extraction from an embedded question. However, the first example, where the object is 'long extracted', is clearly much better than the second, where the subject is 'long extracted'. This asymmetry is robust across a range of examples. We assume that the rule which generates sentence embedding verb types that allow for embedded subject extraction is restricted, not applying to verbs which take a [+wh] complement. Bearing this in mind, consider the differences

[17] Note that the type transition $s\backslash np/s \Rightarrow s\backslash np/(s\backslash np)/np$ is Lambek valid, although the CCG grammar does not allow for this type transition.

[18] It is interesting to note that a solution based on additional types such as $(s\backslash np)/np/(s\backslash np)$, which could be derived by a (non-**L**-valid) instance of division or unary composition, was explored in a draft version of Steedman (1987). This solution was later dropped, presumably because it resulted in unacceptable overgeneration, for example admitting **I believe ran he*, and with no basis for addressing its ungrammaticality within the grammar.

between attempts to analyse the two cases. The following shows a proof for (3.69a) (in which vp is again used to abbreviate the type of VPs, i.e. $s\backslash\boxed{k}np$ for the present account):[19]

(3.70)



In fact this proof is not correct, since we do not assume the constraint $\mathcal{L}_a \subseteq \mathcal{L}_m$ which would license the *wh*-island constraint violation. We might address the partial acceptability of this example as before in terms of speaker uncertainty about the penetrative modality marked on the fronted type for *what*, i.e. uncertainty between the modality shown and another one which would allow for the island violating extraction. Even such an alternative penetrative modality, however, would not allow for the sentence in (3.70b). Thus, consider the types for *who* that might be involved. The type $s_{[+wh]}/(s/\triangle\boxed{a}np)$ allows for long distance extraction, but, as we have seen, does not allow for subject extraction given the modality clash between its penetrative modality $\boxed{a}$ and the subject's boundary modality $\boxed{k}$. The other possible type for *who* $s_{[+wh]}/(s\backslash\boxed{k}np)$, assigned to handle 'string vacuous' movement of the subject, *is* marked with the modality $\boxed{k}$. However, since this type is not constructed using $\triangle$, it provides no basis for binding the embedded subject argument 'at a distance'. Furthermore, its $\boxed{k}$ modality would not license penetration of the boundary specified by *wonder.* Thus, even assuming 'uncertainties' that allow for the partial acceptability of (3.70a), example (3.70b) is still correctly predicted to be unacceptable.

3.4.11 Resumptive pronouns

This approach may provide a basis for handling phenomena involving resumptive pronouns. In a number of languages, a *resumptive pronoun strategy* may be used to construct sentences

[19] For this proof, we assume a treatment of *do*-support in which *do* is assigned a sentential modifier type. Although convenient for immediate purposes, this is not the approach we would take in general. Rather we believe that English subject-auxiliary inversion should receive an 'extraction' based treatment, similar to our proposals for Verb Second behaviour in Germanic languages which are presented in a later chapter.

involving ‘extraction’ (or what corresponds to extraction) which would otherwise be ruled out on the grounds of some island constraint violation. This is illustrated by the following sentences (although the use of resumptive pronouns in English in general gives poor results):

(3.71) a. *Which man do you wonder why mary loves?

b. ?Which man$_i$ do you wonder why mary loves him$_i$?

We can assign resumptive pronouns a type $\Box(\boxed{p}\mathrm{np}/\triangle\mathrm{np})$ with identity semantics (i.e. $\lambda x.x$). This type has a $\triangle$-marked argument which can be bound by the raised type of an extracted element (as we shall see). (Note that the $\triangle$ of the argument prevents the construction of a ‘constituent’ *him John* as a NP.) The type is ultimately a function into np and so may only appear in positions where NPs may appear. The type diverges from the general pattern of $\Box$-marking we have assumed in that the value np subtype of the function is $\Box$-marked. This additional box has the effect of allowing the resumptive pronoun’s type to ‘override’ the effect of boundaries appearing between the resumptive pronoun and some extractor. To illustrate the idea, we shall look at a simplified, somewhat abstract, example:

(3.72)
$$\frac{\mathrm{A}}{\Box(\mathrm{s}/\boxed{q}\mathrm{s})} \quad \frac{\mathrm{B}}{\Box(\mathrm{s}/\mathrm{np})} \quad \Rightarrow \quad \mathrm{s}/\triangle\boxed{r}\mathrm{np}$$

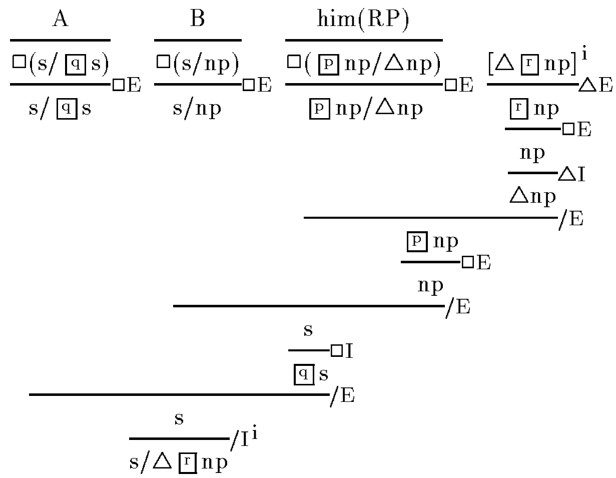
This combination corresponds to extracting the NP complement of B across the boundary specified by A (where the result type $\mathrm{s}/\triangle\boxed{r}\mathrm{np}$ is the argument required by an extractor having penetrative modality $\boxed{r}$). Such a combination can be derived as follows:

(3.73)
$$\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{A}}{\Box(\mathrm{s}/\boxed{q}\mathrm{s})}}{\mathrm{s}/\boxed{q}\mathrm{s}}{\scriptstyle\Box E} \quad \dfrac{\dfrac{\dfrac{\dfrac{\mathrm{B}}{\Box(\mathrm{s}/\mathrm{np})}}{\mathrm{s}/\mathrm{np}}{\scriptstyle\Box E} \quad \dfrac{\dfrac{[\triangle\boxed{r}\mathrm{np}]^{i}}{\boxed{r}\mathrm{np}}{\scriptstyle\triangle E}}{\mathrm{np}}{\scriptstyle\Box E}}{\mathrm{s}}{\scriptstyle /E}}{\boxed{q}\mathrm{s}}{\scriptstyle\Box I}}{\mathrm{s}}{\scriptstyle /E}}{\mathrm{s}/\triangle\boxed{r}\mathrm{np}}{\scriptstyle /I^{i}}$$

The [$\Box$I] inference is licensed only if $\mathcal{L}_{\mathrm{r}} \subseteq \mathcal{L}_{\mathrm{q}}$. Let us assume that it is not. Next consider the comparable combination with the resumptive pronoun, which is proven in (3.75):

(3.74)
$$\frac{\mathrm{A}}{\Box(\mathrm{s}/\boxed{q}\mathrm{s})} \quad \frac{\mathrm{B}}{\Box(\mathrm{s}/\mathrm{np})} \quad \frac{\mathrm{him(RP)}}{\Box(\boxed{p}\mathrm{np}/\triangle\mathrm{np})} \quad \Rightarrow \quad \mathrm{s}/\triangle\boxed{r}\mathrm{np}$$

(3.75)



The proof rests on an additional assumption that is discharged in the final inference. In the earlier proof (3.73), it is the box modality marked on this assumption (corresponding to the the penetrative modality of the extractor type) that determines whether extraction can be made across the boundary, by its effect on whether the introduction inference for the boundary modality is properly licensed. In (3.75), however, the additional assumption combines with the resumptive pronoun to give a subconstituent with type $\boxed{\scriptstyle p}\,\mathrm{np}$. Recall that the condition on the [$\Box$I] rule is that every path to an undischarged assumption must include an independent *subproof* of a box type whose defining language is appropriately related to that of the introduced modality. One possibility for the introduction of the boundary modality in (3.75) to be properly licensed is if $\mathcal{L}_{\mathrm{p}} \subseteq \mathcal{L}_{\mathrm{q}}$ so that the subproof of the $\boxed{\scriptstyle p}\,\mathrm{np}$ type meets this condition. In this case the modality marked on the additional assumption becomes irrelevant.

Note that since the semantics assigned with this type is identity, binding the $\triangle$-marked argument of this function is semantically equivalent to directly binding the argument of the function which combines with the resumptive pronoun. Thus, the result semantics read off (3.75) is as in (3.76a), which, since the pronoun has identity semantics, simplifies to (3.76b). (Note that (3.76b) is precisely the semantics that would be read off the proof (3.73).)

(3.76) a. $\lambda y.\mathrm{A}'(\mathrm{B}'(\mathrm{him}'\,y))$

b. $\lambda y.\mathrm{A}'(\mathrm{B}'\,y)$

We have seen that with a resumptive pronoun type:

$\Box(\boxed{\scriptstyle p}\,\mathrm{np}/\triangle\mathrm{np})$

the modality marked on the value subpart can license the binding of its $\triangle$-marked argument across boundaries. If we took $\mathcal{L}_{\mathrm{p}}$ to be the same as $\mathcal{L}_{\emptyset}$ (which we assume to be a subset of all other defining languages used) then binding of the pronoun would not be sensitive to any modal boundaries. It is not clear whether the binding of resumptive pronouns has *no* island sensitivity. However, the approach allows that the boundary sensitivity of binding the pronoun can be determined by selection of the modality marked on the pronoun.

Note that the $\triangle$-marked argument of our resumptive pronoun type makes it just like any argument that is undergoing extraction. Evidence which supports this view of resumptive pronouns is given by Maling and Zaenen (1982) and Engdahl (1982) concerning coordination in Swedish. Swedish obeys the Coordinate Structure Constraint in that it is impossible to coordinate two clauses where one contains a 'gap' for left extraction and the other does not. As shown in (3.77), the insertion of a resumptive pronoun in place of the extracted element does not improve things (only the relevant extraction dependency is indicated).

(3.77) *Jag läste en bok$_i$ som jag redan glömt [$_{\bar{s}}$ vem som skrivit _$_i$/den$_i$] och [$_{\bar{s}}$ om det är torsdag idag]
I read a book that I have already forgotten [$_{\bar{s}}$ who$_i$ that wrote _$_i$/it$_i$] and [$_{\bar{s}}$ if it is thursday today]

(Note that (3.77) is *not* ruled out by a *wh*-island violation, since this is allowed in Swedish.) However, it is possible to conjoin two clauses where one contains a gap and the other a resumptive pronoun:

(3.78) Jag läste en bok$_i$ som jag redan glömt [$_{\bar{s}}$ vem som skrivit _$_i$] och [$_{\bar{s}}$ hur den$_i$] slutar]
I read a book that I have already forgotten [$_{\bar{s}}$ who$_i$ that wrote _$_i$ and [$_{\bar{s}}$ how it$_i$ ends]

This possibility accords with our account, where the 'binding argument' of the resumptive pronoun and the 'gap' both correspond to $\triangle$-marked np arguments.

This account of resumptive pronouns has the problem of predicting that resumptive pronouns can occur not only within islands, but also *anywhere* where a 'gap' can appear. Thus, the following is predicted to be acceptable:

(3.79) *Which man did Mary like him?

Resumptive pronouns in general exhibit roughly complementary distribution to 'gaps', so that a resumptive pronoun can only be used if a 'gap' is either disallowed or in some way disfavoured (perhaps presenting difficulties for parsing). One possibility for explaining the unacceptability of (3.79) is to argue that the distribution of resumptive pronouns is pragmatically determined. Dowty (1980) suggests a "neo-Gricean conversational principle" which might be paraphrased as: "Where two equally simple expression A and B are available to express a meaning X, but where A unambiguously means X whilst B is ambiguous between meanings X and Y, then B should be reserved for expressing Y." (This principle is used as a basis for a pragmatic account of disjoint reference in binding, which we shall discuss in a later chapter.) Whereas a 'gap' is unambiguously a site for extraction, a pronoun occurrence may have either anaphoric, deictic or resumptive usage. Thus, according to the neo-Gricean principle, we expect that the use of a 'gap' should be favoured for extraction. However, when the use of a 'gap' is ruled out by island constraints etc, the principle does not apply and so resumptive use of a pronoun should be possible.

3.4.12 Language variation and island constraints

Scandinavian languages are of particular interest with respect to the treatment of island constraints since they exhibit considerably greater freedom for extraction. For example, Engdahl (1982) points out that Swedish allows extraction from embedded questions, relative clauses, and sentential complements of nouns. We shall not work through the treatment of these cases in Swedish here, but it should be clear that the present approach should allow such additional possibilities to be straightforwardly encoded. We take it to be an advantage of this approach that such differences between languages can directly addressed within the grammar without needing to claim that the languages exhibit fundamental differences in terms of syntactic structure or government behaviour, etc.

One interesting difference between Swedish and English arises in respect of extraction *from* fronted constituents. This is generally taken to be impossible in English. Consider, for example, the following sentences:

(3.80) a. *Which books did you wonder whose reviews of had annoyed me? (Gazdar *et al*, 1985)

b. *Which actor do you know which article about Mary secretly wrote?

The unacceptability of such examples may explained in a number of ways. For example, Gazdar *et al* (1985) choose to specifically block such examples with a Feature Cooccurrence Restriction that rules out cooccurrence of the [SLASH] and [WH] features. A simpler hypothesis is that these examples be seen as just another case of the island status of embedded questions. By the latter view, we may expect that extraction from fronted constituents will be allowed in languages with fewer constraints on extraction, particularly where extraction from embedded questions is allowed. As we noted earlier, Swedish allows extraction from embedded questions. Engdahl (1982) points out that Swedish does allow the appearance of 'gaps' inside fronted constituents, and provides the following example:

(3.81) Nixon$_j$ undrar jag [hur många porträtt av $\_{j}$]$_i$ det fortfarande hänger $\_{i}$ i Vita huset.
Nixon$_j$ I wonder [how many portraits of $\_{j}$]$_i$ there still are $\_{i}$ in the White House

3.4.13 Discussion: island constraints as syntactic or semantic

A number of authors have argued for the involvement of *semantic* effects on extraction and island behaviour. Recently, Szabolcsi and Zwarts (1990) have proposed a semantically-based categorial theory of island constraints, which focuses on the semantic properties of composed functions. In particular, the account includes claims of the form: "the extraction of a phrase having properties X is allowed provided that the constituent formed by composing material intervening between 'gap' and 'landing site' has semantic properties Y". Detailed discussion of this account is beyond the scope of this chapter.

We shall, however, note one objection to such an approach which suggests that this cannot form the basis of the treatment of island constraints *in general*. This is that such an account inevitably predicts that relatively closely related languages will exhibit very similar island behaviour. (This is not necessarily predicted for distant languages since they may show much less correspondence between lexical items and their semantics.) However, different languages can exhibit quite striking differences in island behaviour, even languages that exhibit as many superficial similarities as English and Swedish. It seems that our approach to handling island constraints must provide the means to encode language specific stipulations concerning island behaviour differences.[20]

We should note one way in which 'semantic effects' on extraction might be properly addressed, at least in part, using the 'syntactic' apparatus we have described. We have seen how it is possible to have sequences of related modalities, for example, as we would find given a set of defining languages related as follows:

$$\mathcal{L}_{\mathrm{v}} \subseteq \mathcal{L}_{\mathrm{w}} \subseteq \mathcal{L}_{\mathrm{x}} \subseteq \mathcal{L}_{\mathrm{y}} \subseteq \mathcal{L}_{\mathrm{z}}$$

This gives a 'hierarchy' of related modalities. Lexical items which form a class may also enter in hierarchical orderings based on various properties, including semantic properties. It is possible to envisage in the case of, say, sentence embedding verbs, that the assignment of boundary modalities in the construction of lexical types might be required to pattern

[20] This argument against purely semantic accounts of island constraints is also of note in relation to Morrill's (1989) unimodal treatment of linguistic boundaries. As we noted earlier, Morrill takes the $\Box$ operator to be an *intensional* type constructor, its distribution on lexical types reflecting intensional characteristics of word meanings. But then we shall not expect closely related languages to exhibit radical differences in island behaviour, since, under Morrill's view of $\Box$, this would imply radical differences in the intensional characteristics of word meanings.

with some hierarchy over the verbs' properties, possibly a semantic one, i.e. so that the selection of modality from some sequence of related modalities might have to be in accordance with the semantic ordering over the items. This would give what might be called a *grammaticalization* of the relevant semantic hierarchy, with the consequence that some syntactically-based process (such as extraction) might show effects that pattern with the semantic hierarchy as a consequence of the process's sensitivity to modal boundaries.[21]

The difference between such syntactically mediated involvement of a semantic hierarchy and direct involvement of it arises in respect of language variation. Although related languages may similarly require that the assignment of boundary modalities must pattern with some semantic hierarchy, it may be that the languages vary as to which range of the hierarchy of modalities the boundary modalities are selected from. Then, although the languages might show considerable differences in respect of island constraints, they will show the same essential *trends* in the relation between verb semantics and island behaviour.

Consider, for example, the hypothesis that the patterning of sentence embedding verbs as non-bridge, bridge and super-bridge mirrors some hierarchy on their semantic properties (a fairly reasonable hypothesis). Then, various possibilities obtain for the island behaviour we might observe amongst a group of related languages. It might be that a particular language allows that none of the verbs act as bridges at all, or that all of the verbs act as super-bridges. Or, it might be that the only verbs which acts as bridges at all correspond to English super-bridges. What we would not expect to find is a language which has bridge verbs corresponding to the bridge verbs of English, but at the same time shows non-bridges corresponding to English super-bridges. Such discussion is clearly very hypothetical, but hopefully it illustrates the idea of how the grammaticalization of semantic properties may cause syntactically-based processes to exhibit effects that pattern with semantic hierarchies.

Even given such possibilities for syntactically-mediated semantic effects, it seems likely that some phenomena will turn out to require explanation in terms of the *direct* involvement of semantic properties. The appropriate demarcation between phenomena requiring syntactic and semantic explanation is something that will only become clear after considerable further research in which attempts are made to construct accounts framed in these alternative terms.

[21] Of course, other cases where some semantic property becomes grammaticalized are known, a familiar case being gender. As this case illustrates, once some semantic property has been grammaticalized, the process of language change may cause the grammaticalized manifestion to 'drift away' from the original semantic property in its distribution.

3.5 Conclusion

This chapter has presented an account in which there are many related notions of boundary, and in which the sensitivity of any process (such as extraction) to the presence or absence of a particular kind of boundary is something that can be directly encoded within the grammar. This is clearly a very radical view of how island constraints and other phenomena involving linguistic boundaries should be handled. We believe that many of the problems that have arisen for handling locality effects on syntactic processes are the consequence of adopting too simplistic a view of the nature of linguistic boundaries and the relations that may obtain between them. Attempts to deal with such problems have often led to highly complicated accounts that are still laden with problems.

Even so, the goal of this chapter has been fairly unambitious. This has been to develop a grammatical apparatus to address locality effects which are based in the syntactic domain. We have not attempted to deal with the possibility of direct *semantic* effects on extraction, although we have discussed some possibilities for how aspects of island behaviour might be seen to pattern with some semantic property as a consequence of the grammaticalization of the property. An important goal for future research is to establish the appropriate demarcation between phenomena requiring syntactic and semantic explanation.

Chapter 4

Word Order and Obliqueness

This chapter is primarily concerned with the treatment of word order in categorial grammar. In particular, a new approach is proposed for the treatment of word order within the extended Lambek calculus framework. What is different about this treatment (for CG, at least) is that it factors apart the specification of the order of a head's complements from the specification of the position of the head relative to them. Such a move may have a number of advantages, for example in providing a basis for the treatment of Verb Second phenomena in Germanic languages. An important consequence of this move is that it allows the incorporation of an account of grammatical relations (GRs), and, in particular, a modified version of the account of GRs proposed by Dowty (1982a) within Montague Grammar. We begin by considering the Montague Grammar approach, and the theory of GRs proposed by Dowty within this framework.

4.1 Montague Grammar and grammatical relations

Montague Grammar is a CG framework that originated in the work of Richard Montague, who was primarily concerned with issues of natural language semantics. It was later developed in application to linguistic problems by a number of authors (e.g. Partee, 1976; Thomason, 1976; Bach, 1979; Dowty, 1978).

In contrast to directional CGs, Montague Grammar uses categorial types that are formed using *directionless* slash connectives. Functions specify the type of the argument with which they combine, and the type of the phrase that results from this combination, but do not specify the relative order in which the function and argument must occur. Instead, Montague Grammar provides a particular rule for each different combination of a function with its argument (different, that is, in terms of the types of the phrases combined), and each rule is associated with an *operation* which specifies how the string associated with the result of the combination is derived from the strings associated with the phrases combined. These operations are responsible for the treatment of word order, as well as tense and case marking and the addition of prepositions. As we shall see, these string combining *operations* go beyond simple concatenation.

Dowty (1982a) points out that this approach provides a natural basis for a theory of grammatical relations (i.e. notions such as *subject* and *direct object*). By this view, the syntactic rules used in all natural languages are largely the same, the primary focus of variation between languages being the specification of the string operations associated with rules. Dowty proposes that it is the language universal syntactic rules (aside from their language particular string combining operations) that define grammatical relations (GRs). Thus, the *subject* is defined to be that complement which combines with an intransitive verb to give a sentence, the *direct object* is defined to be that complement which combines with a transitive verb type to give an intransitive verb, whilst an *indirect object* is defined to be that complement which combines with a ditransitive verb type to give a transitive verb. This means of defining GRs unavoidably places an order upon them, an ordering which corresponds to the traditional notion of obliqueness. Thus, for any verb type of the form (4.1) (where the dots stand for possible further arguments), with x occurring later in the argument order than y, x is a less oblique complement of the verb than y.

(4.1) s/../x/../y/..

Dowty's theory aims to give a *universal* characterization of GRs which does not treat them as primitives (as in Relational Grammar and Lexical–Functional Grammar) and which is not affected by language particular word order facts (a problem when GRs are defined in terms of phrase structure configurations, e.g. as in the Transformational Grammar account).

We next briefly illustrate the Montague approach, using the syntactic rules stated in (4.2).[1] For convenience, we omit each rule's semantic operation, and other details (such as agreement marking by string operations). The string operation for each rule is named beside it as F_n for some n, and a description of the operation *for English* is given (where in each case α is the string of the first phrase mentioned in the rule and β that of the second,

[1] The types shown are given using the standard Montague Grammar notation, where t is the type of sentences, T abbreviates the type of noun phrases ('term phrases'), IV abbreviates the type of intransitive verbs, TV transitive verbs and TV/T ditransitive verbs.

and where $\alpha{\sim}\beta$ denotes the result of concatenating α and β).

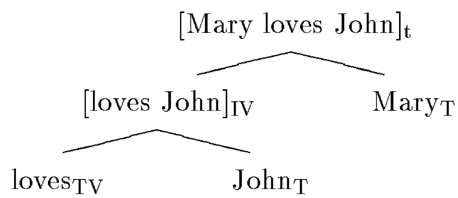
(4.2) a. Subject–Predicate Rule: F_1: IV + T $\Rightarrow$ t
$F_1(\alpha, \beta) = \beta{\sim}\alpha$

b. Verb–Direct Object Rule: F_2: TV + T $\Rightarrow$ IV
$F_2(\alpha, \beta) = \gamma$ (where γ is derived by inserting β immediately after the first word in the string α)

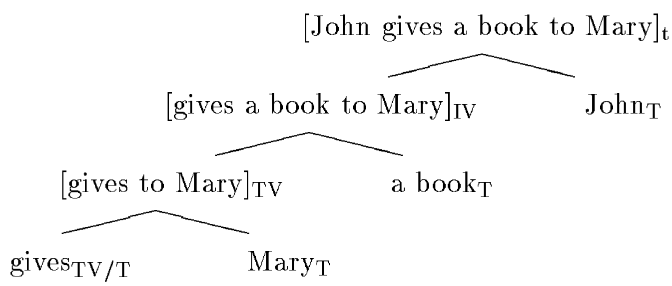
c. Verb–Indirect Object Rule: F_3: TV/T + T $\Rightarrow$ TV
$F_3(\alpha, \beta) = \alpha{\sim}\mathit{to}{\sim}\beta$

The use of these rules is illustrated in the following two analyses (where each binary subtree shows the strings and types associated with two phrases combined and the result of their combination).

(4.3)



(4.4)



Note in particular the combinations under the Verb-Direct Object Rule, which is associated with the non-concatenative operation *right wrap*, whereby the result string is derived by inserting the string of the noun phrase after the first word (i.e. the verb) in the string of the (possibly complex) transitive verb. The instance of right wrap in the first analysis is indistinguishable from concatenation since the TV is lexical rather than phrasal. In the second analysis, the IV phrase *gave a book to Mary* is formed by combining the TV *gave to Mary* with the term phrase *a book*.

The Montagovian approach to GRs has provided the basis for accounts of a number of phenomena, including relation changing (Bach, 1980; Dowty, 1982a, 1982b), control (Bach, 1979), and binding (Bach and Partee, 1980; Chierchia, 1988). As Jacobson (1987) emphasizes, notions such as 'subject' and 'direct object' play no direct role in the grammar under this account. Rather, the properties of, say, subjects are in fact properties of intransitive verbs, the combinations they are involved in and the rules that they undergo.

Before we go on, we should note a divergence between the Montagovian use of the terms direct and indirect object and the perhaps more traditional use of these terms. In the 'traditional' usage, the label *direct object* would be applied the linearly first object (i.e. *a book*) in (4.5a), and to the linearly second object (i.e. *a book,* again) in the Dative Shift variant in (4.5b), whereas *to Mary* would be called a *prepositional object* in (4.5a) and *Mary* an *indirect object* in (4.5b.)

(4.5) a. John gave a book to Mary

b. John gave Mary a book

The Montagovian use of these terms is different. The term *direct object* is used to refer to a NP complement that is the next-to-last subcategorized argument (i.e. immediately before the subject argument). In a sentence like (4.5a), the direct object is the NP complement *a book*, but in (4.5b) the direct object is taken to be the linearly first complement *Mary* (which follows given the Montagovian view of Dative Shift, discussed later in the chapter). The linearly second object in (4.5b) would be called an *oblique* object.

In what follows, we adopt the Montagovian use of these terms. The reader should bear this point in mind, otherwise confusion will result. In practice, we will favour the use of a term *second object*, which collapses together the Montagovian notions of indirect and oblique object, referring to subcategorized NP or PP complements which are sought immediately preceding the direct object in argument order.

4.2 Concatenative CGs

There are an number of CG approaches which attempt to provide a highly general system for stating possible combinations. The Lambek calculus is perhaps the most striking instance of this, with its free calculus of type combinations. Such CG approaches typically adopt an assumption which we might call the *concatenative assumption,* requiring that the string associated with the type that results from some combination is derived purely by concatenation of the strings associated with the types combined. Thus, for CGs that use a *directional* notation, slash directionality can be viewed as specifying *order of concatenation* of the strings associated with the functor and argument expressions.

We have seen that the Montagovian theory of GRs is based around defining GRs in terms of combination-specific rules. As it stands, the theory is obviously incompatible with approaches that attempt to avoid the use of such combination-specific rules. However, it is possible to envisage a modified version of Dowty's theory that such approaches might adopt, which looks directly to the subcategorization of functional types. By this view (which is essentially that of HPSG), the order of subcategorization of a functional type is taken to encode relative obliqueness, allowing the possibility of defining GRs directly in terms of obliqueness (as read off subcategorization). Thus, a *subject* might be defined as the least oblique complement of a verb (i.e. the last complement sought by the functional type of the verb), a *direct object* the next-to-least oblique NP complement, and so on.

Unfortunately, even this modified version of Dowty's account has not been available to CGs that adopt the concatenative assumption ('concatenative CGs') because of problems that arise with the cases for which Montague Grammar uses wrapping operations. Consider the sentence *Mary gave vodka to John.* Under the concatenative assumption, it is not possible to construct a type for *give* whose argument order corresponds to obliqueness that yields the correct constituent order. Thus, *to John* is the most oblique complement of the verb and so must be the *first* argument sought by the verb's type. But then, by the concatenative assumption, *to John* would have to appear next to the verb in the observed word order. For concatenative CGs, the type that would commonly be assigned to *give* to allow for this sentence is (s\np/pp/np), in which the argument order does not correspond to relative obliqueness.

4.3 A new model of word order in categorial grammar

In this section, a new model of the treatment of word order in CG is proposed, within the extended Lambek calculus framework. What is new about this approach (for CG) is that it factors apart the specification of the order of a head's complements from the specification of the location of the head relative to them. Certain linguistic phenomena suggest the appropriateness of this factoring. A prime instance is the Verb-Second behaviour of Germanic languages, exemplified by the following sentences of Dutch:

(4.6) a. ... omdat Jan de appels aan Marie gaf
... because Jan the apples to Marie gave
'... because Jan gave the apples to Marie'

b. Jan gaf de appels aan Marie
Jan gave the apples to Marie
'Jan gave the apples to Marie'

c. gaf Jan de appels aan Marie?
gave Jan the apples to Marie
'Did Jan give the apples to Marie?'

The clause in (4.6a) is a simple Dutch embedded clause exhibiting characteristic verb-final order. (4.6b) shows the corresponding main clause declarative, where the finite verb appears following the first major constituent of the clause, in this case the subject. (4.6c) shows a related yes/no question, in which the verb appears in initial position. Such examples pose a considerable problem for standard CGs, particularly in regard of explaining what the different cases have *in common* in their constituent order. One might attempt to handle such examples by simply listing different verb types that give rise to each of the word orders, but this would give no explanation of why, when the position of the verb changes, the position of everything else stays the same. For the model of word order to be proposed, where the position of heads is in general determined separately from the order of their complements, the treatment of such examples does not present a problem.

Rather than develop the new account in relation to Germanic Verb Second (a task which we leave for later in the chapter), we begin instead by focusing on a different consequence of separating the specification of complement order and head position, namely that it allows a modified version of Dowty's account of GRs to be adopted. In the previous section we noted that Dowty's theory of GRs (or even a modified version of it) has been unavailable to concatenative CGs because of predictions that arise for the relation of complement order and obliqueness which are not borne out empirically. Since the account of word order to be presented separates the specification of the order of a head's complements from the specification of the position of the head, it turns out that such problems are avoided. In what follows, we develop the account in relation to English, assuming a version of Dowty's theory, and paying particular attention to the relation between obliqueness and constituent order. We begin with some discussion of the structure of the lexicon, since to allow the new treatment of word order we are forced to adopt a more complicated model of the lexicon than is generally assumed in categorial work.

4.3.1 A note on the structure of the lexicon

According to a standard view in categorial work, the lexicon is taken to consist of little more than a set of type assignments to words, where each of these types is available to the syntactic domain as a lexical categorization of the relevant word. Such a view may be extended to allow treatment of certain phenomena (e.g. relation changing phenomena) in terms of lexical rules, which give additional type assignments to words on top of those initially assigned. Again, all types are available to syntax.

In contrast to this view, we assume that lexical assignments to words are objects that are built up in several stages. This notion of 'stages in construction' is handled by allowing the lexicon to consist of a number of distinct 'compartments' or subdomains, where the types assigned to any given word will differ in the different compartments. It is not assumed that all types associated with a word in any compartment are available to syntax as possible lexical categorizations. Instead, there is a designated 'final' compartment which specifies the type assignments that are made available to syntax. The type assignments in different compartments are related by lexical rules. These rules specify one or more compartments from which their input types are drawn, and a compartment to which their output types are assigned. Certain compartments specify initial type assignments to words (i.e. ones that are not assigned as a consequence of the operation of a lexical rule).

As a simple example, consider our assumption (discussed in Chapter 2) that *every* lexical type made available to syntax is of the form $\Box$X, i.e. having a single all-encompassing $\Box$ operator. Rather than assume that this requirement holds at all stages in the lexicon, we might let the addition of the operator be the final step in the construction of lexical types. This could be effected by a lexical rule such as the following:

(4.7) 'Box Addition' Rule:

Syntax:	$X \Rightarrow \Box X$
Semantics:	$\lambda i.i$
Input domains:	{e, g, k}
Output domain:	z

(The semantics of the rule is stated as a lambda expression which would apply to the semantics of the input type, yielding the semantics of the output type. Since we assume that box modalities are semantically 'vacuous', the rule semantics is identity.) Observe that we have stated input and output domains for the rule, specified as letters which are taken to 'name' particular compartments of the lexicon. The output domain for the rule would correspond to the 'final' compartment of the lexicon, from which type assignments are made available to syntax.

Such a view of the lexicon could straightforwardly be given a declarative formulation, in which the different compartments are viewed as sets of objects, each object being a type/meaning assignment to a word, and where each lexical rule specifies requirements on the membership of certain sets (corresponding to the rule's output domain) in terms of the membership of other sets (corresponding to the rule's input domains). In what follows we shall not attempt to give our account so formal a statement. However, the reader should bear this view of the lexicon in mind during the discussion that follows.

4.3.2 The factors that determine word order

Under the new approach, observed word order is taken to result from the interaction of the following three factors:

(4.8) A the order of subcategorization by a head for its complements (corresponding to obliqueness),

B the directionality of each argument

C a (lexical) process of *head location*, which determines the position of the head relative to its complements

As we shall see, factors (A) and (B) together determine the linear order of the head's complements. Were it not for factor (C), they would also determine the position of the head relative to its complements. Although this approach to handling word order is meant to apply to heads in general, we initially consider only the case of verbs.

To implement this view of word order, we hypothesize the existence of types for heads which have not undergone head location. These types, which we call *prelocation* head types, are not available to syntax. Head location is effected by a lexical rule that takes prelocation types as input, giving rise to what we call *located* head types. For verbs of English, head location causes the verb to appear following its subject (if present) and preceding its non-subject complements.

4.3.3 Complement order

We begin by considering abstractly the relation between subcategorization and the linear order given for a head's complements. This is determined by the *prelocation* type for a head. The process of head location affects solely the position of the head relative to its complements, and not the relative order of the complements. As an example, consider a function F into type X, requiring arguments A and B, where B is sought first. With the directionality of the arguments, we get four possible types for the function, which are shown in (4.9), together with the linear order these give rise to for A, B and F, and also separately the relative linear order of just A and B:

(4.9)	a.	X/A/B	F B A	B A
	b.	X/A\B	B F A	B A
	c.	X\A/B	A F B	A B
	d.	X\A\B	A B F	A B

The position at which F occurs does not concern us here, since the position of heads is to be determined separately. Looking at the covariation between the type of the functor and the relative order of A and B, we can see that the latter seems to depend solely upon the directionality of the argument for A, and seems unaffected by the directionality of the argument for B. It turns out that the important factor here is that A occurs later than B in the argument ordering specified by F, i.e. that it is less oblique than B (in terms of the modified version of Dowty's theory we assume). In general, for any two arguments A and B (not necessarily successive) of some function such that A is less oblique than B, if A is leftward directional, A precedes B, and if A is rightward directional, it follows B. Generalizing this relation with respect to other arguments, we arrive at an observation that the assignment of leftward or rightward directionality to some argument will cause it to uniformly precede or follow all more oblique complements of the function.

The combination of argument order and directionality assignment is essentially all there is for specifying canonical complement order for concatenative CGs. Hence, this observation about what is basically a technical constraint that emerges from our syntactic apparatus becomes a prediction about a universal constraint on possible (canonical) constituent orders, as follows:

(4.10) Obliqueness Constraint on Constituent Order:
A complement of a head must uniformly precede or follow all more oblique complements of the same head.

This constraint rules out certain orders as possible canonical orders. For example, it predicts that no language may exhibit a canonical constituent order in which a verb's subject occurs *between* its direct and second object complements. To our knowledge, the predictions of this constraint are borne out, i.e. no 'configurational' language exhibits such an order as a canonical order. Of course, we can expect to see cases which violate the constraint in languages that exhibit a considerable extent of free word order, and other cases where processes apply that give rise to non-canonical orderings. For example, such 'forbidden' orders may arise in Verb Second languages like Dutch, where fronting of a direct object may cause it to precede the subject.

We assume the conditions stated in (4.11) on the assignment of directionality to verb arguments in prelocation types for English. Note that in the modified version of Dowty's theory adopted here, we take GR labels to be defined in terms of a combination of relative obliqueness and type. Thus, a *subject* is the last (and therefore least oblique) argument sought by a (prelocation) verb type, a *direct object* is a next-to-last NP or (subcategorized)

PP argument, and a *second object* is a second-to-last NP or (subcategorized) PP argument.

(4.11) Complement Ordering Principles of English:

a. Subjects are sought to the left (i.e. are leftward directional)
b. Direct objects are sought to the left
c. Second objects are sought to the right
d. Non-subject verbal complements are sought to the right

These 'ordering principles' are to a large extent forced on us by the facts of English word order. Since subjects come first in English complement order, they must be assigned leftward directionality in prelocation types, as must direct objects, since they precede second objects. The directional requirement for second objects can be argued for on the basis of facts about the ordering of particles (as we will discuss later in a section devoted to the treatment of particle shift in English). Concerning (4.11d), note that Bach (1979, 1980) argues that the VP complement of subject control verbs is less oblique than the object NP. This forces us to assign rightward directionality for the VP argument to get the appropriate word order, as in *Mary promised John to go.* The complement ordering principles in (4.11) are sufficient for the purposes this chapter, but we certainly don't claim that they are the 'final story'. A correct and complete specification of the complement ordering principles of English is a topic requiring further research.

The complement ordering principles in (4.11) give rise to prelocation types for English verbs such as those in (4.12).

(4.12)

run:	s\np	eat:	s\np\np
put:	s\np\np/pp	give:	s\np\np/pp
will:	s\np/(s\np)	want:	s\np/□(s\np)
believe:	s\np/□sp	tell:	s\np\np/□sp
persuade:	s\np\np/□(s\np)	promise:	s\np/□(s\np)/np

Consider the prelocation type for *give.* If available to syntax, this type would give word orders such as: *John a book gave to Mary.* Of course, this type is *not* available to syntax. Also, it is important to bear in mind that a fundamental claim of our approach is that the position of heads and their complements is determined independently, so that head position in the constituent order that *could* be projected from a prelocation type has no particular significance. Some readers may feel, however, that the approach would lack plausibility if the word order (*inclusive* of head position) that could be projected from the prelocation type was one that did not appear to be a possible word order of any language. It is interesting to note in this regard that (as reported by Travis (1989)) the language Kpelle exhibits precisely such a word order, i.e.

$$\text{Subject} < \text{Object} < \text{Verb} < \text{(argument) PP}$$

4.3.4 Head location

We shall next consider the process of head location, and its treatment as a lexical rule. We saw in Chapter 2 that flexible CGs allow extraction to be handled by assigning appropriate lexical types to 'extracted elements' — types which in effect 'abstract over' the missing element in the domain from which it is missing, this treatment being possible given the flexible type assignment allowed by the syntactic calculus.

This same approach can be used to deal with the head location process. In particular, for some head that has prelocation type H, we can allow it to occur left peripherally in phrases of type X by giving it a lexical type $X/(X/\triangle H)$ (or equivalently $X/(X\backslash\triangle H)$). Such types can be generated by lexical rules of the following general form:[2]

(4.13) a. Head Location:

$$H \Rightarrow X/(X/\triangle H)$$

b. $\lambda f \lambda g.(gf)$

To get the head to appear at some position *amongst* its own complements, X must be some 'natural projection' of H, i.e. a type that could be gained by combining the head with some of its complements in order of decreasing obliqueness. The semantics of the rule is stated in (4.13b) as a lambda expression that applies to the meaning of the input type.

The head location rule for English verbs will cause them to appear left peripheral to their own verb phrase projection. This can be stated as follows:

(4.14) a. English Verb Location rule:

$$vp\phi \Rightarrow vp/(vp/\triangle(vp\phi))$$

b. $\lambda f \lambda g.(gf)$

In this rule, we use vp to abbreviate possible VP types of English (i.e. types of the form $s\backslash A$, where A is any possible subject type of English). Also, we use $(vp\phi)$ to stand for any type vp or function into vp. The located verb types produced only allow the verb to appear left peripherally to its own VP projection, not those of any dominating verbs. Thus, firstly, the located types require the verb to appear left peripherally to some VP projection bearing the same features for tense etc as the prelocated verb type[3] (since in the located type $vp/(vp_\alpha/\triangle(vp_\beta\phi))$, the feature markings α and β are identical). Secondly, the located types do not allow any modal boundaries to intervene between the position where it appears

[2] This rule is essentially an instance of type-raising, as its semantics clearly reveals, although note that it is not a transition allowed by the syntactic calculus.

[3] We assume that features for tense, etc, are marked on the 'innermost' value subtype s of the overall verb type.

and the site where its complements appear. These conditions can only be satisfied when the verb appears left peripherally to its own VP projection.

Under the model of the lexicon we assume, the head location rule takes its input from one or more compartments of the lexicon, compartments which specify prelocation types for verbs, and assigns its output to some further compartment. The rule will apply to all verb types in these input compartments since they will all be of the form specified for the rule's input. In fact, the only lexical process that need apply to the types produced by this rule is 'box addition', and so the compartment to which the located types are assigned will be one named as an input domain for the box addition rule stated in (4.7).

When applied to intransitive verb types vp, the rule yields types of the form $\mathrm{vp}/(\mathrm{vp}/\triangle\mathrm{vp})$ which might seem a little unusual since, with this type, the extraction domain of the verb does not contain any of the verb's arguments, whereas our approach requires that the extraction domain must contain *some* lexical material (i.e. because the Lambek calculus does not allow the assignment of any types to the empty string). Such types are not useless, however, since they provide for derivations in which the extraction domain contains only an adverbial. However, the type does not allow for the derivation of a simple intransitive clauses such as *John runs.*

One possibility here is to remove the restriction, standard in Lambek work, that types cannot be assigned to empty strings. For the natural deduction formulation, the restriction is enforced by stipulating that any independent proof must contain at least one undischarged assumption. For the sequent formulation, it is stipulated that in any sequent $\Gamma \Rightarrow \mathrm{x}$, Γ must be non-empty. If the restriction is removed, then the type-transition:

$$\mathrm{vp}/(\mathrm{vp}/\triangle\mathrm{vp}) \Rightarrow \mathrm{vp}$$

would be admitted by the syntactic calculus (as shown by the proof in (4.15)), and so the located type $\mathrm{vp}/(\mathrm{vp}/\triangle\mathrm{vp})$ would allow also for simple intransitive clauses. (Observe that (4.15) includes a subproof of $\mathrm{vp}/\triangle\mathrm{vp}$ which has *no* undischarged assumptions, corresponding to a type-transition $\Rightarrow \mathrm{vp}/\triangle\mathrm{vp}$).

(4.15)
$$\dfrac{\mathrm{vp}/(\mathrm{vp}/\triangle\mathrm{vp}) \qquad \dfrac{\dfrac{[\triangle\mathrm{vp}]^{i}}{\mathrm{vp}}\triangle E}{\mathrm{vp}/\triangle\mathrm{vp}}/I^{i}}{\mathrm{vp}}/E$$

Allowing types to be assigned to empty strings, however, has some undesirable consequences. For example, *very* might have a type $(\mathrm{n}/\mathrm{n})/(\mathrm{n}/\mathrm{n})$ allowing it to act as an intensifier of adnominals, as in e.g. *a very fat man.* Removing the restriction would allow a type transition:

$$(n/n)/(n/n) \Rightarrow n/n$$

so that the word would also have an adnominal type, admitting **a very man*. For this reason, we shall not make the move of allowing types to be assigned to empty strings as a solution to our problem involving simple intransitive clauses. (Some later proposals may allow this move, by avoiding its undesirable consequences. See footnote 4, in context).

To handle the simple intransitive clause cases, we allow that verbs having an intransitive verb prelocation type may also have this type as a located type. We can achieve this simply by stating a lexical rule of the form vp $\Rightarrow$ vp, which does not alter the form or meaning of its input, but which assigns its 'result' to the compartment for located verb types.

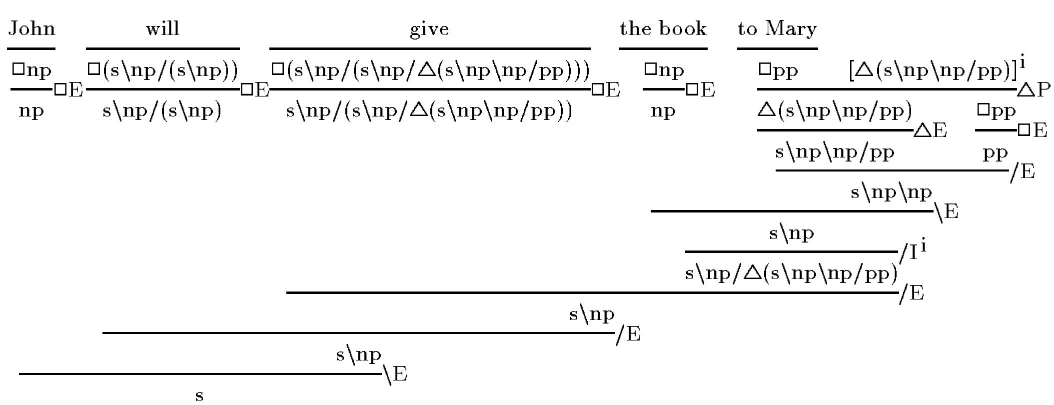
We could in this fashion allow located status for all prelocation verb types which already specify an appropriate constituent order for English (i.e. where the only argument preceding the verb is the subject), allowing simpler proofs than are needed when using the raised verb type. To admit only types specifying an appropriate complement order, we can state the rule as vpϕ $\Rightarrow$ vpϕ, where ϕ does not stand in place of any leftward directional arguments.

Some located verb types are shown in (4.16). Some of these result from the operation of the Verb Location rule, whereas others are identical to the verb's prelocation type (being admitted because they already specify the appropriate complement order for English).

(4.16)	run:	$s\backslash np$
	eat:	$(s\backslash np)/((s\backslash np)/\triangle(s\backslash np\backslash np))$
	give:	$(s\backslash np)/((s\backslash np)/\triangle(s\backslash np\backslash np/pp))$
	will:	$s\backslash np/(s\backslash np)$
	want:	$s\backslash np/\Box(s\backslash np)$
	believe:	$s\backslash np/\Box sp$
	tell:	$(s\backslash np)/((s\backslash np)/\triangle(s\backslash np\backslash np/\Box sp))$
	persuade:	$(s\backslash np)/((s\backslash np)/\triangle(s\backslash np\backslash np/\Box(s\backslash np)))$
	promise:	$(s\backslash np)/((s\backslash np)/\triangle(s\backslash np/\Box(s\backslash np)/np))$

The derivation in (4.17) illustrates this view of word order:

(4.17)



The head location process is essentially lexical in its nature. However, there is a clearly a parallel between this account and others which claim a disparity between the 'base structure' or in some sense 'underlying' position of a head, and its observed position. We shall return to discuss this relation later.

4.3.5 Lexical structure and primitive subcategorization

In work within a number of approaches, it has been argued that the explanation of certain phenomena in the syntactic domain requires access to some form of 'lexical structure', for example subcategorization frames or some kind of semantic structure. In the present account, treatment of various phenomena may depend on access to a kind of 'lexical structure', namely the function-argument structure of prelocation types, which encodes relative obliqueness. Since the prelocation type of a head appears as a subpart of its ultimate (located) type, the information that it encodes would seem to be available in the syntactic domain. A problem arises in this regard which requires that we must extend our framework to ensure the availability to syntax of the obliqueness information encoded in lexical types, and in a way that does not compromise the monostratality of the framework.

As we have seen, the Lambek calculus allows great flexibility in the assignment of types to strings. For example, the type transition in (4.18a), in which two counterdirectional arguments of a function are commuted, is valid. As another example, note that there is, in general, a valid transition from the type assigned to any verb under our account of word order to the type that it would more commonly be assigned under concatenative CGs. An example is shown in (4.18b) for a type of *give*.

(4.18) a. $\mathrm{x\backslash y/z} \Rightarrow \mathrm{x/z\backslash y}$

b. $\Box(\mathrm{s\backslash np/(s\backslash np/}\triangle\mathrm{(s\backslash np\backslash np/pp)))} \Rightarrow \mathrm{s\backslash np/pp/np}$

This suggests a problem that might arise for *syntactic* treatments of phenomena that depend on grammatical hierarchy, namely that because the syntactic calculus allows for the argument structure of functions to be changed, we cannot be sure that the argument order specified on some verb type corresponds to the argument order given for the verb lexically. For example, the transition in (4.18b) produces a type for *give* in which argument order does not correspond to obliqueness.

Our solution to this problem involves adopting two additional connectives, $\not/$ and $\not\backslash$, which have the following elimination rules (identical to those for / and \), but *no introduction rules:*

(4.19)
$$\frac{\begin{matrix}\vdots & \vdots\\ \mathrm{A}\not/\mathrm{B} & \mathrm{B}\end{matrix}}{\mathrm{A}}\not/\mathrm{E} \qquad\qquad \frac{\begin{matrix}\vdots & \vdots\\ \mathrm{B} & \mathrm{A}\not\backslash\mathrm{B}\end{matrix}}{\mathrm{A}}\not\backslash\mathrm{E}$$

The absence of introduction rules means that any occurrences of these connectives must originate lexically. Thus, whilst it is possible to 'convert' a ${/\!\!\!|}$ into a $/$, as in (4.20a), the converse, shown in (4.20b), is not possible. Note that the use of these connectives will not undermine the general flexibility of the approach since we will be as free to derive types involving the connectives $/$ and $\backslash$ as before. Thus, whilst the type transition in (4.20c) is not possible, that in (4.20d) is.

(4.20) a. $\quad x{/\!\!\!|}y \Rightarrow x/y$

b. * $x/y \Rightarrow x{/\!\!\!|}y$

c. * $x{\backslash\!\!\!|}y{/\!\!\!|}z \Rightarrow x{/\!\!\!|}z{\backslash\!\!\!|}y$

d. $\quad x{\backslash\!\!\!|}y{/\!\!\!|}z \Rightarrow x/z\backslash y$

We use these connectives to specify what might be called *primitive subcategorization*, i.e. subcategorization that originates with prelocation head types, whose argument order encodes obliqueness, serving to distinguish it from functional structure that results from syntactic transitions. In this system, any functor type of the form $A{/\!\!\!|}B$ or $A{\backslash\!\!\!|}B$ is one whose argument order must be lexically given, and where the argument B must be less oblique than any complement with which the functor has already combined.

Using these connectives to indicate primitive subcategorization, a verb such as *give* might be given a prelocation type such as (4.21a) which would give rise to the final (located) lexical type assignment shown in (4.21b).

(4.21) a. $\quad s{\backslash\!\!\!|}np{\backslash\!\!\!|}np{/\!\!\!|}pp$

b. $\quad \Box(s{\backslash\!\!\!|}np/(s{\backslash\!\!\!|}np/\triangle(s{\backslash\!\!\!|}np{\backslash\!\!\!|}np{/\!\!\!|}pp)))$

Note that with the adoption of these two new connectives, we now take vp to abbreviate types of the form $s{\backslash\!\!\!|}A$.

This apparatus should allow us to avoid the problems noted above for syntactic treatments of phenomena that depend on obliqueness, i.e. any rule or type involved in such a treatment can make reference to functional types having connective ${/\!\!\!|}$ or ${\backslash\!\!\!|}$, ensuring that the argument structure of the relevant type is lexically given.[4]

[4] We noted in the previous section that removing the restriction against allowing types to be assigned to the empty string, although potentially useful, had undesirable consequences. These problems might be avoided by using the primitive subcategorizations slashes. If we allow independent proofs having no undischarged assumptions, we find that type transitions of the form $X/(Y/Y) \Rightarrow X$ are possible, but ones of the form $X/(Y{/\!\!\!|}Y) \Rightarrow X$ are not. If the noun premodifier type is taken to be $(n{/\!\!\!|}n)$, and the adnominal intensifier *very* $(n{/\!\!\!|}n)/(n{/\!\!\!|}n)$, the problem noted earlier does not arise whereby *very* is also allowed the adnominal type, since the type transition $(n{/\!\!\!|}n)/(n{/\!\!\!|}n) \Rightarrow (n{/\!\!\!|}n)$ is not allowed.

4.3.6 The structure of English noun phrases

We have so far only considered head location as it applies to verbs. We look next at the structure of English noun phrases, and consider whether head location is required in their treatment.

An initial question concerns the functional relationship that holds between nouns and their specifiers, i.e. determiners and possessive NPs. It is standard in categorial work to assume that determiners are functions over nouns, i.e. receiving a type np/n. In constrast, we assume that nouns are functions from specifier types to np (in line, for example, with the position taken in HPSG (Pollard and Sag, 1987)) having types such as:

$$\mathrm{np}\backslash\!\!\!\circ\,\mathrm{det} \qquad \mathrm{np}\backslash\!\!\!\circ\,\mathrm{np}_p$$

(where np_p abbreviates the type np[+possessive], the type of possessive NPs). This view of NP structure more directly recognizes the role of the noun as the *head* of the NP.

Evidence implicating a role for head location in the constituent order of English NPs may be adduced in relation to multiple complement nouns that seek more than one postnominal complement. Consider the clause in (4.22a) involving the verb *to speak*. We assume that both PPs are subcategorized complements, and that the linearly first PP is the less oblique. Given these assumptions, head location must apply to the verb to give rise to the observed constituent order. Consider next the NP in (4.22b) involving the noun *speech.* By analogy to the case in (4.22a), we might expect that the two PPs are arguments, with the linearly first being the less oblique.[5] Hence, head location is also required to handle the observed order for this NP example.

(4.22) a. John spoke to the committee about the budget.

b. A speech to the committee about the budget.

We adopt the hypothesis that English NP structure is very similar to English clause structure. Firstly, we assume that the same ordering principles apply for noun complements as for verb complements, with the noun's specifier being treated as a subject, a first PP argument as a first object, and so on. Secondly, we assume that head location applies for (at least) those cases where a (prelocation) noun seeks more than one argument to its left. Head location for nouns is effected by the following rule, which is essentially the same as that for verb location:

(4.23) a. English Noun Location rule:

$$\overline{\mathrm{N}}\phi \Rightarrow \overline{\mathrm{N}}/(\overline{\mathrm{N}}/\triangle(\overline{\mathrm{N}}\phi))$$

b. $\lambda f \lambda g.(gf)$

[5] Evidence for this relative obliqueness may be adduced from the reflexivization possibilities between the two PP complements, as we will discuss in the next chapter.

In this rule, the symbol $\overline{\mathrm{N}}$ stands for either np⦸det or np⦸np_p, i.e. schematizing over the cases of a noun subcategorizing for its specifier. It would not be difficult to state a single rule to encompass both verb and noun location, although shall not do so here. Some prelocation and located noun types are as follows:

(4.24)

	prelocation	*located*
book:	np⦸det	np⦸det
picture:	np⦸det⦸pp	np⦸det/(np⦸det/△(np⦸det⦸pp))
speech:	np⦸det⦸pp⊘pp	np⦸det/(np⦸det/△(np⦸det⦸pp⊘pp))

4.4 Comparisons with other approaches

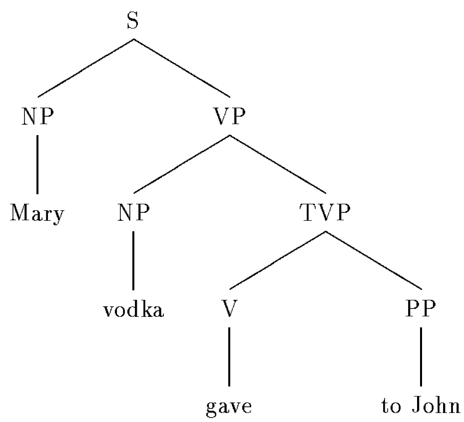
In this section, we draw comparisons between the present approach and others that bear some relation to it. The head location process is essentially lexical in its nature. However, there is a clearly a parallel between this account and others which claim a disparity between the 'base structure' or in some sense 'underlying' position of a head, and its observed position. We consider two examples of accounts which take such a view of English word order, the accounts of Jacobson (1987) and Koster (1988). Then we discuss proposals by Sag (1987) for an account which links constituent order to obliqueness.

4.4.1 Jacobson's account of English

The account of Jacobson (1987) is of particular interest here. Although set within a rather different framework, Jacobson's account has been an important inspiration for the Lambek account proposed above. Jacobson's grammatical framework combines ideas from Montague Grammar and GPSG. In this approach, syntactic types may be functional, as in CG, but the rules for combining types are phrase structure rules, so that analyses take the form of phrase structure trees that have nodes labelled with (possibly functional) types. Jacobson uses VP and TVP as phrasal node labels which abbreviate functional types of the schematic forms s/Arg and s/Arg/Arg respectively (where *Arg* schematises over possible types of subject and object), and uses V as a node label abbreviating lexical verb types. Following GPSG, the ordering of daughters within phrase structure rules is determined by linear precedence rules (types being formed using a directionless connective, as in Montague Grammar). However, phrase structure rules are not derived from Immediate Dominance rules as in GPSG, but rather from highly schematic combination rules, provided in Universal Grammar (UG). Such rules are, in the simplest case, closely analogous to (non-directional) application rules of CG. For example, UG will include a schematic rule for combining subjects and verb phrases. Any language may choose particular options for the language particular instantiation of the rule, for example options of agreement and case marking, and language particular LP statements will determine the relative order in which subject and VP appear.

Were it not that other factors come into play, Jacobson's rules for English would give rise to analyses for strings of English words in which the relative proximity of a verb to its complements correlates with their relative obliqueness. For example, these rules would allow the analysis (4.25) for the string **Mary vodka gave to John.*

(4.25)



The phrase structure rules used in this example closely correspond to functional application, each combination being binary and (if we look beyond the abbreviatory labels) of one the two forms:

$$A \rightarrow A/B \quad B$$

$$A \rightarrow B \quad A/B$$

We can see that, in terms of its basic apparatus, Jacobson's approach embodies a version of the 'concatenative assumption'. Clearly then, without some additional complication of the account, it will suffer precisely the problem we have noted for concatenative CGs regarding Dowty's theory of GRs, namely predicting a relation between relative obliqueness and proximity to the verb, which is not borne out in English word order. The correct word order for such examples is derived by invoking a marked option for combination, termed *Verb Promotion*, involving extraction of the verb.

Jacobson's system permits more complicated realizations of the schematic combination rules, which allow for cases involving extraction. Extraction is handled using gap-passing features, in the style of GPSG. Two distinct gap-passing features are used [SL] (for 'slash') and [DSL] (for 'double slash'), which differ in that the latter allows only for *bounded* extraction (although we shall not consider here how boundedness is enforced). There will be variants of combination rules which allow for filler introduction. Thus, we may have a rule of the form (4.26a) as a variant of a more basic rule of the form (4.26b). The transmission of such slash information proceeds under feature inheritance principles, as in GPSG.

(4.26) a. A $\rightarrow$ B C[SL:D] D

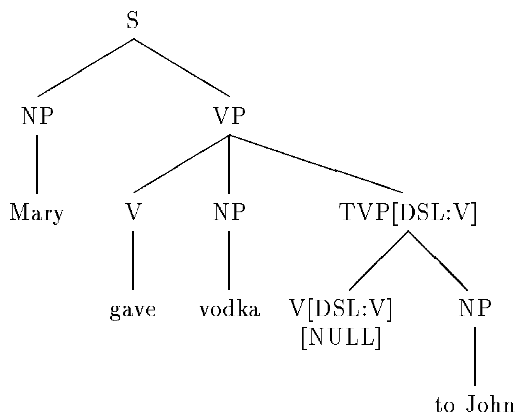
b. A $\rightarrow$ B C

According to Jacobson's marked option of Verb Promotion for English, verbs are extracted from TVPs to appear as a daughter of VP. Thus, the grammar of English includes the following rule:

(4.27) VP $\rightarrow$ V_α NP TVP[DSL:V_α]

(where the α subscript indicates sharing of features). This allows the following derivation, which can be seen as the Verb Promotion variant of the analysis above:

(4.28)



Note that Jacobson's choice of bounding nodes (NP, $\overline{\text{VP}}$ and $\overline{\text{S}}$) excludes (ungrammatical) sentences in which a verb is promoted to appear under some VP other than that which immediately dominates the TVP from which the verb is extracted.

Verb Promotion can be seen as Jacobson's reconstruction of the Montagovian wrap mechanism within a context-free phrase structure approach. Jacobson takes promotion to be a marked option of language, with the grammars of individual languages containing stipulations as to promotions that may or must occur. The grammar of English includes a convention that verbs must be promoted from TVPs.

Jacobson motivates this account in relation to the phenomenon of particle-shift in English and in particular, some problems that arise for handling the particle-shift behaviour observed with double object constructions. However, we delay presenting Jacobson's account of particle shift until later in the chapter, where we show that Jacobson's treatment can be readily 'reconstructed' within our Lambek treatment of word order.

4.4.2 Koster's account of English

Koster (1988) suggests an account of English which assumes that the canonical word order of English involves verb movement. Despite the fact that Koster's framework, a version of GB (Chomsky, 1981), is radically different from Jacobson's framework and our own, we shall see that the primary motivations for Koster's proposals are essentially the same.

Koster argues that there is a difference between the D-structure and S-structure word orders of English, particularly with regard to the position of the verb. He proposes the following D-structure constituent order for English VPs:

(4.29) $\ldots$ NP NP V Prt PRED PP S$'\ldots$

(This is a *maximal* expansion, showing all the positions that may be filled in any construction. Koster uses PRED to indicate the position of NPs PPs and APs with predicative function, assuming that predicative NPs are realized in a position distinct from direct and indirect object NPs.) This order constrasts with the S-structure constituent order which Koster characterizes as follows:

(4.30) $\ldots$ V NP Prt NP PRED PP S$'\ldots$

Koster argues that the change in order is principally the consequence of an *obligatory* verb movement (to SPEC of VP) as follows:

(4.31) $\ldots$ NP NP V$\ldots$ $\Longrightarrow$ $\ldots V_i$ NP NP $[_{\mathrm{v}}\ t_i]\ldots$

(Clearly, there are further dissimilarities between the orders (4.30) and (4.29), but we shall not consider Koster's account for them.)

Koster explains this obligatory movement in terms of the parameter 'direction of government', which he claims can have different values at D-structure and S-structure. Koster proposes the following hypothesis concerning the direction of government by English verbs:

(4.32) The English V governs NPs to the left in its D-structure position and to the right in its S-structure position (Koster, 1988, p5)

Koster links this hypothesis to the historical origins of English as an SOV language. According to Lightfoot (1979), Old English was a predominantly SOV language although exhibiting considerable verb movement. A transition to consistent SVO order then occurred, during the period of early Middle English. Koster suggests that the transition is as yet incomplete, and that NPs are most resistant to the change so that, although most complements are governed to the right, NPs are governed to the left at D-structure and to the right at S-structure. Obligatory verb movement follows as the only way to satisfy the condition imposed by Koster's hypothesis (4.32).

One point to be raised in regard to Koster's account is that he makes no claims as to differing roles that government fulfils at D-structure and S-structure which make the direction of government parameter relevant at both these levels. He does not, for example, claim that government at D-structure is relevant for θ–assignment, and government at S-structure for case assignment (as one might expect). This omission leaves the function fulfilled by verb movement obscure.

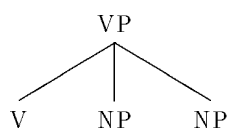
We shall consider how Koster motivates this view of English constituent structure. Koster considers some problems discussed by Barss and Lasnik (1986) that arise for the treatment of reflexivization with English double object constructions. The following examples illustrate the observed behaviour for binding between the two objects:

(4.33) a. John showed Mary$_i$ herself$_i$

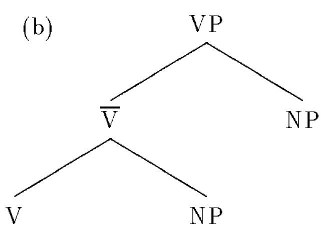
b. *John showed herself$_i$ Mary$_i$

We see that the linearly first object can bind the second but not vice versa. According to the standard GB view, reflexivization is only possible if the binder c-commands the reflexive. However, as Barss and Lasnik point out, this view fails to predict the observed behaviour under any standard view of the structure of the VP with double object constructions. The two most plausible candidates are a flat VP structure, as in (4.34a), or a binary branching structure as in (4.34b).

(4.34)(a)

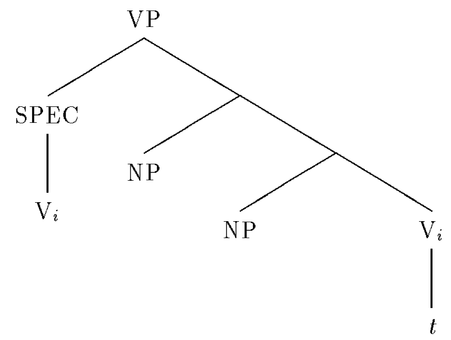


(b)



The structure in (4.34a) leads to the incorrect prediction that binding between the objects should be possible in either direction since the two objects c-command each other. The structure in (4.34b), where the second object c-commands the first, is even worse as it predicts precisely the opposite of the observed behaviour. In the structure given by Koster's account, the first object c-commands the second, correctly predicting the binding behaviour:

(4.35)



Koster also motivates his account in relation to a notion *cohesion*, which he loosely refers to as 'semantic closeness'. Koster assumes, for example, that prepositional objects have greater cohesion to the verb than adjuncts, and that particles are in this sense very 'close' to the verb. In addition, Koster assumes that for idioms, idiomatic 'arguments' are also 'close' to the verb. Koster argues that, in general, word order is characterized by a 'cohesion principle', namely that the more cohesion there is between a constituent and the verb, the closer it is to the verb in unmarked word order. This results from there being a correspondence at D-structure between cohesion and proximity to the verb. He contrasts English with Dutch in this regard: Dutch word order seems to be well behaved with respect to this principle, whereas English systematically deviates from it.

It is clear that Koster's notion of cohesion is essentially the same as that of obliqueness as it is characterized in the Montagovian approach. Thus, the 'semantic closeness' of a highly cohesive complement of the verb corresponds in the Montagovian approach to it occurring earlier in the argument order of the verb. Koster's cohesion principle can be seen to correspond to the prediction discussed earlier that arises for a concatenative CG that attempts to adopt a version of Dowty's theory. Clearly, the versions of $\overline{\mathrm{X}}$–theory that are usually assumed in GB embody a variant of the the concatenative assumption. The correspondence of the two views is even closer if some version of Kayne's (1981) *binary branching hypothesis* is adopted, as Koster indeed does.

4.4.3 Sag's account of complement order

We next consider some proposals concerning the treatment of word order in the HPSG framework described in Sag (1987) and Pollard and Sag (1987). This treatment is of interest here because it links word order to obliqueness.

In HPSG, the subcategorization requirement of words and phrases is explicitly stated by means of a stack-valued subcategorization feature, where order on the stack is taken to encode obliqueness. Constituents are combined using 'phrase structure rules' (which access the subcategorization information of the head element), and the linear order of the daughters of any projection is determined in terms of linear precedence statements. Crucially, HPSG's phrase structure rules for English give a flat structure to the VP. Sag suggests an extension of the linear precedence approach which allows ordering statements whose effect depends on the relative obliqueness of the elements related. For example, the statement $A \ll B$ (simplifying somewhat) requires that any daughter A (i.e. that bear the features A, or can unify with A) linearly precede any daughter B just in case A is a less oblique complement of the head than B. Sag proposes the following rule for English word order, which requires that non-verbal complements must precede more oblique non-lexical complements:

$$\text{COMPLEMENT}[\text{MAJ } \neg\text{V}] \ll \text{COMPLEMENT}[\text{LEX}-]$$

For our account, assignment of directionality to arguments of some function can be seen to play a similar role in specifying possible orderings. Thus, if A is an argument of a functor F, assigning leftward directionality for A corresponds to requiring that A must precede all more oblique dependents of the same head, whereas giving it rightward directionality requires that it follow all more oblique dependents of the same head. Note the difference to Sag's system that the directionality of A bears only on the ordering of the specific argument A relative to more oblique dependents, and does so irrespective of the other dependents' types. In contrast, Sag's rule is a general statement about ordering, and its applicability is conditioned by the categories of the phrases involved.

Our account, Koster's account and Jacobson's account share the characteristic of linking word order to obliqueness under some version of the 'concatenative assumption'. Consequently, each account is forced, in one way or another, to dissociate the specification of verb position from the specification of complement order. For Jacobson's and Koster's accounts, this is achieved by some version of extraction, and for our account by a lexical process of head location. The HPSG account avoids the problems that arise with the concatenative assumption by having a flat verb phrase and handling word order purely in terms of linear precedence. Interestingly, however, the HPSG account recognizes the need to invoke obliqueness to handle English word order, and Sag's proposals allow that obliqueness can be accessed directly by the mechanisms that determine word order.

4.5 Lexical treatments of relation changing phenomena

Our new treatment of word order should provide a basis for the treatment of a range of phenomena that have been argued to depend on grammatical relations and obliqueness, given its incorporation of a modified version of Dowty's theory. Such treatments can be subdivided as to whether they are lexically or syntactically based. We shall not give an illustration of a syntactic treatment in this chapter. A good candidate for an obliqueness dependent phenomenon that requires a syntactic treatment is binding, and that topic is considered in detail in the next chapter. In this section, we consider some lexical treatments of relation changing phenomena. The Montagovian approach allows that relation changing phenomena can be handled by what Dowty calls 'category changing rules', i.e. rules which modify the functional types of verbal constituents.

4.5.1 Passive

Bach (1980) discusses an account of Passive which is based on the operation of a category-changing rule. The rule for agentless passive can be stated as follows:

(4.36) a. $\mathrm{IV} \Rightarrow \mathrm{TV}$

b. $\lambda \mathrm{f}.\lambda \mathrm{x}.\exists \mathrm{y}.[\mathrm{f\,x\,y}]$

The syntactic rule in effect 'deletes' an argument of the function (i.e. since TV abbreviates t/T/T and IV abbreviates t/T). The semantics of the rule is stated in (4.36b) (in somewhat simplified form) as a lambda expression which applies to the semantics of the input TV, abstracting over its object argument and existentially quantifying over its subject argument. Since the output type of the rule is IV, its remaining argument is treated as a subject, even though it corresponds semantically to the object of the input verb. Passivization is viewed as a syntactic process, so that a ditransitive verb would need to combine with its second object before the passive rule could apply. The rule is associated with a string modifying operation that changes the form of the (lexical) verb within the string to have passive morphology (obviously a non-concatenative string operation).

We can adopt a modified version of this account which is based on (schematic) lexical rules. It is convenient to base the account in the lexicon since it is much easier to state the rule as applying to prelocation types (where the only functional structure that needs to be addressed is that which encodes obliqueness) and since this move avoids various problems for handling passive morphology in syntax. (4.37) shows the agentless passive rule, which takes prelocation verb types as input and yields further prelocation types as output.

(4.37) a. Agentless Passive Rule (lexical):

$$(\mathrm{s}\between\mathrm{np}\between\mathrm{np})\phi^n \Rightarrow (\mathrm{s}\between\mathrm{np})\phi^n$$

b. $\lambda \mathrm{f}.\lambda \mathrm{x_1} \mathinner{..} \mathrm{x_n}.\lambda \mathrm{y}.\exists \mathrm{z}.[\mathrm{f\,x_1} \mathinner{..} \mathrm{x}_n\,\mathrm{y\,z}]$

The notation $\mathrm{X}\phi^n$ is schematic for functions into type X, and in particular functions that require n arguments to give X. (Thus, X is of the form $\mathrm{X}\phi^0$, and X/A/B is of the form $\mathrm{X}\phi^2$.)

Furthermore, the occurrence of ϕ^n on both sides of $\Rightarrow$ in (4.37) indicates that the input and output types of the rule require identical sequences of arguments to map them into their respective values. The rule (4.37) maps from types that are functions into (prelocation) transitive verb types to types that are functions requiring the same arguments to give an intransitive verb type. The semantics for the rule, shown in (4.37b) as a lambda expression that applies to the meaning of the input verb, is also schematic. This abstracts over the non-subject arguments of the verb, and existentially quantifies over its subject argument. This semantics for the rule is such that the final NP argument of the output verb type, though a subject by virtue of being the least oblique complement of the verb, actually corresponds semantically to the object of the input verb type. We have omitted some details in the statement of the rule above. For example, the application of the rule also marks the type as [+passive] (such features being marked on the 'innermost' value subtype s), and would be accompanied by an operation changing the lexical phonology of the verb to the appropriate form. This lexical treatment of passive is reminiscent of that suggested for HPSG in Pollard & Sag (1987).

Example (4.38) shows a prelocation type for *give* and the type for the passive participle *given* derived from this by the rule (4.37). Below these are shown the final lexical types that would result after head location and other lexical processes.

(4.38) *give* $\Rightarrow$ *given*

$$\mathrm{s}\obslash\mathrm{np}\obslash\mathrm{np}\oslash\mathrm{pp} \;\Rightarrow\; \mathrm{s}\obslash\mathrm{np}\oslash\mathrm{pp}$$

$$\mathrm{give}' \qquad \lambda \mathrm{x}.\lambda \mathrm{y}.\exists \mathrm{z}.[\mathrm{give}'\,\mathrm{x}\,\mathrm{y}\,\mathrm{z}]$$

$$\Box(\mathrm{s}\obslash\mathrm{np}/(\mathrm{s}\obslash\mathrm{np}/\triangle(\mathrm{s}\obslash\mathrm{np}\obslash\mathrm{np}\oslash\mathrm{pp}))) \qquad \Box(\mathrm{s}\obslash\mathrm{np}/(\mathrm{s}\obslash\mathrm{np}/\triangle(\mathrm{s}\obslash\mathrm{np}\oslash\mathrm{pp})))$$

We can handle Agentive Passive using the lexical rule (4.39). Once again, this rule maps from prelocation types to further prelocation types, and not only rewrites the transitive verb subtype as a intransitive verb subtype, it also adds a PP *by*-phrase to the verb's subcategorization as the latter's most oblique complement. The semantics of the rule are such that the *by*-phrase corresponds semantically to the subject of the input type. Again, this treatment is very much in line with that of HPSG.

(4.39) a. Agentive Passive Rule (lexical):

$$(\mathrm{s}\obslash\mathrm{np}\obslash\mathrm{np})\phi^n \Rightarrow (\mathrm{s}\obslash\mathrm{np})\phi^n\oslash\mathrm{pp}_{by}$$

b. $\lambda \mathrm{f}.\lambda \mathrm{z}.\lambda \mathrm{x}_1 \ldots \mathrm{x}_\mathrm{n}.\lambda \mathrm{y}.[\mathrm{f}\,\mathrm{x}_1 \ldots \mathrm{x}_n\,\mathrm{y}\,\mathrm{z}]$

4.5.2 Dative Shift

As a second example, we consider the case of Dative Shift. Dowty (1982a) presents an account based on a lexical type changing rule. It seems appropriate that an account of

Dative Shift should be lexical, given that it is clearly lexically conditioned in English and other languages. Adapting Dowty's account, we can treat Dative Shift using the lexical rule in (4.40), which, again, creates additional prelocation verb types.[6]

(4.40) a. Dative Shift Rule (lexical):

$$(\mathrm{vp}\backslash\!\!\!\circ\,\mathrm{np}\not\circ\,\mathrm{pp})\phi^n \Rightarrow (\mathrm{vp}\backslash\!\!\!\circ\,\mathrm{np}\not\circ\,\mathrm{np})\phi^n$$

b. $\lambda \mathrm{f}.\lambda \mathrm{x}_1 \ldots \mathrm{x}_\mathrm{n}.\lambda \mathrm{y}.\lambda \mathrm{z}.[\mathrm{f}\,\mathrm{x}_1 \ldots \mathrm{x}_\mathrm{n}\,\mathrm{z}\,\mathrm{y}]$

The semantics of the rule abstracts over the two object arguments so as to reverse the semantic order of these. Thus, the first object of the output type corresponds semantically to the second object of the input type and vice versa.

The use of the Dative Shift rule is illustrated in (4.41) which shows a prelocation type for *give* and the additional type that results from the operation of the rule. Also shown are the final lexical types that the prelocation types give rise to.

(4.41)

give		*give*
$\mathrm{s}\backslash\!\!\!\circ\,\mathrm{np}\backslash\!\!\!\circ\,\mathrm{np}\not\circ\,\mathrm{pp}$	$\Rightarrow$	$\mathrm{s}\backslash\!\!\!\circ\,\mathrm{np}\backslash\!\!\!\circ\,\mathrm{np}\not\circ\,\mathrm{np}$
give′		$\lambda \mathrm{y}.\lambda \mathrm{z}.[\mathrm{give}'\,\mathrm{z}\,\mathrm{y}]$
$\Box(\mathrm{s}\backslash\!\!\!\circ\,\mathrm{np}/(\mathrm{s}\backslash\!\!\!\circ\,\mathrm{np}/\triangle(\mathrm{s}\backslash\!\!\!\circ\,\mathrm{np}\backslash\!\!\!\circ\,\mathrm{np}\not\circ\,\mathrm{pp})))$		$\Box(\mathrm{s}\backslash\!\!\!\circ\,\mathrm{np}/(\mathrm{s}\backslash\!\!\!\circ\,\mathrm{np}/\triangle(\mathrm{s}\backslash\!\!\!\circ\,\mathrm{np}\backslash\!\!\!\circ\,\mathrm{np}\not\circ\,\mathrm{np})))$

4.6 Particle movement in English

In this section we consider the treatment of particle movement in English, which presents a number of problems. The treatment we give is essentially a reconstruction of the account of Jacobson (1987). We begin by considering Jacobson's account.

4.6.1 Jacobson's account of particle movement

Jacobson motivates her view of word order (discussed above) in relation to the phenomenon of particle-shift in English and in particular, the particle-shift behaviour observed with double object constructions, exemplified by the following data, from Emonds (1976):

(4.42) a. The secretary sent out the stockholders a schedule.

b. The secretary sent the stockholders out a schedule.

c. *The secretary sent the stockholders a schedule out.

d. The stockholders were sent out a schedule.

e. *The stockholders were sent a schedule out.

[6] The rule is again schematic, i.e. allowing for further arguments, although the need for this may not be obvious. One case where this is required is double object verbs that take a particle, such as *sent*.

(4.43) a. I'll fix up everybody a meal.

b. I'll fix everybody up a meal.

c. *I'll fix everybody a meal up.

d. Everybody was fixed up a meal.

e. *Everybody was fixed a meal up.

(Speaker intuitions vary as to the acceptability of some of these sentences. (4.42a) is found only marginally acceptable by some. However, judgments of the unacceptability of the starred examples is robust.)

Jacobson discusses some problems that arise for the transformational analysis of particle-shift. This account invokes a Particle Movement transformation, that moves the particle, which is assumed to originate adjacent to the verb, rightwards past a NP. Limiting this movement to be over a single NP seems to account for the ungrammaticality of the examples (4.42c) and (4.43c). But then a problem arises for explaining why this movement is not allowed for the passive examples (4.42e) and (4.43e). One possible solution to this involves invoking the presence of a passive trace. Jacobson discusses some data concerning a constraint on the distribution of pronouns in English which indicates sensitivity to the presence of *wh*-trace but not passive trace, and argues that an account of the data in (4.42) and (4.43) which depends on the presence of a passive trace must explain why the presence of a passive trace affects one process and not the other.

Jacobson points out that the generalization covering the examples in (4.42) and (4.43) can be straightforwardly expressed in terms of grammatical relations as follows: a particle may follow a direct object but not the second object of a ditransitive.

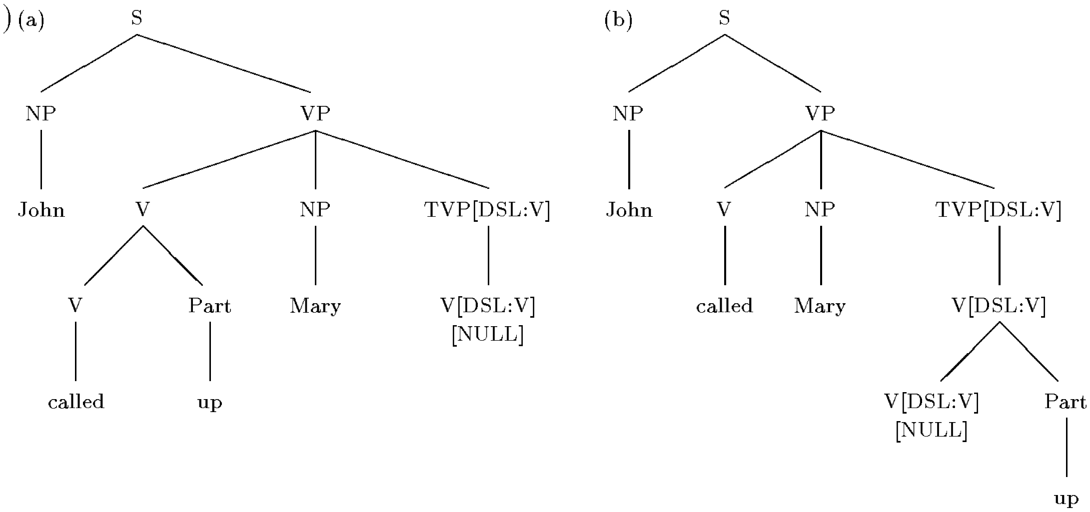
Jacobson's account of Particle Movement is based on the idea that the verb can combine with its particle to form a complex verb, as allowed by the following phrase structure rule:

(4.44) $\underset{[+P]}{\mathrm{V}} \rightarrow \underset{[-P]}{\mathrm{V}}\ \ \mathrm{Part}$

Particle movement alternations arise as a consequence of Verb Promotion, and in particular, whether the verb that undergoes promotion is simply the lexical verb, or is the verb-plus-particle complex. If the latter, then the particle appears adjacent to the verb. However, if the verb moves without its particle, then the particle appears at the position corresponding to the extraction site for the verb (i.e. immediately adjacent to this site), giving the cases where the verb and particle are separated. Since both the extraction and landing site of the verb are to the left of the second object in double object constructions, the particle always precedes the second object.

The following trees illustrate this account in relation to the verb *call up*.

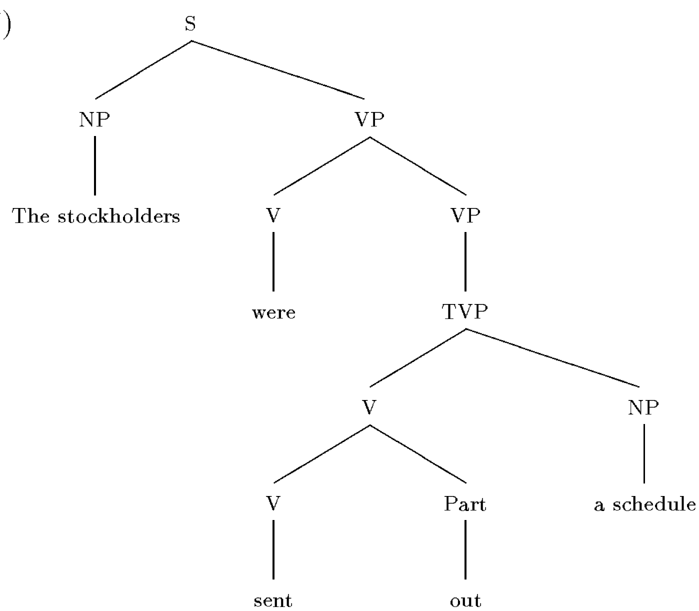
(4.45)



Jacobson suggests an treatment of (agentless) passive based on the phrase structure rule (4.46), which allows the derivation (4.47) for the example in (4.42e).

(4.46) $\underset{[+\text{Pass}]}{\text{VP}} \rightarrow \text{TVP}$

(4.47)



(Note that this structure does not involve verb promotion. This is because it does not involve the TVP+DObj rule, whose statement as a filler rule invokes the movement of the verb.) The essential structure of the TVP (i.e. ignoring movement) is the same as in the active case (with the second object following the verb-plus-particle node), and so the second object again must follow the particle, as before.

4.6.2 A Lambek account of particle movement

We next consider how Jacobon's account of particle movement can be adapted to our CG approach. We assume that particles are directly subcategorized for by prelocation verb types, always being the most oblique complement sought by a verb (in line with a standard view, assumed also by Jacobson). For example, *call* has the prelocation type:

call: s⍀np⍀np⌿part

We can read off the complement order this type would yield as:

subj < obj < particle

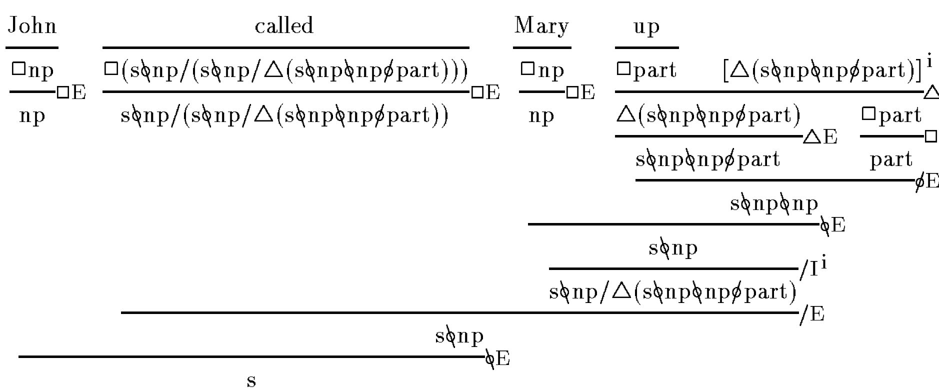
Thus, this type, ultimately, gives rise to constituent order as in *John called Mary up*.

We assume that a process of lexical incorporation is possible for verbs that take particles, applying at the level of prelocation types. For *call* this would obviously give:

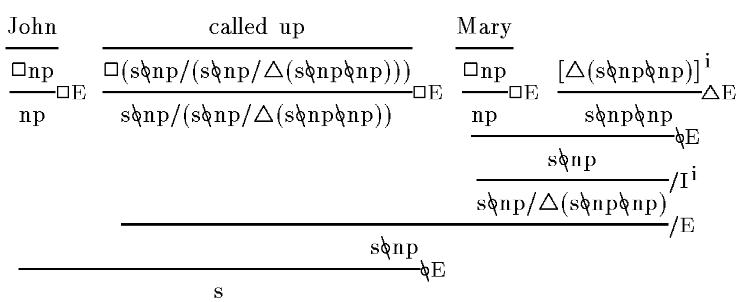
call up: s⍀np⍀np

In this case, the final (located) type for the verb gives rise to word orders in which the particle appears adjacent to the verb, as in *John called up Mary*. The following proofs illustrate the cases with and without particle incorporation:

(4.48)



(4.49)



Let us consider how this account fairs with the case of double object constructions. We assumed in (4.11) that second object (np and pp) arguments are assigned rightward directionality in prelocation verb types. The reason for doing so can be seen in relation to examples such as *The secretary sent the stockholders out a schedule*. Since the second object argument is less oblique than the particle (under the assumption that particles are always the most oblique complement) only the directionality of the second object is of relevance in determining their relative order. Since, the second object must follow the particle, it must be assigned right directionality. Of course, in the absence of more oblique complements than particles, the directionality assigned for particles themselves has no observable effect on word order, so we will simply assume rightward directionality for them.[7]

Consider (4.50), which shows active and passive prelocation types for *sent,* with and without particle incorporation. (Of course, the Dative Shift rule must also have applied to produce these types.)

(4.50)		*active*	*passive*
	sent:	s⦸np⦸np⊘np⊘part	s⦸np⊘np⊘part
	sent out:	s⦸np⦸np⊘np	s⦸np⊘np

The active type for *sent* (without particle incorporation) specifies rightward directionality for its second object. This ensures that it follows the particle, giving (ultimately) constituent order as in *The secretary sent the stockholders out a schedule.* This directionality for the second object argument is maintained when the Passive rule is applied, with the consequence that the passive type's sole object (i.e. corresponding to active type's second object) is again required to follow the particle. Thus we have constituent order as in *The stockholders were sent out a schedule.* With particle incorporation, the particle must appear adjacent to the verb in both active and passive cases. Thus, our account does not admit the ungrammatical cases in which the particle follows a second object, or its equivalent in the passive construction. With passivization, the types that arise with and without particle incorporation both require that the particle appears adjacent to the verb, allowing two alternative derivations for a sentence such as *The stockholders were sent out a schedule.*

Let us consider some points that favour this account over Jacobson's. For our account, particle incorporation really is treated as a lexical process. In Jacobson's account, the association of particle and verb takes place in syntax (even though, for reasons to do with the LP treatment of word order, the verb plus particle constituent is marked as [+Lex]).

Koster (1988) points out that there is lexical variation as to the acceptability of examples where the particle occurs immediately beside the verb, and offers the example in (4.51a) as one that is ungrammatical:

(4.51) a. *John read off Mary the figures

b. John read Mary off the figures

[7] Rightward directionality for particle arguments would seem to accord with the result of lexical incorporation, i.e. that we have *call up* and not *up call.* However, there is no particular reason why we should assume that directionality should determine order of concatenation for a *lexical* process.

Koster argues that if particle incorporation were possible, and responsible for examples where the particle appears adjacent to the verb, sentences such as (4.51a) should be acceptable in general, casting doubt on a incorporation based solution. It is not clear to the author that such examples are in fact much worse than others such as (4.42a), but we shall accept Koster's claim for the moment as a basis for discussion. For an account in which particle incorporation is treated as a lexical process, such variation is not surprising, since we can then expect lexical conditioning of the incorporation process, just as for other lexical processes like Dative Shift.

Emonds (1976) presents some interesting data, which bears on the lexical status of the particle incorporation process. There are some modifiers, such as *right,* which can modify particles (as in *right up*). The presence of the modifier affects the word order possibilities that arise. In particular, the modified particle must appear following a direct object:

(4.52) a. I looked up the answer.
b. I looked the answer up.
c. *I looked right up the answer.
d. I looked the answer right up.

This data is easily explained when particle incorporation is viewed as a lexical process. The constituent *right up* is a phrasal element, constructed in syntax, and therefore not available to a lexical process of particle incorporation. Thus, particle incorporation cannot create a lexical verb *look right up,* as would be required to admit (4.52c).

4.7 Verb Second phenomena of Germanic languages

In the remainder of the chapter we look at the Verb Second (V2) behaviour of Germanic languages, and consider its treatment under our approach to word order.

Most of the Germanic languages exhibit certain characteristics of word order that have been referred to as *Verb Second* behaviour.[8] This involves the occurrence of what appear to be distinct subtypes of clauses, which exhibit systematic differences in word order, most notably with respect to the position of the finite verb. The occurrence of these subtypes of clause is commonly identified with the distinction between subordinate and main clauses.

[8] It is commonly assumed that English is an exceptional Germanic language in this regard, not showing V2 behaviour. However, there are various constructions of English that suggest the involvement of a V2 constraint, so it is perhaps more appropriate to view English as exhibiting limited and less productive V2 behaviour, rather than being a non-V2 language. We shall not dwell upon this issue here.

We begin by restricting our attention to V2 behaviour in Dutch. However, the approach we present is applicable to the treatment of V2 behaviour more generally, and we shall consider the V2 behaviour of some other Germanic languages later on.

4.7.1 The V2 behaviour of Dutch

The basic facts of Dutch word order may be summarized as follows. Two major subclasses of clause arise in respect of V2 behaviour, which are distinguished in terms of certain word order facts, and whose occurrence is commonly identified with the distinction between main and subordinate clauses.[9]

In Dutch subordinate clauses, verbs generally occur in a cluster, including the finite verb, which is preceded by NP complements and (most) PP complements, and typically followed by verbal complements. Complementized sentential complements always follow the verb.

In main clauses, this same pattern generally holds but for some systematic differences, most notably in regard of the position of the *finite* verb. In main clause declaratives and constituent questions, the finite verb must occur in second position, i.e. following the first major constituent of the clause. In yes/no-questions and imperatives, the finite verb appears in first position. We refer to these three word order possibilities as verb-final, verb-second and verb-first, respectively.[10] There is considerable flexibility as to what constituent may occur in the initial position of verb-second clauses. It may be a NP or PP etc complement or adverbial drawn from either the main clause, or from a dependent clause or other verbal complement. In the case of so-called *complex fronting,* the initial constituent may even be a complex consisting of a non-finite verb plus some of its complements.

Some of these possibilities are illustrated by the clauses in (4.53). In (4.53a) we see a simple subordinate clause, and in (4.53b) a corresponding main clause yes/no question in which the finite verb occurs in first position. (4.53c) and (4.53d) show verb-second

[9] This characterization is overly simplistic, since, although main clauses are typically verb-second and subordinate clauses are typically verb-final, there exist clear cases where main clauses show verb-final word order and where subordinate clauses show verb-second word order. See, for example, Haider (1986) for discussion and data relevant to this point concerning German. However, in the remainder of this paper we shall simply follow the assumption, ignoring the problems for this.

[10] To avoid possible confusion, it is probably worth emphasizing the distinction between the use of the terms 'Verb Second' and 'verb-second'. The term verb-second identifies a subset of clauses distinguished by the occurrence of the finite verb in second position, whereas the term Verb Second is used (somewhat confusingly) to refer to the general phenomenon whereby main and subordinate clauses exhibit different word orders (i.e. verb-first, verb-second and verb-final).

clauses in which a subject and a questioned PP object occur in first position, respectively.

(4.53) a. ... omdat Jan appels aan Marie gaf
... because Jan apples to Marie gave
'... because Jan gave apples to Marie'

b. Gaf Jan appels aan Marie?
gave Jan apples to Marie
'Did Jan give apples to Marie?'

c. Jan gaf appels aan Marie
Jan gave apples to Marie
'Jan gave apples to Marie'

d. Aan wie gaf Jan appels?
to whom gave Jan apples
'To whom did Jan give apples?'

In (4.54) we see a similar set of examples, but where the finite major verb of (4.53) has been replaced by an auxiliary/main verb complex. These examples show that only the *tensed* verb's position changes in main clauses; the position of other verbs remains unchanged.

(4.54) a. ... omdat Jan appels aan Marie heeft gegeven
... because Jan apples to Marie has given
'... because Jan gave apples to Marie'

b. Heeft Jan appels aan Marie gegeven?
has Jan apples to Marie given
'Did Jan give apples to Marie?'

c. Jan heeft appels aan Marie gegeven
Jan has apples to Marie given
'Jan gave apples to Marie'

d. Aan wie heeft Jan appels gegeven?
to whom has Jan apples given
'To whom did Jan give apples?'

It is important to note that despite these marked word order differences relating to V2 behaviour, in other regards the word order of Dutch main and subordinate clauses show a clear correspondence. This correspondence is itself something that an adequate account of Dutch V2 must be able to explain.

A number of possibilities present themselves for how we might deal with these facts of Dutch word order, irrespective of grammatical framework adopted. One option is simply to provide a set of rules and category assignments etc that directly give rise to the different orders of main and subordinate clauses. Such a solution would seem unsatisfactory for precisely the reason just mentioned: failure to explain the common facts of main and subordinate word orders. Alternative strategies that have been considered involve taking either main or subordinate clause order to be in some sense 'basic', with the non-basic order being derived from the basic order in some manner.

Within the TG framework, it came to be agreed (after some debate) that the word order of subordinate clauses is basic, and that of main clauses derived by the operation of two transformations. Firstly, a transformation that moves the verb to a sentence initial position. This was originally assumed to be a root transformation, explaining the link between V2 behaviour and the main/subordinate clause distinction (though, as we have noted, there are problems for this assumed link). Secondly, an optional transformation which moves some other constituent to a position to the left of the landing site of the verb. It is the optionality of this second transformation that allows for the occurrence of both verb-first and verb-second clauses. For the case of Dutch, an important paper in settling the debate over which clause order should be viewed as basic in favour of subordinate clause order was Koster (1975). Koster showed that a number of word order facts can be straightforwardly explained on this hypothesis, but are highly problematic under the converse hypothesis. The most compelling of these arguments concern the position of particles, and we shall briefly return to consider this argument later.

The treatment of V2 phenomena presents a considerable problem for concatenative CG approaches. Before we go on to develop our account of Dutch V2, we will consider a CG account of V2 proposed by Hoeksema (1985).

4.7.2 Hoeksema's account of Dutch Verb-Second

The treatment of V2 phenomena presents a considerable problem for CG approaches. Without recourse to some surface/deep structure distinction it would seem that the only option is to provide different lexical categorizations that give rise to the different word orders of main and subordinate clauses. But then, how are we to explain the commonalities between main and subordinate clause word order? Hoeksema (1985) suggests an account which attempts to explain the interrelation of main and subordinate clause orders in terms of a schematic lexical rule which maps from subordinate clause (finite) verb types to verb types for main clauses. The rule is as follows:[11]

(4.55) Verb-Second Rule:

$$\mathrm{S}\backslash\mathrm{X}_1 \ldots \mathrm{X}_n \Rightarrow (\mathrm{E}/\mathrm{X}_n \ldots \mathrm{X}_{i+1}\ \mathrm{X}_{i-1} \ldots \mathrm{X}_1)\backslash\mathrm{X}_i$$

where E is Hoeksema's symbol for main clauses. The notation $\backslash\mathrm{X}_1 \ldots \mathrm{X}_n$ is intended to indicate a sequence of n arguments each with connective $\backslash$. The rule transforms the func-

[11] This statement of the rule appears different from Hoeksema's due to our use of different CG notations, Hoeksema using the Lambek notation. However, the rule is in essence unchanged.

tional type of the input verb, reversing the order of the input verb's arguments and switching the connective on each to be /—with the exception of a single argument that retains its backward directionality and is made the first argument that the function seeks. The output types of this rule directly give main clause word orders.

Let us consider an example. The verb *gaf* ('gave') is assumed to have a subordinate clause type s\np\np\pp allowing for subordinate clauses such (4.53a). The operation of Hoeksema's rule yields three main clause types, as shown in (4.56) (where the subscripts on the verb's arguments are there only to help the reader). These types allow for main clause analyses, as illustrated by the derivations in (4.57) and (4.58):

(4.56) gaf (gave) $s\backslash np_1\backslash np_2\backslash pp \Rightarrow E/pp/np_2\backslash np_1$
$s\backslash np_1\backslash np_2\backslash pp \Rightarrow E/pp/np_1\backslash np_2$
$s\backslash np_1\backslash np_2\backslash pp \Rightarrow E/np_2/np_1\backslash pp$

(4.57)

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{Jan}}{np} \quad \dfrac{\text{gaf}}{E/pp/np_2\backslash np_1}}{E/pp/np_2}\backslash E \quad \dfrac{\text{appels}}{np}}{E/PP}/E \quad \dfrac{\text{aan Marie}}{pp}}{E}/E$$

(4.58)

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{aan Marie}}{PP} \quad \dfrac{\text{gaf}}{E/np_2/np_1\backslash pp}}{E/np_2/np_1}\backslash E \quad \dfrac{\text{jan}}{np}}{E/np_2}/E \quad \dfrac{\text{appels}}{np}}{E}/E$$

There is clearly a sense in which Hoeksema's account can be seen to side with the TG view of V2, i.e. with main clause word order being derived from subordinate clause word order, though Hoeksema's approach allows this transfer to be made without the need of TG's intermediate step of building some deep structure clause having subordinate order.

Hoeksema's rule (4.55) does not cover all main clause possibilities. Some of these shortcomings have rather obvious solutions that are in line with the general trend of the account. For example, the account does not allow verb-first clauses. This can be remedied by including the following rule:

(4.59) Verb-First Rule (first version):
$$S\backslash X_1 \ldots X_n \Rightarrow E/X_n \ldots X_1$$

Hoeksema's rule does not allow for subordinate clause verbs which seek any of their arguments to the right, as occurs, for example, with verbs that take a verbal complement. Again this is easily remedied. The following two rules allow for verb-first and verb-second clauses where the subordinate clause finite verb takes complements to the right, and are equivalent to the previous versions just in case an input verb seeks no arguments to its right.

(4.60) Verb-First Rule (second version):

$$\mathrm{S}\backslash\mathrm{X}_1 \ldots \mathrm{X}_\mathrm{m}/\mathrm{X}_{\mathrm{m}+1} \ldots \mathrm{X}_\mathrm{n} \Rightarrow \mathrm{E}/\mathrm{X}_{\mathrm{m}+1} \ldots \mathrm{X}_\mathrm{n}/\mathrm{X}_\mathrm{m} \ldots \mathrm{X}_1$$

(4.61) Verb-Second Rule (modified version):

$$\mathrm{S}\backslash\mathrm{X}_1 \ldots \mathrm{X}_m/\mathrm{X}_{m+1} \ldots \mathrm{X}_n \Rightarrow (\mathrm{E}/\mathrm{X}_{m+1} \ldots \mathrm{X}_n\ /\mathrm{X}_m \ldots \mathrm{X}_{i+1}\ \mathrm{X}_{i-1} \ldots \mathrm{X}_1)\backslash\mathrm{X}_i$$

Even when Hoeksema's account is developed along these lines, a number of problems still remain. In particular, rules such as (4.55) and (4.61) can only allow for verb-second clauses in which the initial constituent is itself a direct argument of the finite verb. Cases where the initial constituent has been extracted from one of the verb's complements are not covered. For example, in (4.62a) the initial constituent is not an argument of the finite auxiliary, but rather an argument of the past participle main verb. In (4.62b) the initial constituent has been extracted from an embedded clause.

(4.62) a. Dit boek heeft Jan gelezen!
This book has Jan read
'This book, Jan has read!'

b. Wat zei Jan dat Marie kocht?
What said Jan that Marie bought
'What did Jan say that Marie bought?'

Such examples appear to support the TG view that the initial constituent of verb-second clauses is extracted to its position.

A second problem for Hoeksema's account is that it is unclear how we can specify the semantics for his rule (or the modified versions of it). The rule is schematic for many different non-schematic rules that would transform one functional type into another, and each of these rules would need to have its semantics stated separately. Applying Hoeksema's rule to some input verb will generally yield several different output types and the transformations yielding each of them will have different semantics.

4.7.3 A Lambek treatment of Dutch Verb-Second

We have seen that accounts of V2 behaviour have often assumed subordinate clause order to be basic, with main clause order derived from it. For the new CG approach to word order we have developed, a further possibility arises for explaining the interrelation of main and subordinate clause word orders. This is to assume that the distinct verb types required for related main and subordinate clauses derive from a common prelocation type by different rules of head location. Recall that prelocation types determine complement order. Under such a model, we will expect that main and subordinate clauses should exhibit corresponding constituent order except for the position of the head, i.e. the verb. Such a correspondence is the case for verb-first and verb-final clauses. This correspondence to an extent breaks down in verb-second clauses with respect to the relative ordering of the initial constituent. However, the position of the initial constituent in verb-second clauses receives a natural explanation if (in line with TG) we assume that it is an extracted item.

The general character of the head location process for main clause verbs is obvious. It specifies that the verb should appear at a position left adjacent to its own sentential projection. Such verb types directly give rise to verb-first clause order, and also allow for verb-second clauses by extraction of the topic constituent.

A less obvious issue is what should be the character of the head location process giving rise to subordinate clause order. We will argue that the head location option selected for verbs of Dutch other than main clause finite verbs is the *null option,* i.e. that the prelocation types are extended located status without needing to undergo a type-changing head location rule. This claim is supported by a number of observations.

Firstly, arguments for this claim can be adduced from observations about particles in Dutch. We noted that Koster (1975) was a significant paper in establishing the view that main clause word order was derived from subordinate clause word order in the TG framework. Some of the most compelling of Koster's arguments concern the distribution of particles. In the following examples, the particles are in italics:

(4.63) a. Hij gaf zijn vader het geld *terug*
he gave his father the money back
'He gave the money back to his father'

b. ... omdat hij zijn vader het geld *terug*gaf
because he his father the money back gave
'... because he gave the money back to his father'

Observe that the particle appears following all of the verb's other complements in both main and subordinate clauses. Koster demonstrates that the distribution of particles in main clauses corresponds to the distribution of their associated verb in subordinate clauses. Under standard TG assumptions, it is expected that particle and verb originate at the same position (i.e. are adjacent in deep-structure). Koster argues that the location of particles in main clauses can be straightforwardly explained in an account in which verb and particle are base-generated in clause final position and then the verb is fronted, whereas an account in which verb and particle are base-generated in second or initial position and then the particle moved down to clause final position would be highly problematic, since the rule of 'Particle Movement' would be impossible to state without a host of *ad hoc* stipulations.

For the present account, Koster's arguments can be taken on board with little substantial alteration. We have assumed, in line with a fairly 'traditional' view, that particles are the most oblique complement sought by prelocation verb types. The claim of a null head location process leads to a prediction that a verb should appear adjacent to its particle in subordinate clause word order, and that the distribution of particles in main clauses should correspond to the subordinate clause distribution of the verbs that introduce them. Both of these predictions are borne out, as Koster has demonstrated.

Secondly, the hypothesis that a null head location process applies for Dutch subordinate clause verbs leads to the prediction noted earlier that arises for any concatenative approach that adopts a version of Dowty's theory of GRs, i.e. relative obliqueness should be mapped out in terms of relative proximity to the head, such that for complements which appear to the same side of the head, greater proximity to the verb indicates increasing obliqueness. For Dutch double objects constructions, where we observe constituent orders such as:

NP NP NP V

NP NP PP V

we expect that relative obliqueness should pattern with linear order: the linearly first NP (i.e. the subject) being the least oblique, and the linearly first object less oblique than the second. If binding behaviour is taken as an indicator of obliqueness (see Chapter 5), this prediction is borne out:

(4.64) a. ... dat ik Jan$_i$ zichself$_i$ toonde
'... that I Jan himself showed'

b. *... dat ik zichself$_i$ Jan$_i$ toonde
'... that I himself Jan showed'

The account of V2 we propose is different from that of TG in that the common facts about main and subordinate clause word order are explained not by deriving main clause order from subordinate clause order, but by positing the existence of a common point of origin for the verb types that give rise to them, i.e. prelocation types. However, since we claim for Dutch that subordinate clause verb types are derived under the null head location option, it might be argued that, under an intuitive characterization, this position is essentially the same as that of TG, i.e. with main clause order derived from subordinate clause order. The difference between the two positions is perhaps not obvious for Dutch, but, as we will see, for other Germanic languages the difference is clearer. For Scandinavian languages, both main and subordinate clause verbs have non-null head location processes. In this case, whereas TG still argues for the priority of subordinate clause order in determining main clause constituent order, our CG approach gives priority to prelocation types that specify an order distinct from that of either main or subordinate clauses.

Distinguishing clause types

Since the different clause types exhibit different distributions, we must featurally distinguish them, so that subcategorizing elements can specify requirements on the form of their complement purely in terms of its type. We require two features for this purpose. Following Uszkoreit (1987a) in his treatment of German word order within the GPSG framework, we use a feature [±Main] to distinguish clauses which have 'main' and 'subordinate clause' word orders,[12] and a feature [±Topic] to distinguish clauses that do and do not have a topic in initial position. Thus, verb-final clauses are s[−Topic,−Main], verb-first clauses s[−Topic,+Main], and verb-second clauses s[+Topic,+Main]. For convenience, we abbreviate these types as s, m and m_t respectively.[13]

Verb-final clauses

Since we assume the null head location option for Dutch subordinate clauses, it follows that the position at which the finite verb appears is determined by the directionality that the verb type specifies for its complements. Concerning the 'ordering principles' for Dutch verbs, i.e. conditions on the assignment of directionality to arguments of prelocation verb types, we here simply assume that all NP and PP complements are sought to the left, and all verbal complements to the right. These assumptions are inadequate, but are sufficient for the immediate purpose of illustrating the account. These conditions allow for prelocation verbs types such as the following:

(4.65)	kent	(knows)	$s\obackslash np\obackslash np$
	heeft	(has)	$s\obackslash np\oslash(s\obackslash np)$
	geloof	(believes)	$s\obackslash np\oslash\Box sp$
	vertelde	(told)	$s\obackslash np\obackslash np\oslash\Box sp$

Note that these types are ultimately functions into type s, i.e. bearing the [Main] feature value used to identify subordinate clauses. This marking of prelocation types as [−Main] is

[12] Note, again that there is not a straightforward correspondence between the distribution of verb-first, verb-second and verb-final clauses and the (actual) main/subordinate clause distinction, and so the use of the feature name [Main] should not be taken too seriously. We use [+Main] for clauses which have the word order typical for main clauses, i.e. verb-first and verb-second clauses, and [−Main] for verb-final clauses.

[13] The use of s here perhaps provides for some confusion. The intention is that where s is shown alone, the feature values [−Topic,−Main] are to be assumed 'by default', but that wherever the features are explicitly mentioned, the stated values take precedence.

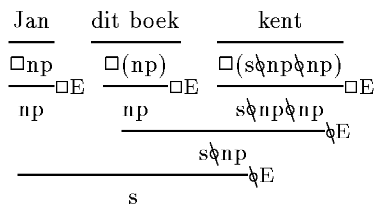
merely for convenience, and not intended to have contentful significance.[14] Such prelocation types are accorded located status without needing to undergo a head location rule, and ultimately give rise to lexical types such as (4.66), which allow for subordinate clause examples such as (4.67), as the proofs (4.68) and (4.69) illustrate.[15]

(4.66)	kent	(knows)	$\Box(s \obslash np \obslash np)$
	heeft	(has)	$\Box(s \obslash np \oslash (s \obslash np))$
	geloof	(believes)	$\Box(s \obslash np \oslash \Box sp)$
	vertelde	(told)	$\Box(s \obslash np \obslash np \oslash \Box sp)$

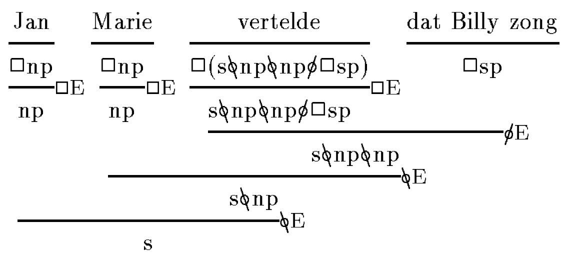
(4.67) a. ... omdat Jan dit boek kent
because Jan this book knows
'... because Jan knows this book'

b. ... omdat Jan Marie vertelde dat Billy zong
because Jan Marie told that Billy sang
'... because Jan told Marie that Billy sang'

(4.68)... omdat



(4.69)... omdat



Verb-first clauses

The following head location rule yields located types that ultimately give rise to main clause finite verb lexical types:

(4.70) a. Main Clause Finite Verb Location:

$$V \Rightarrow m/(s/\triangle V) \qquad \text{(where V is a finite verb)}$$

b. $\lambda f \lambda g.(gf)$

[14] We could, for example, adopt a more complicated scheme which featurally distinguishes main, subordinate and prelocation types, e.g. with a feature [vs] for 'verb status' taking values main, sub and preloc, the latter being marked on prelocation types. Then, the step by which prelocation types are accorded located status to allow for subordinate clauses might be made explicit in the operation of a lexical rule of the form: s[vs:preloc]$\phi \Rightarrow$ s[vs:sub]ϕ. We follow the path we have taken because it simplifies presentation.

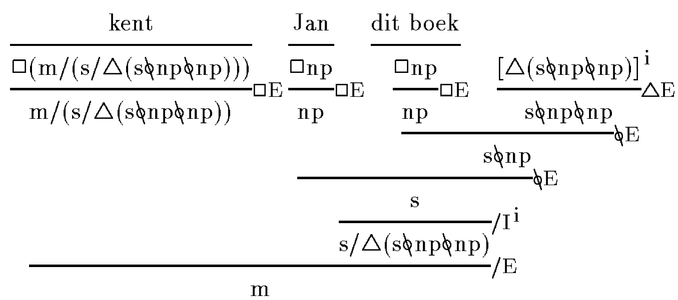
[15] Note that the type for *heeft* in (4.66) would *not* give the appropriate word order for most subordinate clauses, given that Dutch exhibits a phenomenon of *verb raising* or *verb clustering,* which results in the auxiliary's VP complement appearing as a discontinuous constituent. For example, the VP *appels aan Marie gegeven* appears discontinuously in (4.54a). We do not attempt to handle verb raising here.

The types produced by this rule are ultimately functions into type m, the type of verb-first clauses. (4.71) shows some final lexical type assignments that result from the application of this rule.

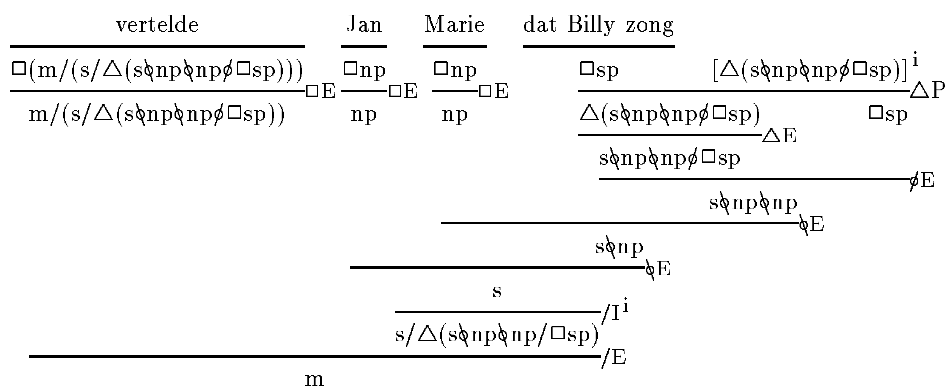
(4.71)	kent	(knows)	□(m/(s/△(s⦸np⦸np)))
	heeft	(has)	□(m/(s/△(s⦸np⊘(s⦸np))))
	geloof	(believes)	□(m/(s/△(s⦸np⊘□sp)))
	vertelde	(told)	□(m/(s/△(s⦸np⦸np⊘□sp)))

These types only allow a finite verb to appear left-peripherally to its *own* sentential projection (for the same reasons that this is true also for the types produced in the account of English). The proofs (4.72) and (4.73) illustrate the use of such types in deriving verb-first variants of the examples in (4.67).[16]

(4.72)



(4.73)



Verb-second clauses

We next consider the treatment of verb-second clauses. We assume that the initial constituent of verb-second clauses, commonly termed its *topic,* is extracted to the location at which it appears. Such topic constituents require raised types of the form m_t/(m/△□X),

[16] Note how in (4.72), the verb's arguments are combined to give a type s/△(s⦸np⦸np). This provides a parallel between the present account and an earlier account of Dutch subordinate clause word order offered by Steedman (1985). That account, which is set within a flexible CG framework, emphasizes the possibility of forming similar (non)constituents using type-raising and composition (with types based on standard slash connectives, i.e. without △, ⊘ or ⦸). Steedman motivates these proposals in relation to coordination, the treatment of extraction and the possibility of incremental interpretation.

i.e. enabling them to appear left peripherally to a constituent of type m from which they have been extracted. The following lexical rule generates these types:

(4.74) a. Topicalization Rule:

$$\mathrm{X}\phi^n \Rightarrow (\mathrm{m}_t/(\mathrm{m}/\triangle\Box\mathrm{X}))\phi^n \qquad \text{where X} \in \text{TOPIC}$$

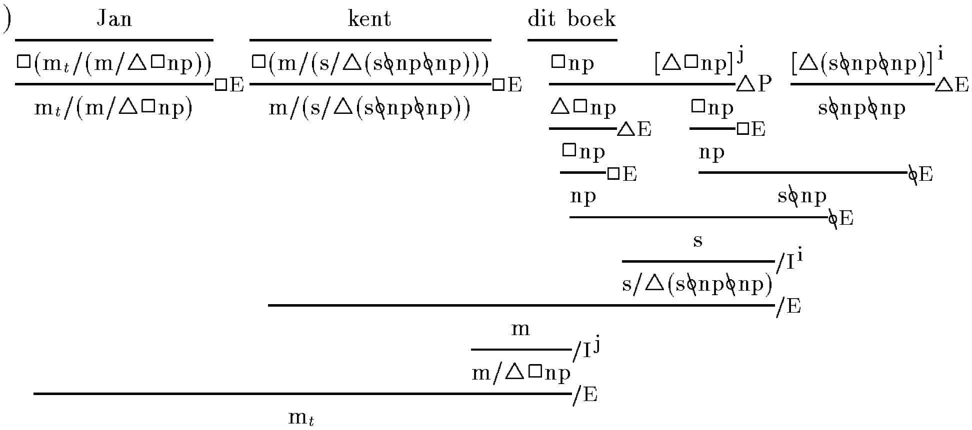
b. $\lambda \mathrm{f}.\lambda \mathrm{x_1} \mathinner{..} \mathrm{x_n}.\lambda \mathrm{g}.[\mathrm{g}(\mathrm{f}\mathrm{x_1} \mathinner{..} \mathrm{x_n})]$

We assume that the rule applies in the lexicon prior to the final 'box addition' step (cf. rule (4.7)). We restrict the operation of the rule in terms of a set TOPIC which specifies the topicalizable types. A full characterization of the membership of this set is beyond the scope of this work, but it evidently includes NPs PPs and adverbials, amongst others. The rule is schematic to allow for cases where a lexical type is not topicalizable itself, but is a function into a type that is topicalizable. For example, since noun phrases can be topicalized, we must provide types for nouns which when combined with their determiner yield a NP topic type. Thus, the noun type $\mathrm{np}\backslash\!\!\!\circ\mathrm{det}$ gives rise to a type $\mathrm{m}_t/(\mathrm{m}/\triangle\Box\mathrm{np})\backslash\!\!\!\circ\mathrm{det}$, a function from determiners to NP topic types. The semantics of the rule, given in (4.74b), is also schematic. This is a 'generalized' version of type raising, and involves abstracting over the arguments of a function before type raising the result.

The $\Box$-operator which appears in the topic types $\mathrm{m}_t/(\mathrm{m}/\triangle\Box\mathrm{X})$ is their penetrative modality (cf. Chapter 3), determining the behaviour shown with respect to modal boundaries. This is the 'unmarked' box (i.e. that having the defining language $\mathcal{L}_\emptyset$), and so, under the Chapter 3 assumption that $\mathcal{L}_\emptyset$ is a subset of all other defining languages used, these topic types would uniformly allow for unbounded extraction. However, since different types typically exhibit different behaviour with respect to island constraints it might, in a more detailed development of the account, be necessary to provide different topicalization rules to generate the topic types of different input types or to condition the operation of the rule with respect to this. We ignore this issue here.

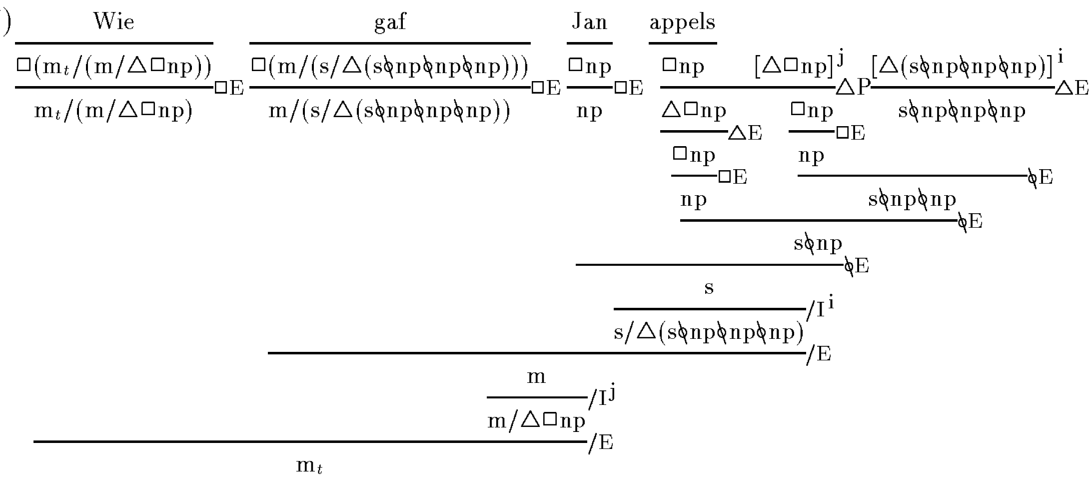
The following examples show some proofs for verb-second sentences:

(4.75)



(4.76) Wie gaf Jan appels?
who gave Jan appels
'Who did Jan give apples?'

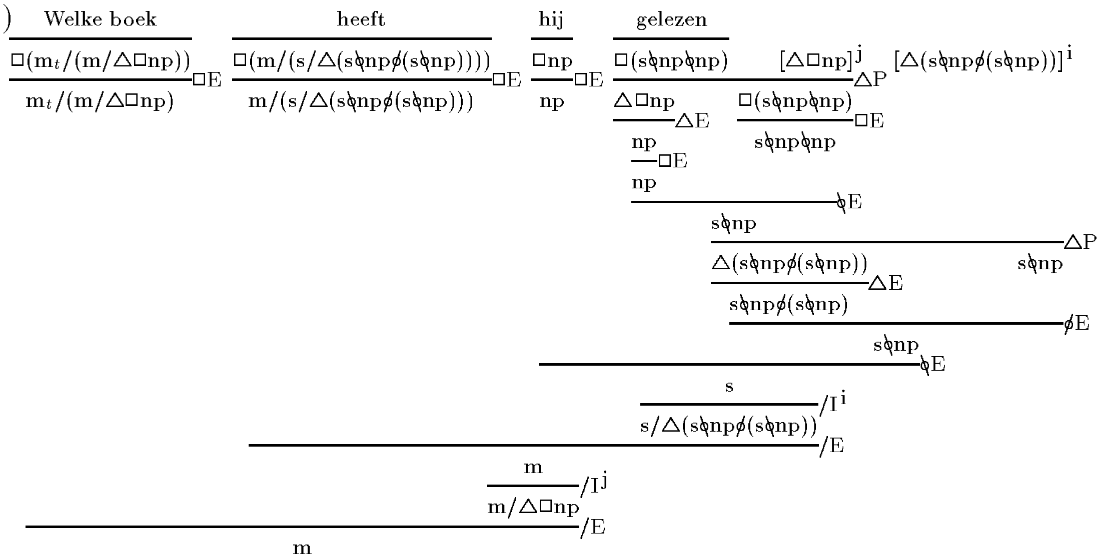
(4.77)



For the preceding examples, the topic is a direct argument of the fronted finite verb. In the next, the topic is extracted from the VP complement of the finite verb.

(4.78) Welke boek heeft hij gelezen?
which book has he read
'Which book did he read?'

(4.79)



Complex fronting

An interesting possibility for verb-second clauses involves what has been termed *complex fronting*, where a non-finite verb may be preposed into sentence-initial position, possibly together with some complements and modifiers, as the following examples illustrate:

(4.80) a. Jan heeft de appels aan Marie gegeven
Jan has the apples to Marie given
'Jan gave the apples to Marie'

b. de appels aan Marie gegeven heeft Jan
the apples to Marie given has Jan

c. aan Marie gegeven heeft Jan de appels
to Marie given has Jan the apples

d. gegeven heeft Jan de appels aan Marie
given has Jan the apples to Marie

This phenomenon has commonly been referred to as VP-preposing, but as examples (4.80c,d) show, sub-VP projections can also be fronted. It has been argued, however, that the topic of Dutch main clauses must be a constituent. For example, it is impossible to topicalize a non-constituent consisting of two NPs, or an NP and an argument PP:

(4.81) *de appels aan Marie heeft Jan gegeven
the apples to Marie has Jan given
'Jan gave the apples to Marie'

These facts present a problem for accounts which assume that VPs have an essentially flat structure, since sub-VP projections must then be seen as non-constituents. To allow for complex fronting, such accounts must allow some kind of structure to be given for sub-VP projection 'non-constituents', and do so in a way that explains the difference between these cases and those such as (4.81), where the fronting of non-constituents is not allowed. If the admission of non-constituents requires considerable violence to the general apparatus of the approach, a problem arises of how to explain the otherwise consistent facts about constituent order observed with complex fronting. GPSG accounts of complex fronting (in German) are given in Nerbonne (1986), Johnson (1986) and Uszkoreit (1987b). These accounts involve a radical modification of the GPSG framework which, as Bouma (1988) observes, gives analyses reminiscient of CG analyses.

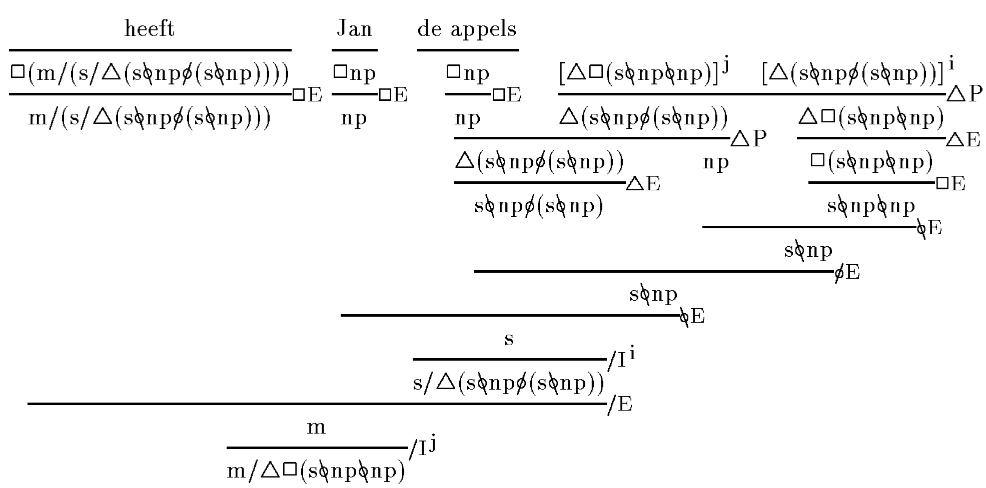
Bouma (1988) suggests that a more natural account can be given in a FCG framework, in which the frontable sub-VP projections arise straightforwardly when a verb has combined with some, but not all, of its arguments. The FCG framework that Bouma uses is augmented with a GPSG-style gap feature mechanism for handling extraction. Here, we shall briefly consider how a related treatment of complex fronting may be given in the Lambek framework, without using a GPSG-style treatment of extraction.

Bouma's central observation concerning complex fronting and a CG treatment of constituency carries across directly to our framework. For each of the complex fronting cases in (4.80), our grammar allows the fronted complex to be analysed as a single constituent. The initial position of this constituent can be handled in line with the extraction of the other topics. The question arises of how we should generate the raised verb types needed to allow for the topicalization of not only verbs, but also their many projections. In fact, the Topicalization rule given above can already provide such types. For example, from the prelocation type $s\backslash np\backslash np\backslash pp$ of the verb *gegeven,* the rule would yield a type:

$$m_t/(m/\triangle\Box(s\backslash np\backslash np))\backslash pp$$

allowing a derivation for (4.80c) as shown in (4.82) (the proof of which has been split into two parts for reasons of space).

(4.82)(a)



(b)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{aan}}{\Box(pp/np)}}{pp/np}\Box E \quad \dfrac{\dfrac{\text{Marie}}{\Box np}}{np}\Box E}{pp}/E \qquad \dfrac{\dfrac{\text{gegeven}}{\Box(m_t/(m/\triangle\Box(s\backslash np\backslash np))\backslash pp)}}{m_t/(m/\triangle\Box(s\backslash np\backslash np))\backslash pp}\Box E}{m_t/(m/\triangle\Box(s\backslash np\backslash np))}\backslash E \qquad \begin{array}{c}\text{heeft Jan de appels}\\ \vdots \\ m/\triangle\Box(s\backslash np\backslash np)\end{array}}{m_t}/E$$

This method for generating topic types for complex fronting has the disadvantage that it would give an enormous expansion in the size of the lexicon. We might alternatively look for a means of generating these types in the syntactic domain to avoid this problem, although we will not attempt to solve this problem here.

The facts of complex fronting in Dutch are, however, far more complicated than the above examples indicate, and so the above discussion is merely intended to suggest that the present approach may provide an appropriate framework for the study of complex fronting. The details of an adequate account remain a topic for further research.

Right extraposition in main clauses

We next briefly consider the possibility of extraposition or right extraction in Dutch, looking at the case of relative clause extraposition. This occurs in both main and subordinate clauses, and in both cases can be treated as extraction to the right of s (i.e. s[−Main,−Topic]), since proofs for main clause examples involve a subproof of a constituent of type s as a subpart. The goal here is merely to show that our treatment has no problem handling the cooccurrence of topicalization and right extraposition in main clauses.

The sentences in (4.83) show examples of relative clause extraposition in main clauses:

(4.83) a. Wie heeft de man gezien die de appels at?
whom has the man seen that the apples ate
'Whom did the man that ate the apples see?'

b. Jan gaf het boek aan Bill dat Marie kocht
Jan gave the book to Bill that Marie bought
'Jan gave the book to Bill which Marie bought'

c. Jan heeft het boek gelezen dat Marie kocht
Jan has the book read that Marie bought
'Jan has read the book which Marie bought.'

We can handle such cases by introducing another lexical rule, similar to that for topicalization, which generates raised types that extract relative clauses to a right peripheral position in the clause. We use *rel* to abbreviate the postnominal modifier type np⍀det⍀(np⍀det).

(4.84) a. Relative Clause Extraposition Rule:

$$\mathrm{rel}\phi^n \Rightarrow (\mathrm{s}\backslash(\mathrm{s}\backslash\triangle\mathrm{rel}))\phi^n$$

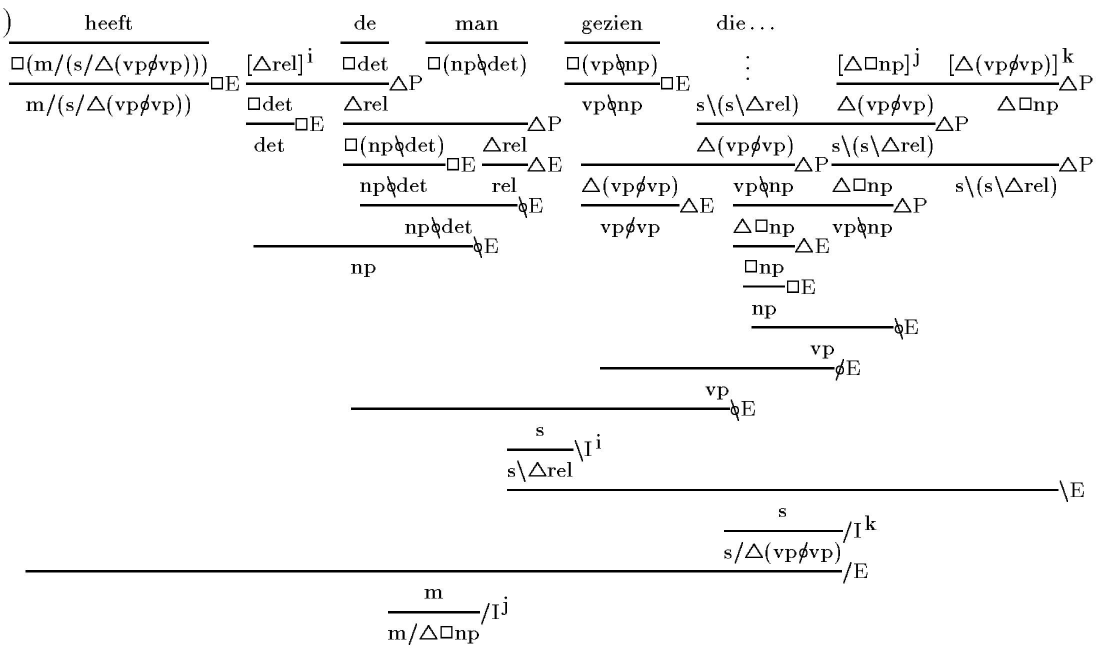
b. $\lambda f.\lambda x_1 \ldots x_n.\lambda g.[g(f x_1 \ldots x_n)]$

Note that the types produced by this rule allow only bounded movement of the relative clause, given the absence of any penetrative modality. The operation of the rule would give relative pronouns such as the following (ignoring the identities of the penetrative and boundary modalities):

$$\Box(\mathrm{s}\backslash(\mathrm{s}\backslash\triangle\mathrm{rel})/\Box(\mathrm{s}/\triangle\Box\mathrm{np}))$$

This type allows the following proof of the example in (4.83a). The proof shows just the derivation of the m/△□np constituent required as argument by the topic type of *wie*, and the subproof of the relative clause as type s\(s\△rel) is omitted.

(4.85)



A note on the V2 behaviour of German

We will not say much about German here other than that we assume essentially the same treatment for German V2 as for Dutch, i.e. with the head location rule for main clause finite verbs causing them to appear left adjacent to their own sentential projection, and the null head location option applying for all other verbs. Evidence that the null head location option applies for subordinate clause verbs can again be adduced from observations about the distribution of particles (or separable prefixes), very much as for Dutch.

4.7.4 The V2 behaviour of Scandinavian languages

The constituent order of Scandinavian languages such as Swedish and Norwegian is somewhat different from that of the West Continental Germanic languages such as Dutch and German, but they do exhibit characteristic V2 behaviour. We will here consider primarily the case of Swedish, although most of the generalizations stated apply also for other Scandinavian languages, and the methods we describe could equally well be applied to them.

The general facts about Swedish constituent order can be summarized as follows. The finite verb of Swedish main clause declaratives and constituent questions must occur in second position, i.e. following the first major constituent of the clause. In yes/no-questions, imperatives, and conditional clauses that lack a complementizer, the finite verb appears in first position. Some of these main clause possibilities are illustrated in the following examples, drawn from Platzack (1986a).

(4.86) a. Erik hade verkligen köpt boken
Erik had really bought the book

b. Erik köpte verkligen boken
Erik bought really the book

c. Den boken köpte Erik i London
That book bought Erik in London
'That book, Erik bought it in London'

d. *Erik verkligen hade köpt boken
Erik really had bought the book

e. *Erik verkligen köpte boken
Erik really bought the book

f. Vad hade Erik köpt i London?
What had Erik bought in London

g. hade Erik verkligen köpt boken?
had Erik really bought the book

When a non-subject complement precedes the finite verb, 'inversion' is observed, with the subject appearing following the finite verb.

In embedded clauses, the subject in general precedes the finite verb, so that a SVO order is observed. A sentential adverb, when present, typically appears before the finite verb giving an order such as:

Subj Adv V

as, for example, in the following sentences:

(4.87) a.	Jag	frågade	om	Erik	verkligen	hade	skrivit	boken
	I	asked	if	Erik	really	had	written	the book
b.	Jag	frågade	om	Erik	verkligen	köpte	boken	
	I	asked	if	Erik	really	bought	the book	

This demonstrates that the second (or first) position requirement on the position of the finite verb is absent in embedded clauses. This constrasts with the following two orders exhibited in corresponding main clauses:

Subj V Adv

Adv V Subj

of which the first is exemplified in (4.86). Thus, Swedish embedded clauses exhibit a general SVO ordering, which is in various regards similar to English word order.

Accounts of Scandinavian V2 behaviour within TG and GB have typically assumed that that the verb-first and verb-second orders of main clauses are derived from a deep structure SVO order characteristic of subordinate clauses. For example, the account of Platzack (1986) gives the following S-structure for the example in (4.86b):

(4.88) $[_{c''}$ Erik$_i$ $[_{c'}$ $[_{comp}$ köpte$_j]$ e$_i$ verkligen$_{advp}$ $[_{I'}$ $[_I$ v$_j]$ boken$_{np}]]]$

(Platzack assumes clauses to be a projection of COMP.) Observe that the position of the finite verb arises under movement to COMP, and that the 'topicalized' subject moves to a position to the left of C$'$. The base-generated constituent order is SVO. Taraldsen (1986) provides an account of Norwegian V2 which also assumes embedded clause word order to be basic, with the base-generated structure of Norwegian clauses as in (4.89) (where W represents an arbitrary string of sentence adverbials). X^m is the landing site for *wh*-phrases and topics, and finite verbs move to COMP in main clauses.

(4.89) $[_{s^2}$ X^m $[_{s^1}$ COMP $[_s$ NP W $[_{vp}$ V NP ...$]]]]$

For our account, we can, as before, handle the common facts of word order between Swedish main and subordinate clauses in terms of common prelocation types giving rise to the final lexical type assignments to verbs. The difference to the treatment of Dutch (and German) is that we assume that non-null head location processes apply for both main and subordinate clause verb types. We shall briefly sketch such an account.

Firstly, we assume the features [$\pm$Main] and [$\pm$Topic] as in the account of Dutch, with the symbols s, m and m$_t$ abbreviating featurally distinguished clause types as before. All verbs are subject to a head location process which provides types allowing them to appear left adjacent to their own VP projection, these types being featurally identified as subordinate clause types. The following rule is required for this:

(4.90) a. Swedish Subordinate Clause Verb Location rule:

$$\mathrm{vp}\phi \Rightarrow \mathrm{vp}/(\mathrm{vp}/\triangle(\mathrm{vp}\phi))$$

b. $\lambda f \lambda g.(gf)$

This is essentially the same as the rule for English verb location. As in the account of English, we might allow that prelocation types that already specify an appropriate complement order be accorded located status without needing to undergo the rule. In addition, finite verbs are subject to a second head location rule (4.91) which allows them to appear in clause initial position, these types being marked as main clauses types. This rule is similar to that used in the account of Dutch. The position of the initial constituent in verb-second main clauses is handled, as before, in terms of extraction.

(4.91) a. Swedish Main Clause Finite Verb Location rule:

$$\mathrm{V} \Rightarrow \mathrm{m}/(\mathrm{s}/\triangle\mathrm{V}) \qquad \text{(where V is a finite verb)}$$

b. $\lambda f \lambda g.(gf)$

For example, the two head location rules yield the located types (4.92b) and (4.92c) from the prelocation type for *köpte* ('bought') shown in (4.92a).

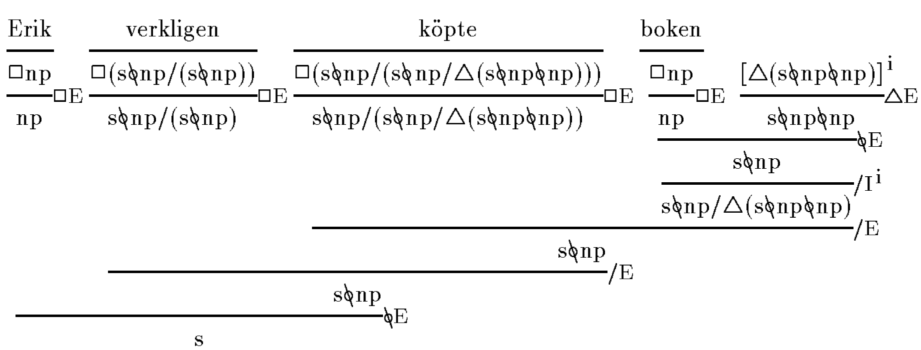
(4.92) a. s\np\np

b. □(s\np/(s\np/△(s\np\np)))

c. □(m/(s/△(s\np\np)))

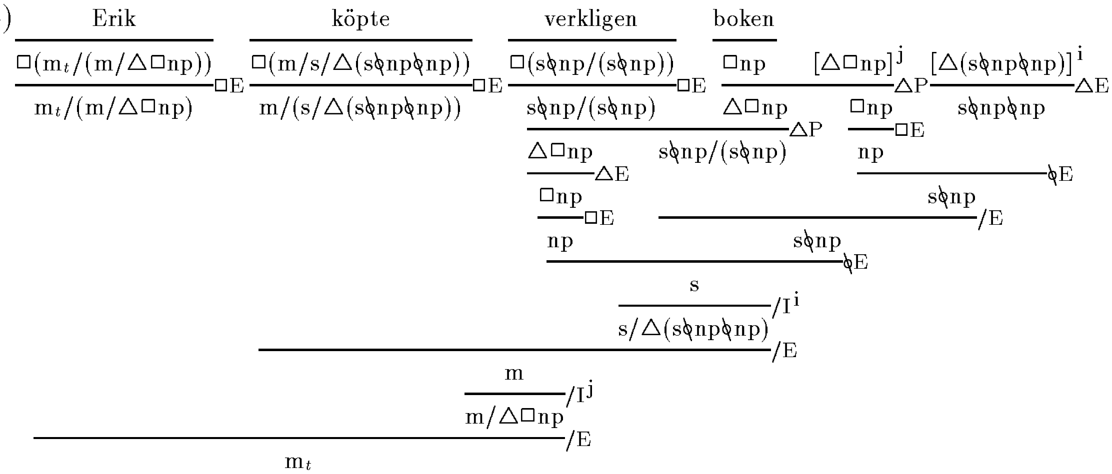
Consider this account in relation to our earlier discussion about adverb position. The fact that adverbs such as *verkligen* may appear between the subject and verb of embedded clauses suggests that they receive a subordinate clause VP premodifier type: □(s\np/(s\np)), which allows the proof (4.93) for the embedded clause of example (4.87b).

(4.93)... om



This type for the adverbial also allows for the related main clause (4.86b), where the adverbial appears following the verb, as the proof in (4.94) illustrates.

(4.94)



It is not possible to derive the embedded clause order *Erik verkligen köpte boken* as a main clause (cf. the ungrammatical (4.86e)), i.e. as type m_t (or m), because in a main clause derivation, *köpte boken* is analysed as a clause lacking an extracted subject, whilst *verkligen* is a (subordinate clause) VP modifier.

The account as we have sketched it overgenerates in allowing the adverb to appear following the finite verb in an embedded clause, admitting examples such as:[17]

(4.95)	*Jag	frågade	om	Erik	köpte	verkligen	boken
	I	asked	if	Erik	bought	really	the book

The reason for this is obvious if we look at the subordinate clause derivation in (4.93). Observe that a (subordinate clause) VP node (i.e. type s⍀np) appears as a projection of the hypothesized (prelocation) verb type in the proof. The adverbial could apply to this VP node, in which case the word order in (4.95) would result. This overgeneration indicates the need for a more fine grained featural distinction to be made between the different s subtypes occurring in located verb types than is achieved using the feature [±Main] (cf. footnote 14). We shall not attempt to develop the account in this direction here.

This treatment of Swedish V2 is clearly distinct from that of TG and GB even at an intuitive level. We don't explain the common facts about main and subordinate clause constituent order by assuming that subordinate clause order is basic. Instead, the commonalities are explained in terms prelocation types, which specify a constituent order that is distinct from that of both main and subordinate clauses.

[17] It is interesting to note that this main clause order for verb and adverb *is* permitted in some Swedish embedded clauses (Platzack, 1986b). This order is in general ungrammatical, however, the exceptions occurring only in certain embedded clauses that are asserted, and where certain other root phenomena (such as Topicalization, Left Dislocation and VP Preposing) are also possible. This suggests that these embedded clauses should receive main clause analyses, i.e. be analysed of type m_t.

It is interesting to ask what advantages this view might have over that of TG and GB. Recall that Koster (1988) motivates his claim for residual SOV structure in English in relation to some problems that arise for the treatment of a range of phenomena when the base-generated order of English is assumed to be SVO. These problems are avoided under Koster's residual SOV hypothesis. Given the many similarities to be observed between English word order and the embedded clause word order of Scandinavian languages, we can expect that similar problems will arise for the treatment of the Scandinavian languages under the TG/GB hypothesis of SVO order as basic. These problems are avoided for the CG account of English by separating the specification of complement order (in terms of prelocation type) from the specification of head position. We can expect that our account will also avoid these problems for Scandinavian languages, by rooting the treatment of complement order in prelocation types where order of subcategorization reflects obliqueness.

4.8 Conclusion

We have presented a CG account of word order which separates the specification of the order of a head's complements from the specification of the position at which the head appears relative to them. This move has a number of advantages, allowing the adoption of an account of grammatical relations, and providing quite straightforwardly for the treatment of V2 phenomena.

More generally, our approach allows accounts to be given for different Germanic languages where, despite considerable word order differences, the accounts of the languages are to a large extent similar. This similarity reflects the fact that the different Germanic languages are in many ways closely related. For example, we may find that related terms of English and Dutch (e.g. the cognate verbs *give* and *geven*) have very similar initial (prelocation) types within the lexicon, which are subject to many lexical rules (e.g. Dative Shift) that are essentially the same in the two languages. The more obvious differences observed between such related languages can be seen to arise mainly from language particular 'ordering principles' and language particular requirements on head position.

It is hoped that the present approach, in allowing language particular facts about word order to be encoded without losing sight of the commonalities of related languages, will allow for productive cross-linguistic research to be pursued within a categorial framework.

Chapter 5

Binding

This chapter addresses the treatment of binding phenomena. We focus particularly on the treatment of anaphor binding, though some attention is also given to the binding of personal pronouns. An account of binding is developed in the extended Lambek calculus framework. This account provides a point of synthesis for the proposals of the preceding chapters. Firstly, 'command' constraints on binding are handled in terms of relative obliqueness, depending crucially for this on the account of word order presented in Chapter 4, with its incorporation of a treatment of grammatical relations. Secondly, locality constraints on binding are handled in terms of the polymodal treatment of linguistic boundaries presented in Chapter 3, hopefully demonstrating that the utility of the polymodal system extends beyond the treatment of just island constraints. The Lambek account is initially developed in relation to binding in English, but later in the chapter, a considerable amount of space is devoted to the treatment of long distance reflexivization in Icelandic, which is of particular interest in terms of its locality behaviour. We also consider a second Lambek account of binding, due to Moortgat (1990b), and develop it to avoid a number of problems. We begin by considering some previous approaches to binding.

5.1 Previous accounts of binding

5.1.1 The standard approach

The predominant view of the treatment of binding follows the formulation of Chomsky (1981) which may be summarized as follows (in which "bound" means "coindexed with a c-commanding constituent" and "free" means not bound):

(5.1) A. an anaphor is bound in its governing category

B. a pronominal is free in its governing category

C. a referential expression is free

This approach focuses on stating constraints on the syntactic domain, relative to the anaphor, in which a c-commanding antecedent must appear (where *governing category* in (5.1) denotes the relevant domain). We shall not go into the details of how 'governing category' has been defined. An example is that the reflexive in (5.2a) has the tensed clause for its governing category. Then, Principle A explains why in (5.2a) *Bill* is a possible antecedent for the anaphor whilst *John* is not. As another example, an anaphor occurring in a a picture noun complement (a 'picture noun anaphor') has the overall NP as its governing category only if the NP has a possessive NP specifier, explaining the contrast illustrated by (5.2b) and (5.2c). Note that a possessive NP specifier provides a possible antecedent for a picture noun reflexive, as in (5.2d).

(5.2) a. John_i thinks that Bill_j likes $\text{himself}_{i*/j}$

b. John_i likes a picture of himself_i

c. *John_i likes Mary's picture of himself_i

d. John likes Mary_i's picture of herself_i

It is generally assumed that syntactic binding is the only option available for English anaphors, and Principle A states a condition on such syntactic binding. Principles B and C state conditions under which coreference is *not* allowed, a phenomenon referred to as *disjoint reference*. For example, Principle B allows coreference in (5.3a) but not in (5.3b):

(5.3) a. He_i thinks that Mary likes him_i

b. *He_i likes him_i

5.1.2 Pragmatic accounts of obligatory non-coreference

Various problems have been noted for handling disjoint reference phenomena in terms of obligatory syntactic 'principles'.[1] Firstly, there is the problem of 'accidental coreference'. For example, (5.4), a principle C violation, might be uttered on seeing John Smith, but failing to recognize him:

(5.4) He_i looks just like John Smith_i

[1] See Reinhart (1983) for discussion.

Secondly, it is clear that under appropriate pragmatic conditions, violations of the disjoint reference principles can be perfectly acceptable, as in (5.5a,b):

(5.5) a. [The man who I saw]$_i$ was John$_i$.

b. Only John voted for John.

c. Only John voted for himself

Dowty (1980) suggests that disjoint reference arises for pragmatic reasons, from a "neo-Gricean conversational principle" which we might paraphrase thus:

> "Where two equally simple expressions A and B are available to express a meaning X, but where A unambiguously means X whilst B is ambiguous between meanings X and Y, then B should be reserved for expressing Y."

By this view, the coreferential reading of *he likes him* is syntactically possible but disfavoured pragmatically because the same meaning can be unambiguously expressed by *he likes himself*. In respect of (5.5b), note that this sentence differs subtly from, say, (5.5c). Whereas the former would be an appropriate response to a question *Did anyone vote for John?*, the latter would not.

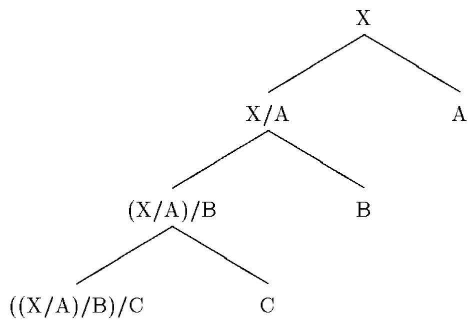
This idea is developed by Reinhart (1983), who argues that syntactic considerations determine only when a pronoun may be bound. A pronoun that is not bound may be assigned a referent under discourse (i.e. extra-sentential) processing, whereby 'accidental' coreference may arise. Reinhart assumes that anaphors *must* be bound within some specified domain. For personal pronouns, bound anaphora is optional, so that a pronoun may either be bound by, or accidentally coreferential with, a c-commanding referential NP, but can only be coreferential with a non c-commanding NP. Disjoint reference facts are handled pragmatically, in terms of whether or not a speaker chooses to exploit the bound anaphora options that the syntactic construction used makes available.

5.1.3 Obliqueness vs. c-command

Within approaches where order of subcategorization is taken to encode relative obliqueness (such as Montague Grammar and HPSG), it has been suggested that the c-command condition on binding of the standard view be replaced by a condition stated in terms of order of subcategorization (and thereby obliqueness). For example, a notion *F-command* has been used in Montague grammar, which is stated in terms of function-argument structure (Bach & Partee, 1980; Chierchia, 1988).

The F-command relation is such that an argument of a functor F-commands the 'earlier' (and therefore more oblique) arguments of the same functor and their subconstituents. For example, in the type (((X/A)/B)/C), argument A F-commands arguments B and C, whilst B F-commands only argument C. We can see that F-command is to some extent similar to c-command from the following diagram:

(5.6)



This diagram represents the combination of a function with its arguments in a tree-like fashion. We can see that the F-command relation holds of the same pairs of elements as stand in the c-command relation in the structure. Pollard and Sag (1990) suggest a notion *o-command*, which is similar to F-command but stated purely in terms of order of subcategorization and obliqueness, rather than function-argument structure. It is clear that F-command, o-command and c-command play essentially the same role in the respective theories. We speak simply of *command* when the given context does not require us to be specific as to which notion is used.

Chierchia (1988) notes one case where F-command avoids a problem that arises for c-command. F-command correctly predicts the possibilities for binding between the two objects of double object constructions, since direct objects are taken to be less oblique that second objects.

(5.7) a. Mary showed $John_i$ to $himself_i$

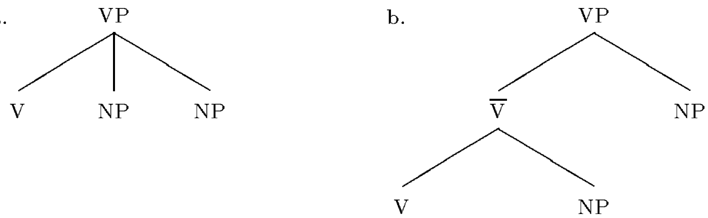
b. *Mary showed $himself_i$ to $John_i$

(5.8) a. Mary showed $John_i$ $himself_i$

b. *Mary showed $himself_i$ $John_i$

Chierchia points out that c-command fails to predict the ungrammaticality of (5.8b) under the widely assumed structure shown in (5.9a). Barss and Lasnik (1986) also note this problem, and consider in addition the alternative possible structure shown in (5.9b). This structure fares even worse than (5.9a), since it predicts precisely the converse of the observed grammaticality/ungrammaticality for the examples in (5.8).

(5.9) a. b.



5.1.4 Discourse controlled reflexives in English

It is commonly assumed that syntactic binding is the only option available for anaphors in English. Thus, reflexivization in an example such as (5.10a) is taken to be essentially the same phenomenon as that in (5.10b):

(5.10) a. John$_i$ painted a picture of himself$_i$

b. John$_i$ likes himself$_i$

Jackendoff (1972) first observed that a picture noun reflexive does not in general require a c-commanding antecedent:

(5.11) The fact there is a picture of himself$_i$ hanging in the post office is believed (by Mary) to be disturbing Tom$_i$ (Jackendoff, 1972)

Some further relevant examples, from Pollard and Sag (1990) (of which some are in turn drawn from other sources), are shown in (5.12). These examples suggest that picture noun anaphors and possessive reciprocals are not subject to either the c-command or specified domain constraints that govern the behaviour of most reflexives.

(5.12) a. A fear of himself$_i$ is John$_i$'s greatest problem

b. John$_i$'s campaign requires that pictures of himself$_i$ be placed all over town

c. The agreement that [Iran and Iraq]$_i$ reached guaranteed each other's trading rights in the disputed waters until the year 2010

d. The pictures of each other$_i$ with Ness made [Capone and Nitty]$_i$ somewhat nervous

e. John$_i$ was furious. The picture of himself$_i$ in the museum had been mutilated.

Pollard and Sag refer to such anaphors as *exempt anaphors*, since they are exempt from the command and domain requirements characteristic of syntactic binding. Note that in (5.12e), the antecedent occurs in a different clause to the anaphor, suggesting very forcefully that exempt anaphors may be discourse controlled. Pollard and Sag argue that this data indicates that recent attempts to handle examples such as (5.12b) in terms of conditions for expansion of binding domains are mistaken.

Although picture noun anaphors and possessive reciprocals appear to be exempt from the command and domain constraints that govern syntactically bound anaphors, this does not mean that they are simply free in their distribution. Pollard and Sag (1990) discuss some *non-grammatical* constraints on exempt anaphors that have been proposed in the literature. One of these constraints involves *point of view*. In particular, if the exempt anaphor appears in a stretch of discourse which reportively presents a given individual's point of view, the exempt anaphor is typically required to take that individual as its referent. This is illustrated by the following examples (from Pollard and Sag, 1990):

(5.13) a. John_i was going to get even with Mary. That picture of himself_i in the paper would really annoy her, as would the other stunts he had planned.

b. ?*Mary was quite taken aback by the publicity John_i was receiving. That picture of himself_i in the paper had really annoyed her, and there was not much she could do about it.

In (5.13b), where Mary's view is presented, taking John as the referent for the reflexive is most infelicitous.

Do syntactic and discourse binding possibilities overlap?

Given that English anaphors exhibit *both* syntactic and discourse binding, an important question arises as to the alternative circumstances under which the two kinds of binding are possible. We next consider the account of binding of Pollard and Sag (1990), since it offers one answer to this question, namely that syntactic and discourse binding show *complementary* distribution. Later, we argue that Pollard and Sag's view of the data is incorrect, and that the possibilities for syntactic and discourse binding overlap.

Pollard and Sag's account of exempt anaphors

Pollard and Sag (1990) offer an account of binding within the HPSG framework which is intended to explain both when syntactic binding of anaphors occurs (what Pollard and Sag call *obligatory binding*[2]) and the command/domain requirements on syntactic binding, as well as the circumstances under which exempt anaphors arise.

Recall that subcategorization dependencies in HPSG are expressed in terms of a syntactic feature SUBCAT, which takes as its value a list of *signs,* indicating the complements that a given lexical item combines with syntactically, where the order of signs on the SUBCAT list

[2] The use of this term is perhaps a little unfortunate, since it carries implicit assumptions about the nature of syntactic binding. We will argue later in this section that examples arise where *both* syntactic and discourse binding are possible, and so referring to the syntactic option as 'obligatory' seems inappropriate

corresponds to their relative obliqueness. Pollard and Sag's account of binding in English is based on a condition that can be (informally) paraphrased as follows:[3]

(5.14) *Principle A. of the the HPSG Binding Theory:*
An anaphor complement of a head must be co-indexed with some (referential) less oblique dependent of the same head—provided such exists.

This condition states when syntactic binding of an anaphor is obligatory, and also the requirements for a suitable antecedent in this case. The condition gives rise to both the locality constraint on syntactic binding (i.e. the binder must occur within the 'subcategorization domain' of the same head) and the command constraint (in terms of obliqueness).

If the head which governs an anaphor does not specify any less oblique referential complements, then the condition in (5.14) does not apply and so places no constraint on the anaphor. It is under these circumstances that exempt anaphor behaviour is taken to occur. This situation arises in the case of picture noun anaphors and possessive reciprocals. For example, in (5.15a), the noun *picture* specifies two complements—its specifier and the anaphoric complement (i.e. the PP).[4] The determiner is less oblique than the anaphoric complement, but the condition (5.14) does not apply since determiners are not referential. In (5.15b), the specifier is a possessive NP which *is* referential. Hence, the condition (5.14) does apply, so that the sentence is ungrammatical since the anaphor is not coindexed within the immediate domain of its head (the noun). Note that coindexation of the anaphor with the possessive NP, as in (5.15c), satisfies the condition. A reciprocal may appear *as* a possessor NP, as in (5.16). Again, HPSG's Principle A does not apply, and so exempt behaviour is predicted.

(5.15) a. John$_i$ likes every picture of himself$_i$

b. *John$_i$ likes Mary's picture of himself$_i$

c. John likes Mary$_i$'s picture of herself$_i$

(5.16) The children like each other's friends.

Non-exclusive possibilities for syntactic and discourse binding

It should be clear that Pollard and Sag's account predicts that syntactic and discourse binding show *complementary* distributions. We next consider if this predication is correct.

[3] See Pollard and Sag (1990) for details of the more rigorous statement of this condition.

[4] Pollard and Sag treat PPs headed by "case-marking" prepositions as inheriting their 'referential index' (which includes features that specify whether or not an item is anaphoric) from their NP complement, in line with a common assumption that purely case-marking prepositions do not contribute to the semantic content of the phrase that they head. This inheritance of referential index has the consequence that where a case-marking preposition has a anaphor for its complement, the resulting PP itself behaves as an anaphor.

Consider the following examples, in which Mary's point of view is presented:

(5.17) a. Mary$_i$ really enjoyed the race, and so she$_i$ decided that the winning boy would receive a framed picture of herself$_i$ as a special prize.

b. Mary really enjoyed the race, and so she decided that the winning boy$_i$ would receive a framed picture of himself$_i$ as a special prize.

c. ?*Mary really enjoyed the race, and so she decided that the mother of the winning boy$_i$ would receive a framed picture of himself$_i$ as a special prize.

Pollard and Sag's account predicts that only (5.17a) should be acceptable (since only in it does the antecedent for the reflexive satisfy the point of view requirement), but example (5.17b) is also clearly acceptable. The obvious difference between (5.17b) and (5.17c) is that in (5.17b) the antecedent commands the reflexive but in (5.17c) it does not.

The sentences in (5.18) present John's point of view, but Mary is taken as antecedent, and so Pollard and Sag's account predicts that both sentences should be ungrammatical.

(5.18) a. ?*John was going to get even this time. He told Mary$_i$ that he had received some highly compromising photographs of herself$_i$ and that his silence was going to cost her a lot of money.

b. John was going to get even this time. He told Mary$_i$ that she$_i$ would receive some highly compromising photographs of herself$_i$ and that the negatives were going to cost her a lot of money.

The difference between the two examples is that in the grammatical sentence, the antecedent occurs within what is standardly viewed as an appropriate domain, whilst in the other sentence it does not.[5] The data in (5.17) and (5.18) show that picture noun reflexives are subject to the point of view requirement only when the antecedent fails to satisfy the command and (standard view) domain requirements for syntactic binding, strongly suggesting that syntactic binding *is* possible for picture noun anaphors. Since such anaphors clearly also exhibit *exempt* behaviour, it appears that the possibilities of syntactic binding and discourse controlled behaviour are *not* disjoint.

If we are correct to claim that an anaphor can be bound by an antecedent occurring outside the domain of the head governing the anaphor, then this clearly presents a considerable problem for Pollard and Sag's account. Their treatment of locality constraints on syntactic

[5] One possible counterargument here is that it is not clear that John's point of view *is* maintained throughout in (5.18b), i.e. Mary's point of view may take over. The following discourse at least indicates that John's point of view is not *necessarily* displaced, although the possibility remains that (5.18b) is ambiguous as to whose point of view is dominant:

> John$_i$ was going to get even this time. He$_i$ told Mary that she would receive some highly compromising photographs of himself$_i$ and that the negatives were going to cost her a lot of money.

The examples (5.18) are at the very least in accordance with the hypothesis to be proposed.

binding crucially depends on the anaphor and antecedent occurring as dependents of the same head, and it is unclear how the approach could be modified to characterize locality constraints on syntactic binding where antecedent and anaphor are not codependents.

When are anaphors exempt?

We have argued that Pollard and Sag are wrong to claim that picture noun anaphors exhibit only discourse based coreference. It remains possible that Pollard and Sag's account does correctly characterize the conditions under which exempt anaphors *can* (rather than must) occur. We will argue that their account is incorrect in this regard also.

Consider the following examples involving the noun *speech:*

(5.19) a. I overheard John$_i$'s speech to the committee about himself$_i$

b. I overheard John$_i$'s speech to himself$_i$ about the committee

c. I overheard John's speech to the committee$_i$ about itself$_i$

d. *I overheard John's speech to itself$_i$ about the committee$_i$

e. *I overheard John's speech about the committee$_i$ to itself$_i$

These examples suggest that the noun's complementation mirrors that of the verb *to speak*, i.e. the *to*-phrase and *about*-phrase are complements of the noun, with the *to*-phrase being less oblique than the *about*-phrase. For sentences where the *to*-phrase has a referential NP complement, Pollard and Sag's account predicts that an anaphoric *about*-phrase should be unable to show exempt behaviour. The following example shows that this is not the case:

(5.20) Mary$_i$ was furious. A speech to the committee about herself$_i$ had been highly critical, but she had been given no chance to reply.

The next example (involving the noun *report*, which shows the same complementation as *speech*) indicates that an anaphor occurring in two-complement noun constructions can also be syntactically bound by an antecedent occurring outside the NP:

(5.21) John was furious. He discovered that Mary$_i$ had seen a report to the committee about herself$_i$, despite his express instructions that she be told nothing.

Note however, that a possessive NP specifier prevents such an anaphor taking an antecedent occurring outside the NP for both discourse binding (attempted in (a)) and syntactic binding (attempted in (b)):

(5.22) a. ?*Mary$_i$ was furious. John's speech to the committee about herself$_i$ had been highly critical, and she had been given no chance to reply.

b. ?*Mary$_i$ overheard Bill's speech to the committee about herself$_i$.

This data (particularly (5.20) and (5.22a)) suggests that Pollard and Sag's explanation of why the presence of a possessive NP specifier prevents exempt behaviour for a picture noun anaphor is incorrect, i.e. it cannot be simply because the possessive constitutes a less oblique dependent of the same head (the noun).

5.1.5 Previous categorial accounts of Binding

We next briefly consider two previous categorial approaches to binding.

Storage based accounts

Bach and Partee (1980) and Chierchia (1988) present accounts of binding within the (extended) Montague Grammar framework that are based on the technique of Cooper-storage (Cooper, 1975; 1983). We will briefly sketch a storage based approach.

Firstly, expressions are associated with a syntactic type, a semantic translation and a STORE as follows:

<string, type, translation, STORE>

The STORE is used to hold information concerning pronouns that occur within the expression. Example lexical entries for a verb and a proper noun are as follows (we assume a non-directional CG approach):

(5.23) <loves, S/NP/NP, loves′, $\emptyset$ >
<john, NP, john′, $\emptyset$ >

Note that these lexical entries have empty STOREs. The entry for a reflexive pronoun, however, has a non-empty STORE:

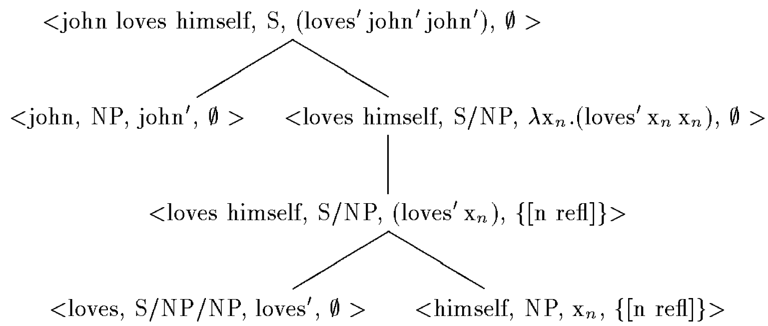
(5.24) <himself, NP, x_n, {[n refl]}> (where n is some integer)

Observe that the pronoun has been assigned a variable for its translation, and that the STORE contains an entry which records the presence of a reflexive pronoun, and that its translation variable has subscript n. When expressions are combined, STORE information is inherited 'upwards' onto the result of the combination. At some node, a rule called the *Reflexive Interpretation Rule* applies, accessing information from a STORE and using it to perform a semantic operation, as follows:

(5.25) Reflexive Interpretation Rule:
<E, Cat, T_1, ST_1> $\Rightarrow$ <E, Cat, T_2, ST_2>
where Cat $\in$ {VP, TVP}, [n refl] $\in ST_1$, $ST_2 = ST_1 -$ [n refl],
and $T_2 = \lambda x_n.(T_1\ x_n)$

The variable x_n must already occur within the translation T_1. The rule applies this translation to the named variable and then abstracts over the variable in the resulting expression, so that the two variable occurrences become co-bound by a single lambda operator. When the resulting functor is applied to its next argument, this expression will fill two argument positions in the resulting translation (i.e. one for each variable occurrence). The following derivation illustrates the use of the rule:

(5.26)



For a sentence such as *Mary*$_i$ *showed John herself*$_i$, in which binding occurs between the subject and the second object (corresponding to the first and third arguments of the verb), the STORE entry introduced by the reflexive must be inherited up through more ‘nodes’ of the derivation before the application of the Reflexive Interpretation Rule.

It is a characteristic of this approach that an anaphor must be F-commanded by its antecedent. This follows simply because the Reflexive Interpretation Rule must apply at a higher node in the ‘tree’ (i.e. function-argument structure) than the node where the anaphor appears. Locality constraints on reflexivization can be handled, for example, by stipulating that stored entries for reflexives cannot be inherited onto derivation nodes of certain types (e.g. NP and S).

Szabolcsi’s account of reflexivization

Szabolcsi (1987a) proposes an account of binding within the framework of Combinatory Categorial Grammar (Steedman, 1987; Szabolcsi 1987b). This account was briefly discussed in Chapter 3, but we will refresh our memory of the main ideas before we go on to discuss it. As we saw, a CCG formulation consists of a set of combination rules such as (5.27), including rules of functional application together with some further rules such as type-raising and composition. The CCG treatment of rule semantics is based around the use of *combinators* rather than the lambda calculus.

(5.27)	(>)	$X/Y,\ Y \Rightarrow X$	
	(<)	$Y,\ X\backslash Y \Rightarrow X$	
	(**T**)	$X \Rightarrow Y/(Y\backslash X)$	$X \Rightarrow Y\backslash(Y/X)$
	(**B**)	$X/Y,\ Y/Z \Rightarrow X/Z$	$Y\backslash Z,\ X\backslash Y \Rightarrow X\backslash Z$
	($\mathbf{B}^2$)	$X/Y,\ Y/Z/W \Rightarrow X/Z/W$	$Y\backslash Z\backslash W,\ X\backslash Y \Rightarrow X\backslash Z\backslash W$

Szabolcsi suggests that reflexives be treated as functions over verbs, with types that are such that when combined with a verb type V, reduce its arity by one (i.e. return a type like V but with one argument removed). In line with the CCG view of semantics, Szabolcsi argues against treating reflexives as if they introduce a variable in the semantics that is later bound by an operator. Instead, the reflexive is assigned a combinatory semantics which, in effect, 'identifies' the verb's eliminated argument position with another one of its argument positions.[6] For example, Szabolcsi suggests that reflexives be assigned the type (5.28a), corresponding to that of an object NP raised over its verb, and with semantics **W**, a 'duplicating' combinator whose meaning is shown by the equivalence in (5.28b):

(5.28) a. $(s\backslash np)\backslash(s\backslash np/np)$

b. $\mathbf{W} = \lambda f \lambda x.[f\,x\,x]$

This type allows the derivation for the sentence *John loves himself* shown in (5.29), with meaning as shown in (5.30a), equivalent to the expression in (5.30b).

(5.29)

$$\dfrac{\dfrac{\text{John}}{np}\quad \dfrac{\dfrac{\text{loves}}{s\backslash np/np}\quad \dfrac{\text{himself}}{(s\backslash np)\backslash(s\backslash np/np)}}{s\backslash np}<}{s}<$$

(5.30) a. **W** loves′ john′

b. loves′ john′ john′

Szabolcsi uses an obliqueness based command relation, which depends on the adoption of a version of Dowty's theory (i.e. with argument order encoding obliqueness). We have seen that, taking this view of subcategorization, Montague Grammar requires the use of wrapping operations to handle word order. Szabolcsi does not adopt the use of wrapping, instead opting to 'simulate' wrap. We shall not dwell on the details of Szabolcsi's wrap simulation method. The method is complicated, for example requiring that all NP and PP verb complements are obligatorily type-raised over their verb, and also that direct objects are lexically assigned types which are not only raised over verbs but which are then divided

[6] Similar proposals have been developed independently within the framework of Applicative Grammar by Declés *et al*, 1986.

over the raised types of indirect objects.[7] This treatment results in a rather complicated structure for English double object constructions.

We next consider some problems of Szabolcsi's account. We have already noted one problem: Szabolcsi's simulated wrapping method assigns a complicated and rather unnatural structure to double object constructions.

A second problem was discussed in Chapter 3 and relates to locality constraints on reflexivization. As Szabolcsi points out, the availability of operations such as type-raising and composition means that, without some further complication of the account, it is possible to derive ungrammatical examples such as **Mary$_i$ thinks that John loves herself$_i$*, as the following derivation illustrates (repeated from Chapter 3):

(5.31)
```
Mary   thinks   that    John    loves     herself
----   ------   ----    ----    -----   -----------
 np    vp/sp    sp/s    s/vp    vp/np   vp\(vp/np)
       -------------B   -------------B
           vp/s             s/np
           ---------------------B
                  vp/np
                  ------------------------------<
                               vp
 ---------------------------------<
                  s
```

Szabolcsi suggests that this problem can be avoided by requiring that English reflexives can only apply to functors that are "lexical in some sense". However, no definite statement of this method is given, and so we cannot evaluate whether an effective treatment of locality constraints on binding could be framed in these terms. As we saw in Chapter 3, Morrill (1989) suggests how locality contraints on binding may be handled in a modified version of Szabolcsi's account, by incorporating a modal treatment of linguistic boundaries.

Note that reflexives having semantics **W** are only of use for dealing with cases where the semantic association that results from reflexivization involves *consecutive* arguments of the verb. Szabolcsi points out that reflexives having a different meaning are required to deal with cases such as *John showed Mary himself*, where the arguments identified correspond to the first and third arguments of the verb type.[8] The reflexive semantics required corresponds to the lambda expression: $\lambda f \lambda x \lambda y.[f\,y\,x\,y]$. Although it is undesirable that reflexives should require different meanings to deal with different cases, this does not itself present a problem provided that reflexives require only a limited number of different lexical assignments (i.e. type/meaning pairings) to allow for all the different cases that occur. Kang (1988a) suggests that a problem will arise for applying Szabolcsi's account to binding in Korean. The Korean

[7] The instance of division used is slash-mixing, so that these types in effect lexically encode the use of mixed composition to give the reordering of complements that simulates wrap.

[8] Recall that, given Szabolcsi's adoption of a version of Dowty's categorial theory of grammatical relations, the ditransitive type she uses is *not* the type usually used with concatenative CGs.

reflexive *caki* allows for long distance reflexivization, as the following examples illustrate:

(5.32) a. John$_i$-i Mary-ka caki$_i$-lul cohahanta-ko sayngkakhanta.
John(nom) Mary(nom) REFL(acc) like–COMP think
'John$_i$ thinks that Mary likes him$_i$.'

b. Sue$_i$-ka John-i Mary-ka caki$_i$-lul cohahanta-ko sayngkakhanta-ko mitnunta.
Sue John Mary REFL like–COMP think–COMP believe
'Sue$_i$ believes that John thinks that Mary likes him$_i$.'

Kang suggests that to allow binding of this kind with Szabolcsi's account, the reflexive would need to be assigned a raised type that could apply to the result of composing together the verbs which specify the anaphor and antecedent arguments plus any verbs that are intermediate in the structure. Thus, a partial derivation for (5.32a) is shown in (5.33):

(5.33)

```
          caki-lul                 cohahanta-ko    sayngkakhanta
 -----------------------------    -------------   --------------
 (s\np\np)/(s\np\np\np)              s\np\np          s\np\s
                                   ------------------------------B^2
                                            s\np\np\np
 ---------------------------------------------------------------->
                       s\np\np
```

The first and third arguments of the composed function correspond to the anaphor and antecedent, respectively, so the reflexive would need semantics $\lambda f \lambda x \lambda y.[f\,y\,x\,y]$. A partial derivation for (5.32b) is:

(5.34)

```
             caki-lul                cohahanta-ko   sayngkakhanta-ko   mitnunta
 ------------------------------------ ------------ ------------------ ----------
 (s\np\np\np)/(s\np\np\np\np)            s\np\np          s\np\s         s\np\s
                                                         ----------------------B^2
                                                                s\np\np\s
                                         -----------------------------------B^2
                                                     s\np\np\np\np
 ----------------------------------------------------------------------------->
                           s\np\np\np
```

This time, it is the first and fourth arguments of the composed function that must be identified, requiring the reflexive to have semantics $\lambda f \lambda x \lambda y \lambda z.[f\,z\,x\,y\,z]$. A further level of embedding between reflexive and antecedent would require first and fifth arguments of a composed verb complex to be identified, and so on *ad infinitum*. Consequently, to handle Korean long distance reflexivization, Kang argues that Szabolcsi's analysis would be forced to claim that *caki* is infinitely ambiguous as to its type and meaning.

We have seen that Szabolcsi's account adopts a version of Dowty's theory as a basis for handling command effects on reflexivization. However, this move does not in itself guarantee that direction of binding will obey obliqueness. For example, Szabolcsi discusses a hypothetical reflexive *heself* with type (s/np)/(s\np/np) and semantics **W**, which would allow examples such as (5.35) in which the direction of binding violates the obliqueness constraint. The meaning assigned under this derivation would be (loves$'$ john$'$ john$'$).

(5.35)

```
Heself                loves      John
-----------------     --------   ----
(s/np)/(s\np/np)      s\np/np    np
-------------------------------->
          s/np
          ------------------------->
                     s
```

The problem here is that, of the two arguments that are identified semantically, the one that remains after the reflexive has combined with the verb (which corresponds to the 'binder' argument) is the *more oblique* argument.

It appears, therefore, that type assignment to reflexives must be constrained to ensure that binding obeys the command relation. Szabolcsi achieves this by stipulating that reflexive types should be of the form of *order preserving type raised NPs* (i.e. of the form x/(x\np) or x\(x/np)). This ensures that, when combined with a reflexive, a verb type is changed from x/np or x\np to x, so that the argument that is removed is the *outermost* argument of the verb. Consequently, the binder must be a later, and therefore less oblique, argument of the verb. However, even the use of this stipulation to ensure that binding obeys the obliqueness command requirement is considerably complicated by the fact that (as mentioned earlier) Szabolcsi's simulated wrap method requires that direct object NPs have types which are of the form of NPs that have undergone (order preserving) type raising *followed by* (disharmonic) division.

5.2 Binding and the Lambek calculus

The remainder of this chapter is concerned with the treatment of reflexivization in the Lambek framework. We consider two Lambek accounts: one developed by the author, and a second which originates in the proposals of Moortgat (1990b), and which is developed here to overcome some problems. We look at both of these accounts in some detail, since we consider both to be serious candidates for treating reflexivization within the Lambek framework. Despite the fact that the two accounts differ considerably, there is, as we shall see, little to choose between them in terms of empirical adequacy.

These accounts serve as a point of integration for the preceding chapters of the thesis. Thus, both depend on the account of word order developed in Chapter 4, using the modified version of Dowty's theory of grammatical relations as a basis for handling command effects on binding. Also, the polymodal treatment of linguistic boundaries is used as a basis for handling locality constraints on binding. The accounts are initially developed largely in respect of English data. Following this, we shall go on to develop an account of long distance reflexivization in Icelandic, which depends crucially on the polymodal treatment of linguistic boundaries.

5.2.1 A Lambek account of binding

We begin by considering the treatment of 'syntactic binding' of anaphors. Later, we will have something to say about discourse controlled anaphors. Our account is based on the use of a permutation operator which, unlike the permutation operators we have seen so far (that have S4-like behaviour), exhibits behaviour most closely related to that of necessity in the modal logic K. The account serves as a basis for introducing such K permutors and illustrating their use.

K-modal permutors and anaphor types

Under the account to be proposed, reflexives are assigned a fairly simple syntax and semantics. Their type is $\Box(\text{np}/\ominus\text{np})$, with identity semantics (i.e. λx.x). The basic form of this type should be familiar from the account of resumptive pronouns in Chapter 3. As with that account, the argument of the anaphor's functional type serves the purpose of allowing the argument position occupied by the anaphor to be bound (though this time anaphorically bound, instead of bound by a *wh*-item). This argument is distinguished, being marked with a binding specific permutation operator $\ominus$, which has K-like necessity behaviour.

The natural deduction rules for $\ominus$ are as in (5.36) (with (a) being given in before/after format). We assume that, as with the other modal rules, these rules do not affect the semantics associated with the types manipulated.

(5.36) (a)
$$\begin{array}{c} X_1 \cdots X_n \\ \vdots \\ X_0 \end{array} \quad \Rightarrow \quad \begin{array}{c} \dfrac{\ominus X_1}{X_1}\ominus E \quad \cdots \quad \dfrac{\ominus X_n}{X_n}\ominus E \\ \vdots \\ \dfrac{X_0}{\ominus X_0}\ominus I \end{array}$$

(b)
$$\dfrac{\ominus X \quad Y}{Y \quad \ominus X}\ominus P_1$$

(c)
$$\dfrac{X \quad \ominus Y}{\ominus Y \quad X}\ominus P_2$$

The permutation rules are essentially the same as for $\triangle$. However, in place of the separate introduction and elimination rules given for $\triangle$, we have a single rule which adds introduction and elimination steps in a single rule application. This rule should be read as follows: given a proof of X_0 from hypotheses $X_1,\ldots,X_n$, we may construct a proof of a type $\ominus X_0$ by substituting for each hypothesis X_i the subproof: $\dfrac{\ominus X_i}{X_i}\ominus E$

The corresponding sequent rule is easily stated as follows:

(5.37) $$\frac{X_1, X_2, \ldots, X_n \Rightarrow X_0}{\ominus X_1, \ominus X_2, \ldots, \ominus X_n \Rightarrow \ominus X_0}[\ominus LR]$$

(Note that the rule has been marked as $[\ominus LR]$ because it affects types occurring on both the left and right of the $\Rightarrow$ arrow.) The sequent permutation rules are also straightforward:

(5.38) $$\frac{\Gamma, \ominus x, y, \Lambda \Rightarrow z}{\Gamma, y, \ominus x, \Lambda \Rightarrow z}[\ominus P] \qquad \frac{\Gamma, x, \ominus y, \Lambda \Rightarrow z}{\Gamma, \ominus y, x, \Lambda \Rightarrow z}[\ominus P]$$

Recall that the S4-like character of $\triangle$ is revealed by the fact that the following three type transitions are valid:

(5.39) a. $\triangle(x/y) \Rightarrow \triangle x/\triangle y$

b. $\triangle x \Rightarrow x$

c. $\triangle x \Rightarrow \triangle\triangle x$

The K-like character of $\ominus$ is shown by the fact that, for the corresponding type-transitions, only the first is valid:

(5.40) a. $\ominus(x/y) \Rightarrow \ominus x/\ominus y$

b. * $\ominus x \Rightarrow x$

c. * $\ominus x \Rightarrow \ominus\ominus x$

As we have seen, the behaviour of $\triangle$ allows that $\triangle$-marked arguments may be both created and inherited in syntax, as illustrated by the following theorems:

(5.41) a. $x/y \Rightarrow x/\triangle y$

b. $x/y, y\backslash\triangle z \Rightarrow x\backslash\triangle z$

The different modal behaviour of $\ominus$ is such that $\ominus$-marked arguments can be inherited in syntax but cannot be created. Thus, the following is not admitted:

(5.42) * $x/y \Rightarrow x/\ominus y$

Since $\ominus$-marked arguments may not be introduced in syntax, they must originate lexically, with the types of anaphors and personal pronouns.

Although $\ominus$-marked arguments must originate lexically, this does not mean that they cannot be changed at all. For example, the type-transition in (5.43a), corresponding to the familiar operation of argument-lowering, is allowed. The 'derived inference rule' in (5.43b) shows a more general case which subsumes argument lowering. This flexibility is not superfluous (as we shall see in a later subsection on specifying possible antecedents).

(5.43) a. $x/\ominus(y/(y\backslash z)) \Rightarrow x/\ominus z$

b. *if* $y \Rightarrow z$ *then* $x/\ominus z \Rightarrow x/\ominus y$

The Reflexive Interpretation Rule

For syntactic binding, the cobinding of the reflexive argument (i.e. $\ominus$np in the type $\Box$(np/$\ominus$np)) with another argument is effected by a specific rule, which we term the Reflexive Interpretation Rule (RIR). As a first attempt, the RIR might be stated as in (5.44). (5.44a) shows just the syntax of the rule, whilst (5.44b) shows the rule with lambda semantics.

(5.44) (a)

$$\begin{array}{c}[\ominus \mathrm{X}]^{\mathrm{i}} \\ \vdots \\ \dfrac{\mathrm{Y}}{\mathrm{Y}}\mathrm{RIR}^{\mathrm{i}}\end{array} \quad \text{where Y is A}\backslash\text{X or A/X}$$

(b)

$$\begin{array}{c}[\ominus \mathrm{X}\colon v]^{\mathrm{i}} \\ \vdots \\ \dfrac{\mathrm{Y}\colon f}{\mathrm{Y}\colon \lambda v[f\,v]}\mathrm{RIR}^{\mathrm{i}}\end{array} \quad \text{where Y is A}\backslash\text{X or A/X}$$

The RIR does not change the conclusion type of the proof to which it is applied, but does affect the proof's semantics. Note that the hypothesis of type $\ominus$X that is discharged must correspond to the argument of an reflexive anaphor (or pronoun), since only these types introduce $\ominus$ operators. In the semantics of the result constituent, the lambda operator will bind two instances of the variable v, since v must already appear as a subterm of f. Thus, the reflexive argument is caused to be co-bound with another argument.

This idea is perhaps best presented with an example. For simplicity, we begin by assuming a standard CG treatment of English word order (i.e. without boundary modalities, and not using the word order treatment presented in Chapter 4).

(5.45)

$$\dfrac{\dfrac{\text{John}}{\text{np}} \qquad \dfrac{\dfrac{\dfrac{\text{loves}}{\text{s}\backslash\text{np/np}} \qquad \dfrac{\dfrac{\text{himself}}{\text{np}/\ominus\text{np}} \quad [\ominus\text{np}]^{\mathrm{i}}}{\text{np}}/\text{E}}{\text{s}\backslash\text{np}}/\text{E}}{\text{s}\backslash\text{np}}\text{RIR}^{\mathrm{i}}}{\text{s}}\backslash\text{E}$$

This proof assigns the meaning (5.46a), which, since himself′ corresponds to the identity semantics λx.x, is equivalent the expression in (5.46b).

(5.46) a. λv.(loves′ (himself ′ v) v) john′

b. (loves′ john′ john′)

Reflexivization and agreement

The appropriate treatment of agreement phenomena remains a matter of debate, with some authors arguing against standard syntactic treatments of agreement and in favour of semantically based treatments (e.g. Lapointe, 1980; Hoeksema, 1983). We do not wish to enter into this debate. We assume (without great commitment) a treatment of agreement in binding based on the use of features marked on atomic types. In the type $\Box(\text{np}/\ominus\text{np})$ for *himself*, both the value and argument np types are featurally specified to be 3rd person singular masculine. The operation of the RIR, in equating (unifying) the binder argument with the assumption that is discharged, not only requires that the binder have the major category specified by the reflexive type (i.e. np), but also causes these agreement features to be instantiated on the binder argument. Thus, in the vp type assigned to the string *loves himself*, the np argument of the functor will be instantiated with agreement features for masculine 3rd person singular through the operation of the RIR, avoiding overgenerations such as **Mary$_i$ loves himself$_i$*. Note, however, that not all features on the value and argument of the reflexive type will be the same since we do not want all the features of the value subtype (which determines the reflexive's general distribution, aside from binding requirements) to become instantiated on the binder, e.g. although the value np subtype is marked for objective case, this feature must not be marked on the argument type since it would block subject antecedents.

5.2.2 Obliqueness and command effects

For introducing the RIR, we have reverted to a standard categorial treatment of word order. However, such an account would fail to appropriately handle a range of examples, in particular, cases for which Montague Grammar uses wrapping operations. For example, such an account would predict precisely the opposite of the observed possibilities of binding between the two objects of double object constructions.

(5.47) a. John showed Mary$_i$ herself$_i$ (in the mirror)

b. *John showed herself$_i$ Mary$_i$ (in the mirror)

This is because the RIR requires that the reflexive occurs as a subpart of the constituent to which the RIR is applied, the binder being the next argument sought by the functional type of this constituent. The type assigned to ditransitives in standard concatenative CGs is one in which the direct object is sought before the second object, and so an anaphor appearing as the direct object could be bound by the second object, but not vice versa. (5.48) shows a proof for the VP of (5.47b).

(5.48)

*showed: vp/np/np herself: np/⊖np $[\ominus np]^i$ Mary: np

np/⊖np $[\ominus np]^i$ ⟹ np (/E)

vp/np/np np ⟹ vp/np (/E)

vp/np ⟹ vp/np (RIR^i)

vp/np np ⟹ vp (/E)

This problem can be avoided if we adopt the account of word order presented in Chapter 4, and restate the RIR so that it applies to proofs of functional types whose principal connective is one of the primitive subcategorization slashes, as follows:

(5.49) (a) $[\ominus X]^i$ ⋮ Y ⟹ Y (RIR^i), where Y is A⦸X or A⊘X

(b) $[\ominus X\colon v]^i$ ⋮ $Y\colon f$ ⟹ $Y\colon \lambda v[f\,v]$ (RIR^i), where Y is A⦸X or A⊘X

This move immediately gives an account where binding is subject to an obliqueness based command restriction, like Pollard and Sag's *o-command*. This is because any type X⦸Y or X⊘Y must correspond to a 'natural projection' of some lexical functor, such that the argument Y must be less oblique than any of the arguments with which the functor has already combined. Since the anaphor must combine with a functor before the application of the RIR, it must appear as, or as a subpart of, a more oblique argument of the head that subcategorizes for the binder argument.

Example (5.50) shows a proof for *John loves himself* under this view of word order.[9] This proof assigns the meaning (5.51a) (where the verb *loves* has semantics λf.(f loves′), with loves′ the meaning of the prelocation verb type), which is equivalent to (5.51b).

(5.50)

John: □np ⟹ np (□E)

loves: □(s⦸np/(s⦸np/△(s⦸np⦸np))) ⟹ s⦸np/(s⦸np/△(s⦸np⦸np)) (□E)

himself: □(np/⊖np) ⟹ np/⊖np (□E)

np/⊖np $[\ominus np]^i$ ⟹ np (/E)

$[\triangle(s⦸np⦸np)]^j$ ⟹ s⦸np⦸np (△E)

np s⦸np⦸np ⟹ s⦸np (⦸E)

s⦸np ⟹ s⦸np (RIR^i)

s⦸np ⟹ s⦸np/△(s⦸np⦸np) ($/I^j$)

s⦸np/(s⦸np/△(s⦸np⦸np)) s⦸np/△(s⦸np⦸np) ⟹ s⦸np (/E)

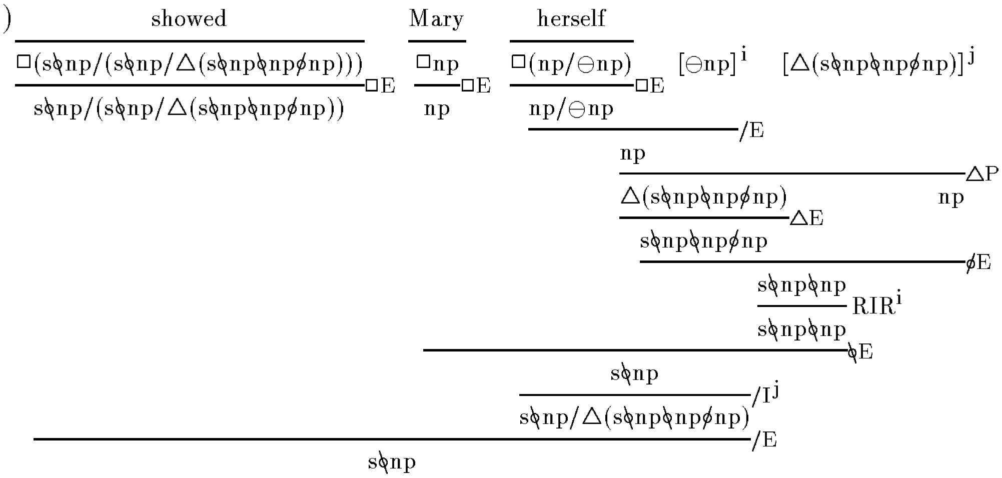
np s⦸np ⟹ s (⦸E)

(5.51) a. λf.(f loves′) λg.λv.(g (himself′ v) v) john′

b. (loves′ john′ john′)

[9] We ignore for the moment subject boundary modalities marked on verb types.

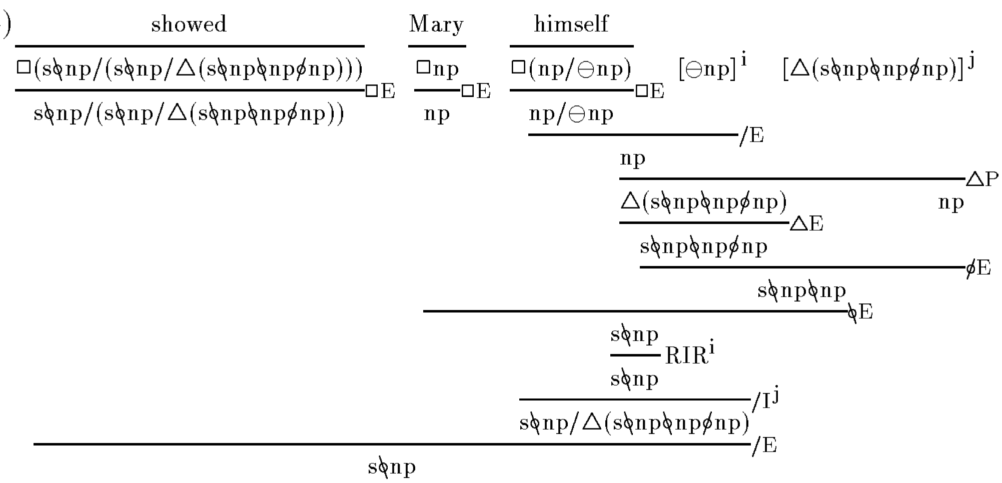
The following is a proof for the VP of example (5.47a), with semantics as in (5.53). No proof is possible for the ungrammatical (5.47b).

(5.52)


(5.53) a. (λf.f showed′)(λg.(λv.g (himself′ v) v) mary′)

b. showed′ mary′ mary′

In the examples so far, the binder and anaphor arguments are consecutive in the functional structure of the verb. (5.54) shows binding of a second object by a subject, i.e. between the verb's first and third arguments. The meaning assigned is as in (5.55).

(5.54)


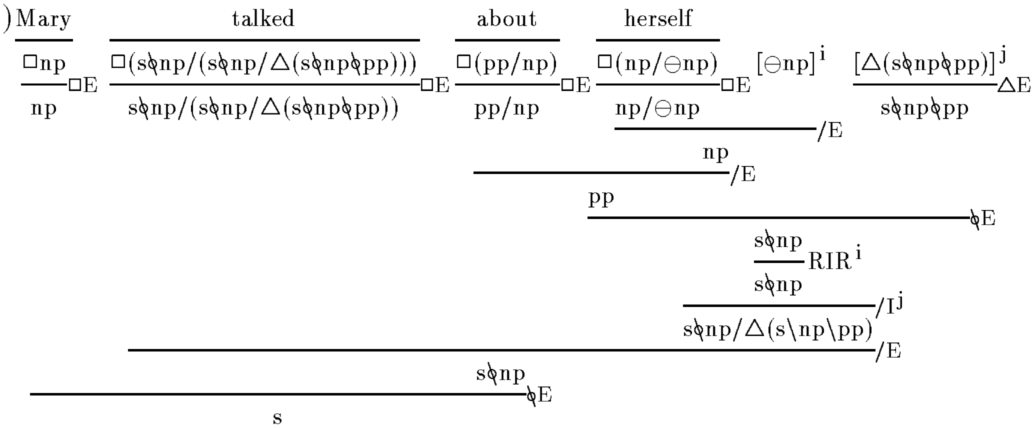
(5.55) a. λf.(f showed′) λg.λv.(g (himself′ v) mary′ v)

b. λv.showed′ v mary′ v

This example has an alternative proof, in which the RIR is applied at a later stage to a constituent which includes the lexical verb (rather than just the hypothesized prelocation verb type that is later discharged), which assigns a meaning equivalent to (5.54).

As noted above, this account does not require an anaphor to occur as a direct argument of the lexical head that subcategorizes for the binder. Instead, an anaphor may also occur as a *sub-part of* such an argument. For example, a syntactically bound anaphor may occur within a PP as in (5.56) (without needing to 'pied-pipe' the anaphor, as required in Szabolcsi's account (Szabolcsi, 1987a)).

(5.56)



Thus, although syntactic binding requires an o-commanding binder, o-command need not be *local* (i.e. with the reflexive and binder occurring as codependents of a head).

We next briefly look at a case which is an apparent contradiction to the command requirement on binding, and consider how it may be handled in a way consistent with o-command. This problem arises in respect of examples such as:

(5.57) Mary spoke to John_i about himself_i

In (5.57), the apparent binder for the pronoun *John* appears within a PP, and so does not c-command the reflexive under any obvious phrase structure that might be assigned, e.g.

(5.58) $[_{\text{vp}}$ spoke $[_{\text{pp}}$ to John$]$ $[_{\text{pp}}$ about himself$]]$

This example presents a similar problem for o-command, i.e. even assuming the *to*-phrase to be a less oblique dependent than the *about*-phrase, the NP *John* still doesn't o-command the reflexive since it is not itself a direct dependent of the verb. In the GB framework, it has been suggested that such examples involve *reanalysis*, whereby *spoke to* is taken to be reanalysed as a complex verb which takes the apparent complement of *to* as its direct object, so that instead of (5.58) we would have:

(5.59) $[_{\text{vp}}$ $[_{\text{v}}$ spoke to$]$ John $[_{\text{pp}}$ about himself$]]$

Szabolcsi (1987a) has suggested that a similar solution be adopted in CG treatment, i.e. with there being a lexical entry *spoke to* which is assigned a type that directly subcategorizes for a direct object NP. We could also adopt this solution. However, an alternative solution is suggested by the account of Pollard and Sag (1990). Pollard and Sag point out that examples such as (5.57) seem only to arise in cases where the binder NP appears within an argument (as opposed to adjunct) PP, where the preposition fulfils only a "case-marking" role. Pollard and Sag argue that case-marking prepositions should be treated as semantically vacuous, so that if the PP, which does o-command the reflexive, is allowed to be the binder, this yields the same semantic result as if the reflexive had been bound by the preposition's object NP. (In Pollard and Sag's account, case-marking prepositions are taken to 'inherit' the referentiality of their NP object. Hence, making a reflexive coreferential with some PP whose head is a case-marking preposition results in coreferentiality of the reflexive with the preposition's object.)

Within our approach, we might adopt a related solution. Recall the distinction between argument and adjunct PPs. In CG, this distinction is commonly reflected in the type assignments to prepositions so that adjunct PPs have types (ignoring modality) such as $\mathrm{n}\backslash\mathrm{n}$ and $\mathrm{vp}\backslash\mathrm{vp}$, and only argument PPs have type pp. Hence, case-marking prepositions have type pp/np, and can be given identity semantics so that the meaning of the resulting PP is identical to that of its NP object. Consider again the type we give to anaphors:

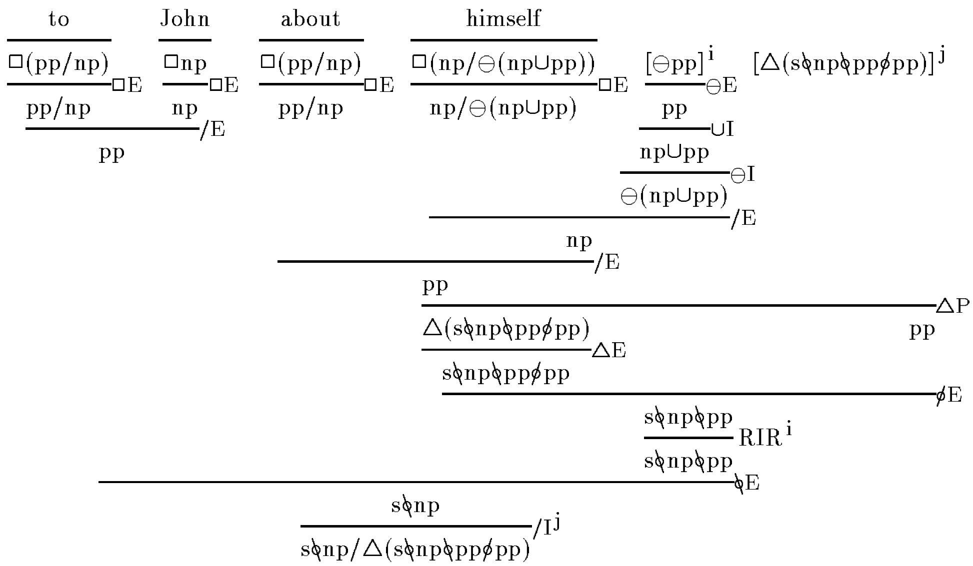
$$\Box(\mathrm{np}/\ominus\mathrm{np})$$

We can see that, ignoring modality, this type is essentially a function from np to np. That the value of this function is type np has the consequence that the anaphor exhibits the distributional behaviour of a noun phrase (restricted by requirements from binding). That the argument of this function is np, on the other hand, has a different consequence — namely that the *binder* for the anaphor must be of type np. Clearly then, we could allow anaphors to have (argument) PP binders by assigning them a further type $\Box(\mathrm{np}/\ominus\mathrm{pp})$. However, the availability of the boolean type-forming operators, discussed in Chapter 2, means that we can instead assign a single type to anaphors

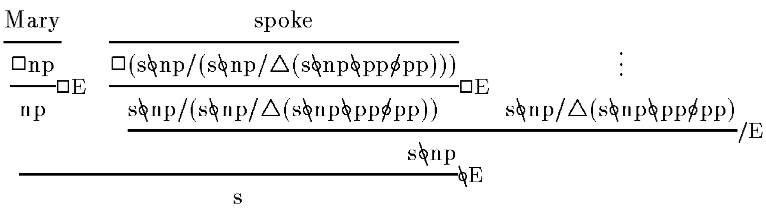
$$\Box(\mathrm{np}/\ominus(\mathrm{np}\cup\mathrm{pp}))$$

which allows them to be bound by both NP and PP antecedents. This is illustrated in (5.60) in a proof for (5.57). The proof has been split into two parts because of its size.

(5.60)(a)



(b)



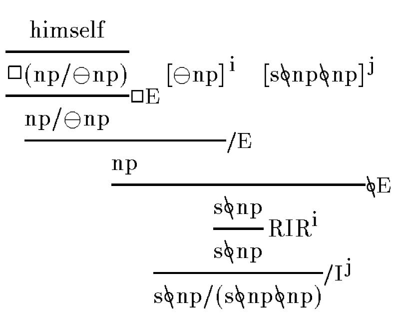
The meaning assigned under this proof is shown in (5.61a), which, since *himself* and both case-marking prepositions have identity semantics, is equivalent to (5.61b). (Again, spoke′ corresponds to the meaning of the prelocation verb type.)

(5.61) a. $(\lambda f.f\ \text{spoke}')\ (\lambda g.(\lambda v.g\,(\text{about}'\,(\text{himself}'\ v))\ v\ )\ (\text{to}'\,\text{john}'))\ \text{mary}'$

b. spoke′ john′ john′ mary′

It is interesting to note that raised types having 'duplicator' semantics, similar to those required for Szabolcsi's account, can be derived in syntax from the simple reflexive type we have specified, although containing primitive subcategorization slashes as required by the RIR. For example:

(5.62)

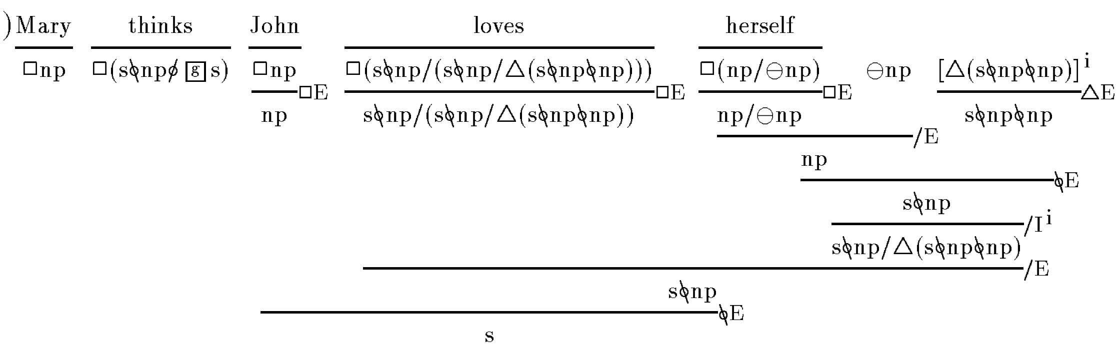


This proof assigns the meaning $\lambda f \lambda x.[f\,x\,x]$, equivalent to the the combinator **W**. Note that the calculus only allows raised types to be derived which do not give command violations.

The two main kinds of constraint on binding relations are *command* constraints (which we take to be obliqueness based) and what we have called *domain* constraints, i.e. conditions to the effect that some anaphor or pronoun must find its antecedent either inside or outside of some domain. In this section, we focus largely on how conditions that some anaphor must find its antecedent *inside* some domain can be handled, although we shall comment briefly on disjoint reference phenomena later on.

Our treatment of locality constraints on binding is based on the polymodal system for linguistic boundaries described in Chapter 3. That modal operators allow a treatment of locality constraints on binding was first suggested by Morrill (1989), in relation to Szabolcsi's account of binding (Szabolcsi, 1987a) (as we saw in Chapter 3, and earlier this chapter).

For our account, the appearance of an anaphor in an embedded modal domain that does not contain the binder is also blocked, given the anaphor type we have assigned. For example, consider the following attempt to bind a reflexive from outside the embedded modal domain in which it appears:

(5.63)


To complete this proof, the string *John loves herself* must be derived as a modal clause. The proof includes an undischarged assumption corresponding to the 'argument' of the reflexive. Since this assumption is not □-modal, the side condition on the box introduction rule [□I] is not met, and so the proof cannot be continued. Thus, with this reflexive type, a reflexive occurring within a modal domain must also be bound within this domain.

However, the possibility arises for assigning types to anaphors and other 'bindable' pronouns that allow binding outside the immediate domain. This can be done in essentially the same way that types were assigned to resumptive pronouns in Chapter 3 which allowed them to 'override' island constraints. Consider the following possible anaphor type, and the subproof (5.64) based on it:

$$\Box(\boxed{\alpha}\,np/\ominus np)$$

(5.64)

$$\dfrac{\dfrac{\Box(\boxed{\alpha}\,np/\ominus np)}{\boxed{\alpha}\,np/\ominus np}\Box E \qquad \ominus np}{\boxed{\alpha}\,np}/E$$

Here, the pronoun has combined with the additional hypothesis corresponding to its argument, giving an (independent) proof of the modal type $\boxed{\alpha}$np. Such a subproof *could* appear embedded under a modal domain (say, $\boxed{\beta}$), just in case the modality $\boxed{\alpha}$ licenses box introduction for the relevant domain modality (i.e. if $\mathcal{L}_\alpha \subseteq \mathcal{L}_\beta$). The $\ominus$ assumption could then be discharged at a later stage so that the pronoun was bound from outside of its immediate modal domain.

We suggest that this framework provides a basis for dealing with cases where the binding of an anaphor extends beyond the minimal modal domain in which it occurs. In the latter part of this chapter, we look at the phenomenon of long distance reflexivization in Icelandic, and show how it can be addressed in this framework.

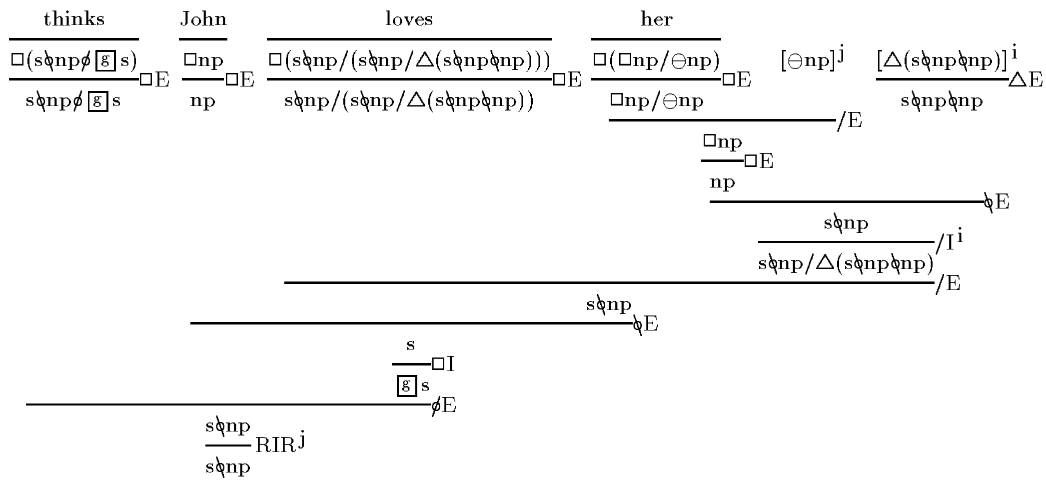
We shall here consider only the case of personal pronouns. Recall that according to Reinhart's account, personal pronouns can exhibit syntactic binding provided that the command condition is satisfied, with disjoint reference behaviour being explained pragmatically. By this view, the *grammar* should place *no* domain constraint on pronoun binding. We can allow such behaviour by assigning personal pronouns a type:

$$\Box(\Box np/\ominus np)$$

The value np of this type is marked with the 'strong' modality $\Box$ (whose defining language $\mathcal{L}_\emptyset$ is assumed to be a subset of all the other defining languages used). This has the consequence that pronouns bearing this type may be syntactically bound from outside of *any* modal domain. Reinhart (1983) assumes that quantified NPs, being non-referential, must syntactically bind a pronoun. Thus, the quantified NP in (5.65) must syntactically bind the pronoun in the embedded clause. A proof for the VP of this example is shown in (5.66).

(5.65) Every girl$_i$ thinks John loves her$_i$.

(5.66)



This proof assigns the reading (5.67a), equivalent to (5.67b) (again, loves′ is the meaning of the verb's prelocation type, the located verb having meaning (λg.g loves′)).

(5.67) a. λv.thinks′ ((λg.g loves′) (λf.f (her′ v)) john′) v

b. λv.thinks′ (loves′ v john′) v

We follow Dowty (1980) and Reinhart (1983) in assuming that the disjoint reference behaviour exemplified in (5.68a) should receive a pragmatic explanation. By this view, it is the availability of the unambiguous statement in (5.68b) which makes the ambiguous statement in (5.68a) infelicitous.

(5.68) a. *John$_i$ loves him$_i$

b. John$_i$ loves himself$_i$

A note on specifying possible binders

The fact that the anaphor (5.69a) has argument type np (strictly $\ominus$np) has the effect of restricting possible binders to be just NPs, whilst the reflexive type (5.69b) allows for both NP and PP binders.

(5.69) a. $\Box(\mathrm{np}/\ominus\mathrm{np})$

b. $\Box(\mathrm{np}/\ominus(\mathrm{np}\cup\mathrm{pp}))$

This picture is slightly complicated by the fact that the calculus allows type transitions that modify $\ominus$-marked arguments, transitions of the 'argument lowering' pattern which are of the general form shown in (5.70):

(5.70) *if* $\mathrm{y} \Rightarrow \mathrm{z}$ *then* $\mathrm{x}/\ominus\mathrm{z} \Rightarrow \mathrm{x}/\ominus\mathrm{y}$

This means that for any bindable constituent with type $x/\ominus y$, the binder does not have to have type y, but can in fact be any type z such that $z \Rightarrow y$. This additional flexibility is not superfluous. For example, recall that in Chapter 3 we argued that subjects be specified as modal domains to handle Subject Condition island behaviour (a detail omitted so far in this chapter to simplify presentation), so that a transitive verb has type:

$$\Box(s\backslash\boxed{k}np/(s\backslash\boxed{k}np/\triangle(s\backslash\boxed{k}np\backslash np)))$$

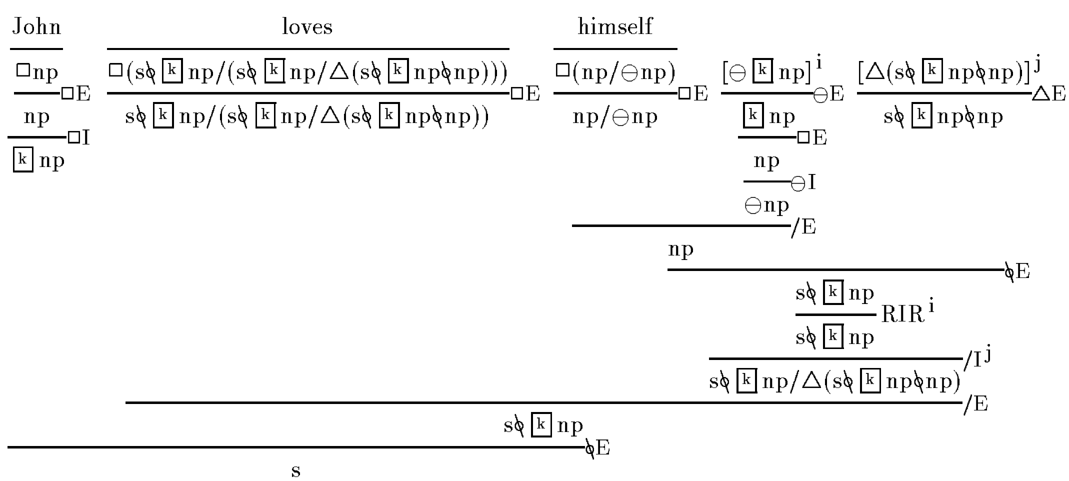
Given the flexibility allowed with respect to potential binders, the presence of the subject boundary modality does not present a problem, and the same anaphor type $\Box(np/\ominus np)$ can handle subject binding. This should be clear since the type transition shown in (5.71a) is valid (which follows given that the transition in (5.71b) is).

(5.71) a. $\Box(np/\ominus np) \Rightarrow \Box(np/\ominus\boxed{k}np)$

b. $\boxed{k}np \Rightarrow np$

The proof in (5.73) illustrates how subject-modality is handled in a simple example.

(5.72)



Subject orientation

There is considerable variation in the binding behaviour exhibited by different languages, and even by different anaphors in the same language. One dimension of variation is requirements on the antecedent. For example, some languages exhibit only *subject antecedent* reflexization, where only subjects are possible antecedents. Anaphors that only allow a subject antecedent are referred to as *subject oriented*. We briefly consider how subject orientation may be handled in the present framework.

One possibility is suggested by the distinct modality we have used in handling the island behaviour of subjects in English. English reflexives do not exhibit a subject orientation restriction. However, let us assume for the moment that the subjects of languages where subject orientation is observed are also marked by a distinct modality (we will use $\boxed{k}$ for this throughout). Consider the possible reflexive a type:

$$\Box(np/\ominus\boxed{k}np)$$

In the light of the discussion of the preceding section, it should be clear that some o-commanding argument of type X is a potential binder for this reflexive just in case:

$$X \Rightarrow \boxed{k}np$$

This is possible if X is a type $\boxed{\alpha}$np such that $\mathcal{L}_\alpha \subseteq \mathcal{L}_\mathbf{k}$. If only subject arguments bear such a type, then only subjects may bind the reflexive.

Such a treatment of subject orientation seems unsatisfactory in a number of regards. Firstly, whether we take the subjects of some language to be marked with a distinct modality is something that should be independently motivated. This should not be done as a trick solely to allow subjects to be picked out for some syntactic process. Furthermore, such a method would allow quite arbitrary orientations to be stipulated. Also, this means of identifying subjects seems both inappropriate and redundant given that the approach already incorporates an account of grammatical relations.

An alternative possibility involves constraining the operation of the RIR in respect of the types of constituents to which the rule may be applied. The version of the RIR stated in (5.73) only allows subject binders, since it only applies to subproofs of VP types.

(5.73)
$$\begin{array}{l}[\ominus np]^i \\ \vdots \\ \dfrac{vp}{vp}\text{RIR}^i\end{array}$$

This means of identifying the subject clearly exploits, rather than ignores, the treatment of grammatical relations we have adopted. To deal with languages which have different reflexive forms that exhibit different antecedent requirements we might need to adopt a number of reflexive operators that have the same modal and permutational inferences, but which are subject to differently restricted versions of the RIR. For example, we might have distinct operators $\{\ominus, \otimes, \ldots\}$, where the RIR for $\ominus$ allows any antecedent, that for $\otimes$ only subject antecedents, and so on. Such an approach should allow us to deal with the occurrence of anaphors that exhibit different antecedent requirements, but it is clearly unsatisfactory that we should need to do this by multiplying the operators that the calculus includes.

Possessives

We next consider the treatment of possessive constructions. In Chapter 4, we argued for an account of NPs in which nouns are viewed as heads which subcategorize for their specifier (either determiner or possessive NP) and which may need to undergo head location. Before we go on to consider this view in relation to binding and extraction, we will look at a standard categorial treatment of possessive constructions and consider some problems it has. We begin by noting some relevant facts about binding and extraction.

It is well known that certain nouns, typically called *picture nouns*, allow NP extraction from a PP complement. The following examples illustrate the asymmetry observed for cases involving picture nouns and non-picture nouns:

(5.74) a. Who did you see a picture of?

b. *Which town did you meet a friend from?

This asymmetry has been explained in terms of an argument/adjunct distinction between the two PPs, picture nouns being assumed to subcategorize for their PP complement. The presence of a possessive NP specifier blocks such extraction:

(5.75) *Who did you like Mary's picture of?

As discussed earlier in the chapter, the presence of a possessive NP also prevents a picture noun anaphor being bound by an antecedent outside the NP:

(5.76) a. John_i likes every picture of himself_i

b. *John_i likes Mary's picture of himself_i

A standard categorial treatment of possessive constructions involves treating *'s* as a lexical entry which receives a type:

$$\mathrm{np/n\backslash np}$$

i.e. a function from np (the possessor) to the (standard) determiner type np/n. A problem arises for using such a type with the present account of binding (or any other o-command based approach), which is that the possessor NP does not o-command the noun, and so should not be able to bind a picture noun anaphor, as in *$Mary_i$'s picture of $herself_i$*. If we were to reverse the argument order for the genitive *'s*, as in:

$$\mathrm{np}\backslash\!\!\!\circ\,\mathrm{np}/\!\!\!\circ\,\mathrm{n}$$

the possessor NP would then o-command the noun and its complement, so that examples such as *$Mary_i$'s picture of $herself_i$* could be derived.

Consider next how boundary modalities could be added to get the observed locality effects. We saw above that the presence of the possessive NP blocks *wh*-movement from the picture noun's complement, and also binding of a picture noun anaphor by the subject of the dominating verb. This behaviour could be handled by having the genitive *'s* specify its noun complement to be a modal domain, as in:

$$\mathrm{np}\obslash\mathrm{np}\oslash\boxed{t}\,\mathrm{n}$$

Such a type, however, would also block binding by the possessor NP. Amending the anaphor type so that this boundary did not block binding would also make binding by the matrix subject possible. Thus, it appears that this treatment of possessives is incompatible with an assignment of boundaries that appropriately characterizes the observed locality behaviour.

Recall that under our view of NP structure, nouns are taken to subcategorize for their specifier, which is assumed to be the noun's least oblique complement. Furthermore, we assume that the 'ordering principles' governing the assignment of directionality to prelocation type arguments are the same for nouns as for verbs. Thus, a noun such as *picture* has a type (5.77a) (in which np_p stands for np[+possessive]), giving rise to the final type assignment (5.77b). We assume that the genitive 's has the type (5.77c), with identity semantics, which combines with the possessor NP, serving to mark it as possessive, i.e. [+poss] (which also, by the way, prevents iteration of *'s* as in **John's's*).

(5.77) a. $\mathrm{np}\obslash\mathrm{np}_p\obslash\mathrm{pp}$

b. $\Box(\mathrm{np}\obslash\mathrm{np}_p/(\mathrm{np}\obslash\mathrm{np}_p/\triangle(\mathrm{np}\obslash\mathrm{np}_p\obslash\mathrm{pp})))$

c. $\mathrm{np}_p\backslash\mathrm{np}$

How might this approach accommodate the relevant locality effects on binding and extraction? Since it is the noun itself that determines whether its specifier is a determiner or possessive NP, it seems that the observed asymmetries may be handled in terms of differences in the modal boundaries specified on the noun form that seeks a possessive NP from that which seeks a determiner. One possibility is that the noun type which seeks a possessive specifies its PP complement as a modal domain, as in the following type:

$$\Box(\mathrm{np}\obslash\mathrm{np}_p/(\mathrm{np}\obslash\mathrm{np}_p/\triangle(\mathrm{np}\obslash\mathrm{np}_p\obslash\boxed{t}\,\mathrm{pp})))$$

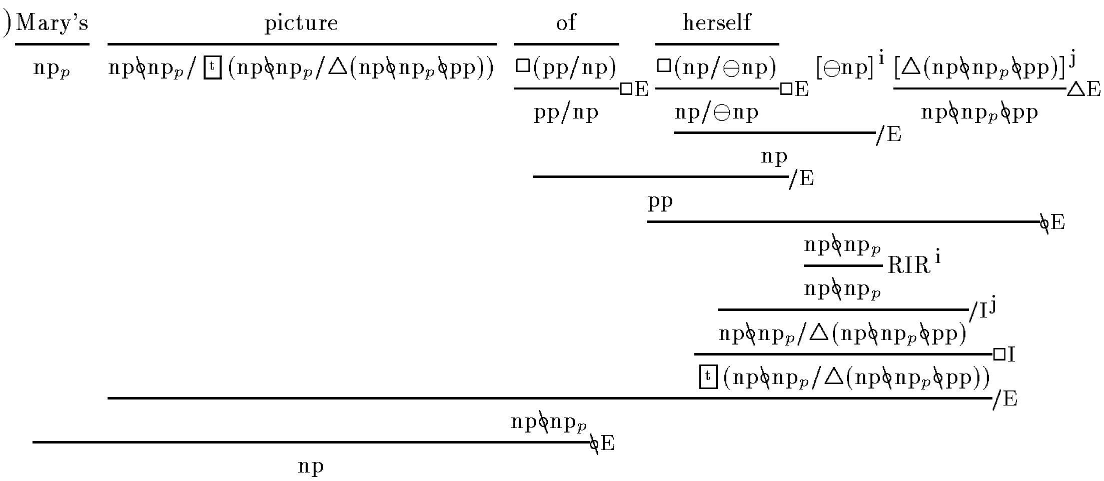
Such a boundary would also block binding of the possessive NP, giving precisely the same problem as before. However, the raised type of the located noun also allows a further possibility for specifying a boundary, as follows:

$$\Box(\mathrm{np}\obslash\mathrm{np}_p/\boxed{t}\,(\mathrm{np}\obslash\mathrm{np}_p/\triangle(\mathrm{np}\obslash\mathrm{np}_p\obslash\mathrm{pp})))$$

Since this modality encloses the domain in which the PP complement is constructed, extraction of the complement of the preposition to a position outside the entire NP is blocked. Binding of a picture noun anaphor from outside the NP is similarly prevented.[10] This type, however, does not block binding of a picture noun anaphor by the possessive NP. This is

[10] This link between the island status of possessed NPs and the unavailability of external antecedents for syntactic binding brings to mind the fact that these effects also receive a common explanation in transformational theory, in terms of the Specified Subject Condition. However, the two explanations are clearly very different in nature.

because binding may be made with respect to the possessive NP argument of the *prelocation* type of the noun, which appears as a subtype of the located type, and falls within the specified modal domain. Thus, *Mary$_i$'s picture of herself$_i$* can be derived as follows:

(5.78)


A note on strict and sloppy readings with VP-ellipsis

In this section, we briefly note some possibilities that arise for our account in respect of VP-ellipsis. The following example illustrates strict and sloppy readings:

(5.79) John$_i$ mentioned his$_i$ merits before Mary did
  a. STRICT: 'before Mary mentioned his merits'
  b. SLOPPY: 'before Mary mentioned her merits'

Gawron and Peters (1988, cited by Szabolcsi, 1989) point out that the following example presents a problem for Reinhart's account of strict readings with VP-ellipsis, which is based on 'accidental coreference' of pronoun and antecedent:

(5.80) Every man$_i$ mentioned his$_i$ merits before Mary did.

Reinhart's account predicts that the strict reading is only available with a referential antecedent, but the antecedent in (5.80) is quantificational not referential. Szabolcsi (1989) points out that similar examples can be constructed with a reflexive, e.g:

(5.81) Every student$_i$ corrected himself$_i$ before the teacher did.
  a. STRICT: 'before the teacher corrected him'
  b. SLOPPY: 'before the teacher corrected himself'

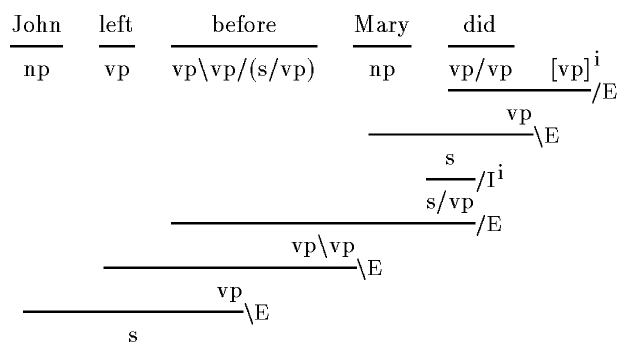
We will show how the strict and sloppy readings for such examples can be derived in the present framework. Since we do not want to address the general problem of VP-ellipsis, we here assume a simple treatment based on the type assigned to the subordinator *before*. We assume that *before* has a VP modifier type vp\vp/s to allow for non-ellipsis cases (ignoring modalities, which are not relevant here). Given this type, the sentence *John left before Mary arrived* receives the following interpretation:

before′ (arrived′ mary′) left′ john′

VP-ellipsis is allowed by assigning *before* an additional type which, instead of seeking a sentential complement, seeks a 'sentence missing a VP' (i.e. a complement of type s/vp), and whose semantics 'duplicates' the meaning of the modified VP in place of the missing VP. Thus, we have the type/meaning assignment in (5.82), which allows the following proof for *John left before Mary did*, with interpretation as in (5.84) (we assume *did* to be semantically vacuous).

(5.82) before: vp\vp/(s/vp) $\lambda x \lambda y.\text{before}'(x\ y)\ y$

(5.83)



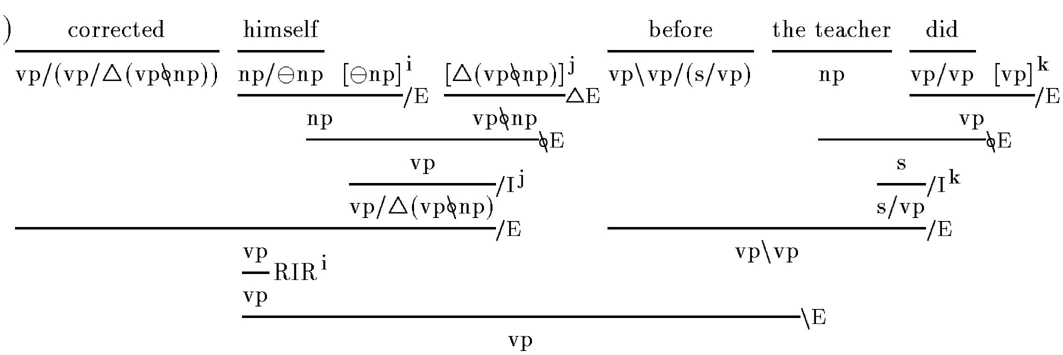
(5.84) a. $(\lambda x \lambda y.\text{before}'\ (x\ y)\ y)\ (\lambda v.\text{did}'\ v\ \text{mary}')\ \text{left}'\ \text{john}'$

b. before′ (left′ mary′) left′ john′

Note that since s/vp can also be derived from np, the lexical type (5.82) for *before* also allows examples without *did* to be derived, e.g. *John left before Mary.*

Given this treatment of VP-ellipsis, the strict and sloppy readings for the examples in (5.81) follow automatically, the difference stemming from the stage in the proof that the RIR is applied. The following proof of the VP of example (5.81) gives the sloppy reading:

(5.85)



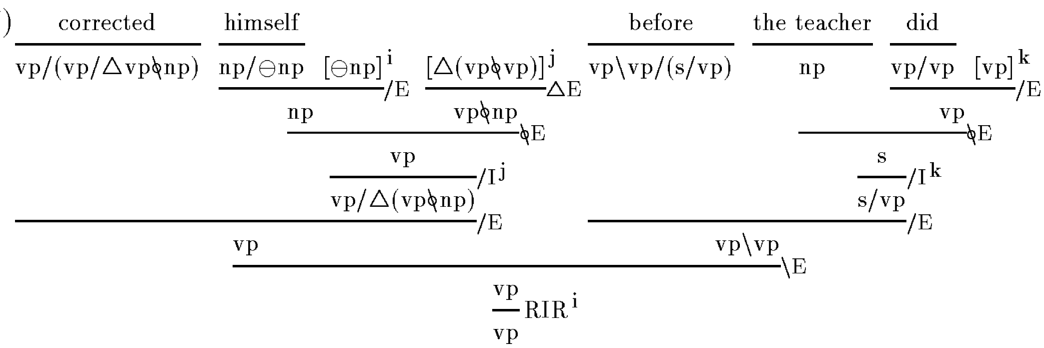
Note that the RIR is applied to the subproof of the VP *corrected himself,* giving rise to a 'reflexive semantic predicate'. This is 'duplicated' by the semantics of *before*, resulting in the sloppy reading. (5.85) assigns the meaning shown in (5.86), where corrected* abbreviates (λf.f corrected$'$), the located verb's meaning (corrected$'$ being the prelocation type's meaning):

(5.86) a. $(\lambda x \lambda y.\mathrm{before}'\,(x\,y)\,y)(\lambda v.\mathrm{did}'\,v\ \ \mathrm{the\text{-}teacher}')(\lambda z.\mathrm{corrected}^{*}(\lambda g.g\,(\mathrm{himself}'\,z)\,z)$

b. $\mathrm{before}'\,(\mathrm{corrected}'\,\mathrm{the\text{-}teacher}'\,\mathrm{the\text{-}teacher}')\ (\lambda z.\mathrm{corrected}'\,z\ z)$

The proof in (5.87) assigns the strict reading for (5.81). In this, the RIR is applied to the subproof of the VP *corrected himself before the teacher did*. Before the application of the RIR, the variable semantics corresponding to the argument of the reflexive is not bound, and is 'duplicated' by the semantics of *before*. These two variable occurrences are only then bound by the operation of the RIR, so that the matrix subject is the antecedent for both, giving the strict reading. (5.87) assigns the meaning (5.88).

(5.87)



(5.88) a. $\lambda z.(\lambda x.\lambda y.\mathrm{before}'\,(x\,y)\,y)(\lambda v.\mathrm{did}'\,v\ \ \mathrm{the\text{-}teacher}')(\mathrm{corrected}^{*}(\lambda g.g\,(\mathrm{himself}'\,z)))$

b. $\lambda z.\mathrm{before}'\,(\mathrm{corrected}'\,z\ \mathrm{the\text{-}teacher}')\ (\mathrm{corrected}'\,z)\ z$

Discourse behaviour of reflexives

Earlier in this chapter, we discussed the occurrence of anaphors in English which are not subject to the usual command and domain constraints, what Pollard and Sag (1990) call *exempt* anaphors. Some examples are shown in (5.12). The referential behaviour of such exempt anaphors appears to be subject only to discourse/pragmatic considerations, such as the 'point of view' constraint discussed. We argued for a view of the data in which the possibilities of syntactic and discourse binding overlap.

We next address the treatment of exempt anaphors, focussing in particular on picture noun reflexives and possessive reciprocals. We take the discourse/pragmatic constraints that limit the referential behaviour of exempt anaphors to be outside the scope of the grammar, but (like Pollard and Sag) we assume that the grammar *should* account for the syntactic conditions under which exempt anaphora is and is not possible.

The basic idea for our account of English exempt anaphors is inspired by a proposal of Kang (1988b). Kang presents an account of reflexivization in English and Korean within a GPSG-like framework. This account is essentially a development/refinement of a storage based account but with 'storage' being handled using a feature that is inherited up phrase structure trees under a variant of GPSG's Foot Feature Principle. According to Kang, some Korean speakers allow discourse binding of the reflexive *caki*, which arises in cases where an occurrence of *caki* has no intrasentential antecedent, instead taking an extrasential antecedent corresponding to some individual established in the discourse. Kang suggests that the difference between speakers who do and do not allow discourse binding can be handled in terms of the presence or absence of the following rule (simplified somewhat):[11]

(5.89) $\text{S[REFL NP]}: f \Rightarrow fc$
where f is the meaning of S[REFL NP] and c is a designated variable

The rule applies only to *root* S nodes. The rule affects only the semantics of the constituent, causing the anaphoric argument to be instantiated as a (designated) free variable (whose value is determined by discourse factors).

The basic idea that we adopt from Kang is that unbound reflexives can be assigned a referent by a process that applies only at certain syntactic 'structural nodes'. Unlike Kang, however, we take the process to be not only semantic in its effect, but also syntactic in that it modifies the syntactic type combination that results under a proof. We can implement the idea that an unbound anaphor can be assigned a discourse referent *at some particular syntactic node* of type X using the following rule:

(5.90) (a) $\begin{array}{l}[\ominus \text{Y}]^{i} \\ \vdots \\ \dfrac{\text{X}}{\text{X}}\text{DIR}^{i}\end{array}$ (b) $\begin{array}{l}[\ominus \text{Y}:v]^{i} \\ \vdots \\ \dfrac{\text{X}:f}{\text{X}:f}\text{DIR}^{i}\end{array}$

The rule (labelled DIR, for Discourse Interpretation Rule) simply allows that a proof of a type X which rests on an assumption of type $\ominus$Y can be modified to give a proof of X with that assumption discharged. The presence of an undischarged $\ominus$ assumption in a proof indicates the presence of an unbound reflexive (or pronoun). The rule, whose semantics is stated in (5.90b), makes no change to the semantics of the proof. However, applying the

[11] The 'designated variable' c referred to in (5.89) is assumed to have special a status in the discourse model associated with the text. For example, it might always have the individual whose point of view is presented assigned for its referent. We won't try to mimic this aspect of Kang's proposal in the account to be presented, but it would not be difficult to modify the account in this way.

rule has the consequence that the variable semantics of the discharged hypothesis appears free in the overall semantics of the proof. In this way, discourse binding is linked into the syntactic system of the grammar, requiring both that the process that gives discourse binding takes place at certain nodes and also that it has the syntactic side effect of discharging an assumption. This syntactic 'tie in' provides the basis for addressing restrictions on when discourse binding may take place, restrictions which we take to be of syntactic origin.

This account is also intended to deal with the referential behaviour of personal pronouns, as well as exempt anaphors. Since there appear to be no syntactically based restrictions on when personal pronouns may show referential behaviour, we begin with the provisional version of the DIR in (5.91), which allows a personal pronoun to undergo the DIR at its 'own' np node, prior to combination with any other lexical material:

(5.91)
$$\dfrac{\begin{array}{c}[\ominus \mathrm{Y}]^{i}\\ \vdots \\ \mathrm{np}_{\alpha}\end{array}}{\mathrm{np}_{\alpha}}\,\mathrm{DIR}^{i}$$

The α subscript on the result type of the proof corresponds to some feature instantiations that are yet to be determined, which we can for now assume to be nil. The following proof shows that this rule allows a lexical personal pronoun to be directly treated as referential (by being assigned a free variable for its interpretation):

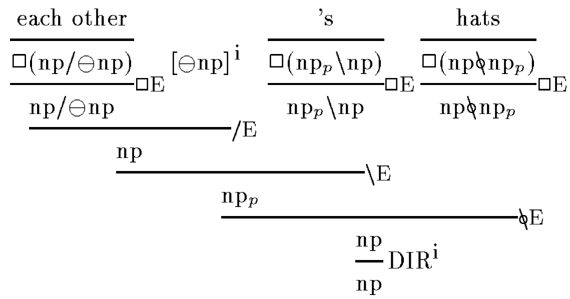
(5.92)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{he}}{\Box(\Box \mathrm{np}/\ominus \mathrm{np})}}{\Box \mathrm{np}/\ominus \mathrm{np}}\,\Box E \qquad [\ominus \mathrm{np}]^{i}}{\Box \mathrm{np}}\,/E}{\mathrm{np}}\,\Box E}{\mathrm{np}}\,\mathrm{DIR}^{i}$$

The rule as it stands is obviously inadequate, since it will freely allow discourse binding of reflexives (just as it does for personal pronouns). To prevent this, we assume that np types are marked with a feature [±ana] ('anaphoric'), an anaphor being a function into type np[+ana], and a personal pronoun a function into np[−ana]. Then, restricting the rule in (5.91) so that $\alpha = [-\mathrm{ana}]$ serves to block application of the rule directly to a lexical anaphor.

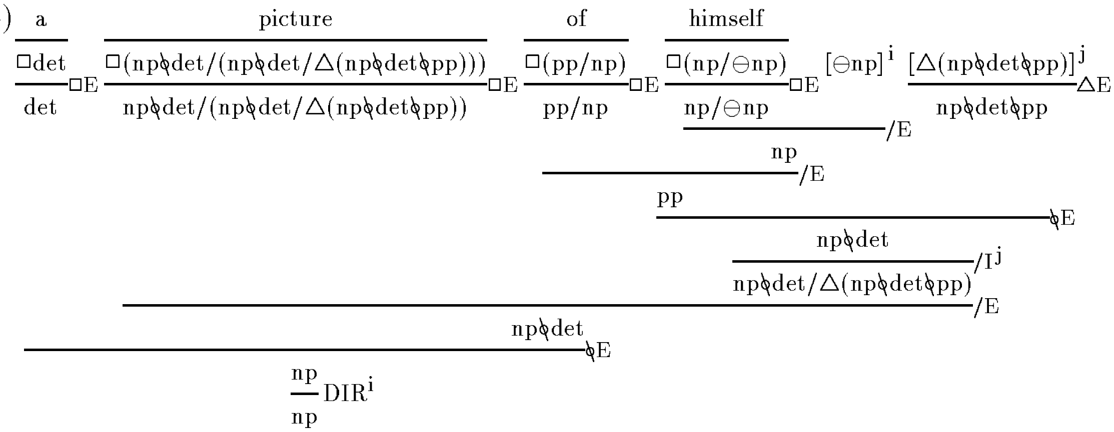
This restriction does not prevent discourse binding of an anaphor when it appears embedded within a proof of some type np[−ana]. This situation arises in the case of picture noun anaphors and possessive reciprocals, as the following proofs illustrate. (Note that we assume that the genitive *'s* combines with the possessor to give a np type with the same featural markings as the possessor's type, except changing the feature [−poss] to [+poss].)

Again, in the meanings assigned by these proofs, the anaphor corresponds to a free variable (for which the assignment of a referent is subject to discourse constraints which we do not attempt to characterize in the grammar).

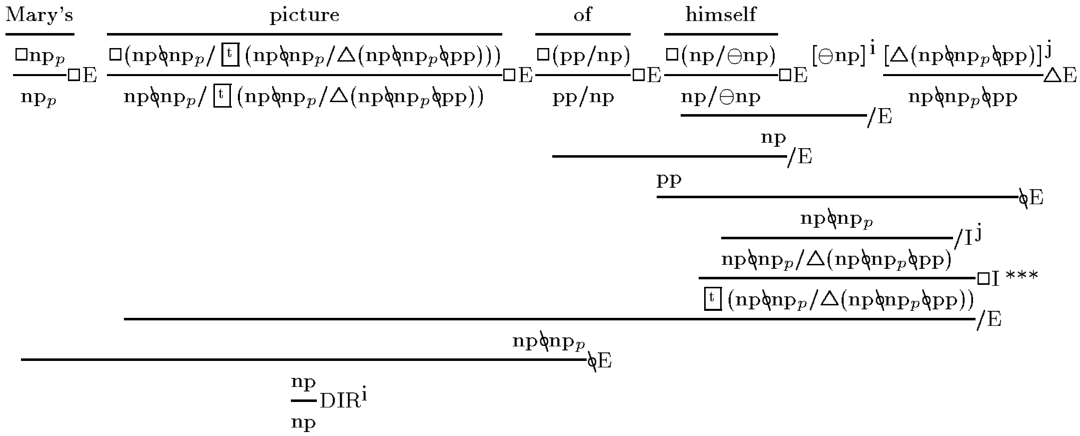
(5.93)



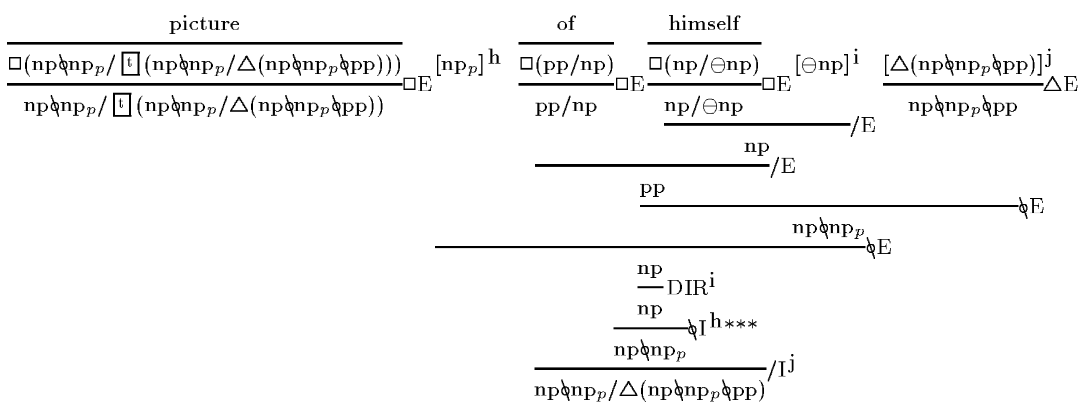
(5.94)



Discourse binding of an anaphor is restricted by the presence of linguistic boundaries in a similar fashion to syntactic binding. Because the ⊖np assumption associated with the anaphor is not box modal, it must be discharged within any modal domains containing the anaphor. For example, since nouns which take a sentential complement specify it to be a modal domain (giving rise to a subset of CNPC restrictions), a reflexive which appears in the sentential complement may not undergo discourse binding at the level of the noun's NP projection. We have seen that exempt behaviour is not possible for picture noun anaphors in possessed NPs. For our account, this fact follows from the modal boundary specified by nouns which take a possessive specifier, as illustrated by the following attempt to prove discourse binding for *Mary's picture of himself* which is not correct because the (starred) □I inference is not properly licensed:

(5.95) 

We noted above that the boundary modality of possessed NPs does not block syntactic binding of a picture noun anaphor by the possessor NP, because the RIR may be invoked within the modal domain, being applied to a projection of the hypothesized prelocation type of the noun. It is probably worth noting that the DIR *cannot* similarly apply within the modal boundary. This is because the DIR applies at nodes with type np, so the hypothesized prelocation noun type would need to be projected all the way to its ultimate value subtype np for the DIR to apply, as in (5.96). However, it is not then possible to complete the proof, given the lack of an introduction rule for the primitive subcategorization slash ⍀.

(5.96) 

We can see that this account links together the facts that possessed NPs are (i) islands to extraction, (ii) do not allow syntactic binding of a picture noun anaphor by an antecedent outside the NP, and (iii) do not allow discourse binding of an picture noun anaphor. Our account appropriately handles the examples we noted earlier which present a problem for the HPSG treatment (repeated here):

(5.97) a. Mary$_i$ was furious. A speech to the committee about herself$_i$ had been highly critical, but she had been given no chance to reply.

b. ?*Mary$_i$ was furious. John's speech to the committee about herself$_i$ had been highly critical, and she had been given no chance to reply.

Discourse binding *is* allowed in (5.97a) because no boundaries are present to block it, and is prevented in (5.97b) as in other possessive constructions because of the boundary specified on possessed nouns.

However, our account as it stands does not allow for examples such as in (5.98) involving expletive NPs *it* and *there.*

(5.98) a. ?John$_i$ knew that there was himself$_i$ left. (Pollard & Sag, 1990)

b. They$_i$ made sure that it was clear to each other$_i$ that this needed to be done. (Kuno, 1987)

In these examples, the anaphor is a dependent of a verb which seeks only one less oblique dependent, the expletive NP. Since, expletives are *non-referential,* the HPSG Principle A does not apply, and the anaphors are predicted to be exempt. We might attempt to accommodate such cases in our account by extending the set of nodes at which the DIR rule is allowed to apply. For examples such as (5.98), we might allow the DIR to apply at nodes whose type is that of a VP which seeks a non-referential subject. (To this end, we would need to adopt a feature distinguishing referential and non-referential NPs.) However, we shall not attempt to extend the account in this way here.

The sequent rules for binding

We have so far only given natural deduction versions of the RIR and DIR. This is because the statement of the corresponding sequent rules presents some problems.

(5.99)
$$\frac{\Delta,\ \ominus \mathrm{y}{:}v,\ \Omega \Rightarrow \mathrm{W}{:}f \qquad \Gamma,\ \mathrm{W}{:}(\lambda v.fv),\ \Lambda \Rightarrow \mathrm{z}{:}g}{\Gamma,\ \Delta,\ \Omega,\ \Lambda \Rightarrow \mathrm{z}{:}g}\,\mathrm{RIR} \qquad \text{where } \mathrm{W} \in \{\mathrm{x}\not\backslash\mathrm{y},\ \mathrm{x}\not/\mathrm{y}\}$$

As a first attempt, we might state the RIR as in (5.99). The intuitive correspondence of this rule to its natural deduction equivalent should be fairly clear. The antecedent $\ominus$y added in to the left premise corresponds to the assumption that is discharged in the natural deduction rule. The type derived as the succedent of the left hand premise reappears in the right hand premise having undergone the appropriate semantic transformation. However, this rule is of no use for theorem proving, since its inclusion would mean that proof search was no longer guaranteed to terminate. The rule does not exhibit the subformula property and its premises are not 'simpler' than its conclusion. Note in particular that the rule does not specify the structure of any types in the conclusion, which means that, during theorem proving, the rule could *always* be applied to a subgoal sequent, giving rise to premises to which the RIR could again be applied, and so on *ad infinitum.*

Instead of (5.99), we might state the rule as:

$$(5.100)\quad \frac{\Omega,\ \mathrm{x}{:}(hv),\ \Delta \Rightarrow \mathrm{W} \qquad \Gamma,\ \mathrm{W}{:}(\lambda v.fv),\ \Lambda \Rightarrow \mathrm{z}{:}k}{\Gamma,\ \Omega,\ \mathrm{R}{:}h,\ \Delta,\ \Lambda \Rightarrow \mathrm{z}{:}k}\mathrm{RIR} \qquad \begin{array}{l}\text{where } \mathrm{W} \in \{\mathrm{p}\obslash\mathrm{y},\ \mathrm{p}\oslash\mathrm{y}\} \\ \text{and } \mathrm{R} \in \{\mathrm{x}\backslash\ominus\mathrm{y},\ \mathrm{x}/\ominus\mathrm{y}\}\end{array}$$

This version of the rule in effect 'compiles in' the step whereby the additional antecedent introduced in the left hand premise in (5.99) is combined with the functor that takes it as argument (i.e. the anaphor type). Theorem proving using this version of the rule *will* terminate, provided we set the requirement that the left premise is proven before the right premise. The applicability of the rule depends on the presence of a type of the form $\mathrm{x}\backslash\ominus\mathrm{y}$ or $\mathrm{x}/\ominus\mathrm{y}$ in the antecedents of the conclusion, and in the left hand premise this type is simplified to x. This ensures that the rule cannot be applied to its own premises in indefinite recursion. The conclusion of the right hand premise has a primitive subcategorization slash for its principal connective. Since these connectives do not have a right rule, the left premise can only be proven if W appears as a subformula of one of its antecedents. Hence, when the left premise *can* be proven, it is simpler than the conclusion and includes only subformulas of the conclusion types. Furthermore, whenever the left hand premise is successfully proven, the value of W carried across to the right hand premise is such that it is also simpler than, and contains only subformulas of, the conclusion sequent.

In fact, the version of the rule in (5.100) is not adequate for our purposes, since it does not allow for cases where there is a mismatch between the types of the binder and the anaphor's argument. As we saw earlier (see subsection on specifying possible binders), an anaphor type $\mathrm{x}/\ominus\mathrm{y}$ can be bound by an argument of any type z such that $\mathrm{z} \Rightarrow \mathrm{y}$. This problem is remedied in the following version of the RIR. Again, we require that the premises are addressed in left to right order during theorem proving.

$$(5.101)\quad \frac{\Omega,\ \mathrm{x}{:}(hg),\ \Delta \Rightarrow \mathrm{W} \qquad \mathrm{q}{:}v \Rightarrow \mathrm{y}{:}g \qquad \Gamma,\ \mathrm{W}{:}(\lambda v.fv),\ \Lambda \Rightarrow \mathrm{z}{:}k}{\Gamma,\ \Omega,\ \mathrm{R}{:}h,\ \Delta,\ \Lambda \Rightarrow \mathrm{z}{:}k}\mathrm{RIR} \qquad \begin{array}{l}\text{where } \mathrm{W} \in \{\mathrm{p}\obslash\mathrm{q},\ \mathrm{p}\oslash\mathrm{q}\} \\ \text{and } \mathrm{R} \in \{\mathrm{x}\backslash\ominus\mathrm{y},\ \mathrm{x}/\ominus\mathrm{y}\}\end{array}$$

Unfortunately, a sequent system that includes (5.101) is not cut-eliminable. For example, the sequent (5.102) can be proven with cut, but not without. However, given the way that $\ominus$ is actually used in the account (i.e. the way that it is marked on lexical types), cut is in practice never needed to prove a linguistic example. Hence, we can exclude cut, and grammars which make use of our account of binding can be parsed.

$$(5.102)\qquad \mathrm{x}\obslash\mathrm{y}/\ominus\mathrm{z},\ \ominus(\mathrm{z}/\mathrm{y}) \Rightarrow \mathrm{x}\obslash\mathrm{y}$$

Similar considerations govern how we may state a sequent version of the DIR. The version in (5.103) corresponds most directly to the natural deduction rule, but for practical purposes, we are forced to adopt the version in (5.104).

$$(5.103)\quad \frac{\Delta,\ \ominus\mathrm{y}{:}v,\ \Omega \Rightarrow \mathrm{np}_{[-\mathrm{ana}]}{:}f \qquad \Gamma,\ \mathrm{np}_{[-\mathrm{ana}]}{:}f,\ \Lambda \Rightarrow \mathrm{z}{:}g}{\Gamma,\ \Delta,\ \Omega,\ \Lambda \Rightarrow \mathrm{z}{:}g}\mathrm{DIR}$$

$$(5.104)\quad \frac{\Delta,\ \mathrm{x}{:}hv,\ \Omega \Rightarrow \mathrm{np}_{[-\mathrm{ana}]}{:}f \qquad \Gamma,\ \mathrm{np}_{[-\mathrm{ana}]}{:}f,\ \Lambda \Rightarrow \mathrm{z}{:}g}{\Gamma,\ \Delta,\ \mathrm{R}{:}h,\ \Omega,\ \Lambda \Rightarrow \mathrm{z}{:}g}\mathrm{DIR} \quad \text{where R} \in \{\mathrm{x}\backslash\ominus\mathrm{y},\ \mathrm{x}/\ominus\mathrm{y}\}$$

5.2.3 Moortgat's account of binding

We next consider an account of reflexivization proposed by Moortgat (1990b), and suggest how it may be developed to avoid some problems and deal with additional phenomena. It turns out that there is little to choose between this account (in its developed form) and our account. Later on, however, we note some differences between the two accounts in terms of elegance and simplicity.

Quantor types

Moortgat (1990b) suggests a novel method for the treatment of quantifier scoping.[12] This method potentially provides a basis for the treatment of a range of problematic natural language phenomena where some item exhibits semantic scope that is not straightforwardly commensurate with the position at which it appears. The most obvious case for this conflict arises with quantifier scoping. The method uses a type-forming operator ^, which gives rise to so-called *quantor* types. A type X^Y is a quantifier which exhibits the distributional behaviour of elements of type X, and which is 'interpreted' (in a sense to be explained shortly) at the level of constituents of type Y. For example, np^s is a NP quantifier that is interpreted at the level of clausal constituents. This connective has the following sequent rules (which are restated with lambda semantics in (5.106)).

$$(5.105)\quad \frac{\Omega,\ \mathrm{x},\ \Lambda \Rightarrow \mathrm{y} \qquad \Gamma,\ \mathrm{y},\ \Delta \Rightarrow \mathrm{z}}{\Gamma,\ \Omega,\ \mathrm{x}\hat{}\mathrm{y},\ \Lambda,\ \Delta \Rightarrow \mathrm{Z}}[\hat{}\mathrm{L}] \qquad \frac{\Gamma \Rightarrow \mathrm{x}}{\Gamma \Rightarrow \mathrm{x}\hat{}\mathrm{y}}[\hat{}\mathrm{R}]$$

$$(5.106)\quad \frac{\Omega,\ \mathrm{x}{:}v,\ \Lambda \Rightarrow \mathrm{y}{:}f \qquad \Gamma,\ \mathrm{y}{:}q(\lambda v.f),\ \Delta \Rightarrow \mathrm{z}{:}g}{\Gamma,\ \Omega,\ \mathrm{x}\hat{}\mathrm{y}{:}q,\ \Lambda,\ \Delta \Rightarrow \mathrm{z}{:}g}[\hat{}\mathrm{L}] \qquad \frac{\Gamma \Rightarrow \mathrm{x}{:}g}{\Gamma \Rightarrow \mathrm{x}\hat{}\mathrm{y}{:}\lambda f.fg}[\hat{}\mathrm{R}]$$

The left rule is the most interesting. Given a quantor type X^Y, the rule seeks to derive a type Y from some subsequence of the types which surround the quantor type, but with a type X put in the place of the the quantor type. It is the substitution of the type X in place

[12] Moortgat relates his proposals to those of Hendriks (1988, 1990), though discussion of this relation is beyond the scope of the present work.

of the quantor type X^Y that gives the latter the distributional character of the former, and the derivation of the type Y provides a basis for handling the interpretation. Observe that in the left premise of (5.106), the type X is assigned a variable for its semantics. In the second premise, this variable is abstracted over and the quantifier's semantics applied to the result, i.e. so that the quantifier has scope at the 'level' of the constituent Y. The right rule simply allows that a quantor type X^Y can be derived from any type X, the rule's semantics being that of type raising. This apparatus provides, in terms of the inference rules for a single connective, a treatment of quantification that in many way corresponds to the Montagovian 'quantifying in' treatment, and also to the transformational 'quantifier raising' analysis.

We can state natural deduction rules for the quantor connective as follows:[13]

(5.107)
$$\frac{\begin{matrix}\vdots\\ \mathrm{X}\end{matrix}}{\mathrm{X\hat{}Y}}\,\hat{}\mathrm{I} \qquad\qquad \begin{matrix}\mathrm{X}\\ \vdots\\ \mathrm{Y}\end{matrix} \;\Rightarrow\; \frac{\begin{matrix}\dfrac{\mathrm{X\hat{}Y}}{\mathrm{X}}\,\hat{}\mathrm{E_a}^{i}\\ \vdots\\ \mathrm{Y}\end{matrix}}{\mathrm{Y}}\,\hat{}\mathrm{E_b}^{i}$$

(5.108)
$$\frac{\begin{matrix}\vdots\\ \mathrm{X}:f\end{matrix}}{\mathrm{X\hat{}Y}:\lambda g.gf}\,\hat{}\mathrm{I} \qquad\qquad \begin{matrix}\mathrm{X}:v\\ \vdots\\ \mathrm{Y}:f\end{matrix} \;\Rightarrow\; \frac{\begin{matrix}\dfrac{\mathrm{X\hat{}Y}:q}{\mathrm{X}:v}\,\hat{}\mathrm{E_a}^{i}\\ \vdots\\ \mathrm{Y}:f\end{matrix}}{\mathrm{Y}:q(\lambda v.f)}\,\hat{}\mathrm{E_b}^{i}$$

Note that the [^E] rule has been presented in before/after form since it extends a proof at both top and bottom. The following proof illustrates the use of a quantor type:

(5.109)
$$\frac{\dfrac{\dfrac{\text{John}}{\mathrm{np}} \quad \dfrac{\dfrac{\text{kissed}}{\mathrm{s\backslash np/np}} \quad \dfrac{\dfrac{\dfrac{\text{every}}{\mathrm{(np\hat{}s)/n}} \quad \dfrac{\text{girl}}{\mathrm{n}}}{\mathrm{np\hat{}s}}\,/\mathrm{E}}{\mathrm{np}}\,\hat{}\mathrm{E_a}^{i}}{\mathrm{s\backslash np}}\,/\mathrm{E}}{\mathrm{s}}\,\backslash\mathrm{E}}{\mathrm{s}}\,\hat{}\mathrm{E_b}^{i}$$

This proof assigns the meaning (5.110a), which simplifies to (5.110b), assuming *every* has the meaning (5.110c).

(5.110) a. every′ girl′ (λv.kissed′ v john′)

b. $\forall x.((\text{girl}'\ x) \rightarrow (\text{kissed}'\ x\ \text{john}'))$

c. every′ $= \lambda a.\lambda b.\forall x.((ax) \rightarrow (bx))$

[13] Pereira (1989) provides a natural deduction formulation for a quantification oriented system that is related to Moortgat's in a number of ways.

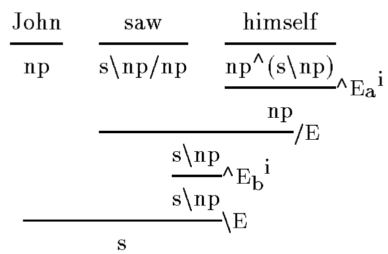
Quantors and syntactic binding

Moortgat suggests that this method can also be used for reflexivization. For example, we can handle cases where a reflexive such as *himself* has a subject antecedent by assigning a type and meaning as follows:

(5.111) np^(s\np) : himself′ (where himself′ $= \lambda v.fvv$)

The reflexive is treated as a 'quantifier' having 'duplicator' semantics (i.e. $\lambda v.fvv$), which has NP distribution and is 'interpreted' at the level of VP. The duplicator semantics causes the reflexive to be co-bound with the next argument of the function which is specified for the level of interpretation. Since the reflexive (5.111) is interpreted at VP, the antecedent is that VP's subject. The following proof illustrates a simple case, and assigns the reading shown in (5.113).

(5.112)



(5.113) a. himself′ (λx.(saw′ x)) john′

b. saw′ john′ john′

Command constraints

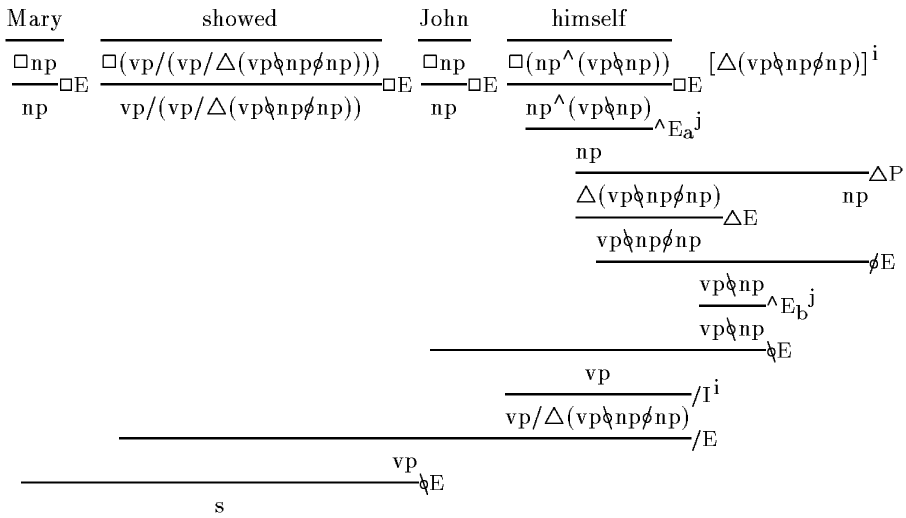
This account runs into the same problem as our account with cases for which Montague Grammar uses wrapping operations. Thus, for Moortgat's account (as for ours), the constituent specified for the level of interpretation must include as a subpart the argument position at which the reflexive quantifier appears. The (standard) type assigned for double object verbs seeks the direct object before the second object, and so it is possible to specify a reflexive quantifier type (namely np^(s\np/np)) that incorrectly allows binding of a direct object by an second object, as in **Mary showed himself*$_i$ *John*$_i$, whilst a type that yields the converse correct binding, as in *Mary showed John*$_i$ *himself*$_i$, cannot be specified.

This problem can be dealt with by adopting the treatment of word order described in Chapter 4. Then, the reflexive quantifier type:

□(np^(s\np\np))

allows the example *Mary showed John*$_i$ *himself*$_i$ to be derived as in (5.114), without admitting the ungrammatical **Mary showed himself*$_i$ *John*$_i$.

(5.114)



Adopting this treatment of word order yields an account where binding obeys o-command, where (again, as with our account) o-command is not required to be local.

Locality constraints

Moortgat's account can also be combined with the polymodal treatment of linguistic boundaries. In particular, the locality of behaviour of a reflexive quantifier X^Y is determined by the modal status of the type X.[14] Thus, the reflexive quantifier type (5.115a) would need to be bound within the minimal modal domain containing the reflexive, because the 'X' is non-modal type (np), whilst the type (5.115b) would not need to be bound within any modal domains having a modality $\boxed{\beta}$ such that $\mathcal{L}_\alpha \subseteq \mathcal{L}_\beta$. With $\mathcal{L}_\alpha = \mathcal{L}_\emptyset$, the anaphor could be bound from outside of any domain.

(5.115) a. $\Box(\mathrm{np}^\wedge(\mathrm{s}\backslash\mathrm{np}))$

b. $\Box(\boxed{\alpha}\,\mathrm{np}^\wedge(\mathrm{s}\backslash\mathrm{np}))$

Discourse binding

We saw above how our account of binding allows treatment of exempt anaphors by the addition of a single rule, without needing additional lexical types for anaphors. A similar move cannot be made for Moortgat's account given the burden that it places on the lexical type/semantics assigned to anaphors. However, it is possible to reproduce the predictions of our account of exempt anaphors in Moortgat's system by providing further lexical types. In particular, anaphors could be assigned an additional type:

[14] The remarks made here about reflexive quantifiers and locality apply equally to the use of Moortgat's method with the polymodal treatment of linguistic boundaries in relation to *any* scoping phenomenon, such as constraints on the scope of ordinary NP quantifiers.

$$\Box(\mathrm{np}_{[+\mathrm{ana}]}{}^{\wedge}\mathrm{np}_{[-\mathrm{ana}]}) : \lambda f.fv$$

specifically to allow for discourse binding examples. The anaphor's syntactic type indicates that it is a $\mathrm{np}_{[+\mathrm{ana}]}$ distributionally, and may be interpreted at the level of any dominating $\mathrm{np}_{[-\mathrm{ana}]}$ (provided that it is 'accessible' given the modal boundaries that are present). The meaning assigned is simply a type raised free variable (assumed to be distinct for each occurrence of the lexical item), with the consequence that in a resulting analysis, after the relevant inferences with respect to the quantor type, the reflexive will simply have the free variable for its interpretation. This type allows for the derivation of precisely the same examples as the DIR in our account.

5.2.4 Comparison of the two approaches

The major practical difference between the two accounts arises in respect of the specification of possible binders for an anaphor (or syntactically bound personal pronoun). For our account, the set of possible antecedents for an anaphor is jointly determined by the statement of the RIR and the type assigned for anaphors. The RIR imposes the constraint that the binder must be the first argument of a type $X \circ\!\!\!\backslash Y$ or $X \oslash Y$, the rule schematizing over the possible instantiations of these types. The type Y is restricted by the anaphor, where an argument of type Y may only bind an anaphor of type $W/\ominus Z$ such that $Y \Rightarrow Z$ is valid. Our treatment for English essentially allows that an anaphor may be bound by *any* o-commanding NP (or argument PP), provided that no modal boundaries intervene.

For (the modified version of) Moortgat's account, what is a possible binder for a reflexive quantifier can only be specified in terms of the subtype Y in the quantor type $X^{\wedge}Y$. This means that the anaphor must be given a different type to allow for each different binder possibility, including the cases where a binder is a subject or object of a verb, or the possessive NP specifier of a noun, or one of two postnominal PP complements, and so on. Different type assignments are also required to allow for the binder being a NP or argument PP. An alternative to giving each anaphor many different types is to give it a single type $X^{\wedge}Y$, where Y is the *union* of all the types needed to allow for the cases just mentioned. Both of these ways of allowing for the different binding possibilities fail to capture the straightforward generalization that covers all the cases. However, Moortgat's approach has the advantage that where an anaphor exhibits subject orientation, this fact can be encoded directly in the lexical type of the anaphor.

A further point to be raised about Moortgat's account is that it encodes directly into the elimination and introduction inferences of a single operator what can be seen to be quite complex behaviour, i.e. what is quite a large part of a treatment of quantifier scoping and anaphor binding. The use of such a complex connective constrasts with an alternative strategy that might be favoured, under which the behaviour specified for individual operators is required to be relatively simple. By this view, operators are seen as simple units which by their combination allow the complex behaviour corresponding to some linguistic phenomenon to be encoded or described.

Our account of binding is based around the operator $\ominus$, which has relatively simple behaviour, being just a permutation operator with K-like necessity behaviour. The RIR is the heart of the account of binding, and specifies a relatively straightforward intuition about the nature of syntactic binding. The RIR is evidently *not* a basic inference rule of $\ominus$.

Natural language exhibits a range of phenomena in which the semantic scoping of some item is not easily commensurate with the position at which the item appears. It remains to be seen whether Moortgat's proposal provides a basis for the treatment of such phenomena in general, or whether treatments based around the use of operators that constitute more primitive descriptive units will lead to more insightful accounts.

5.2.5 Extended binding domains: Icelandic reflexivization

In this section, our account of binding is applied to the phenomenon of long distance reflexivization in Icelandic, first described by Thrainsson (1976). This will show, we hope, that the view of binding that our account embodies, combining an obliqueness based treatment of command with the polymodal treatment of locality constraints, provides an effective framework for addressing phenomena involving extended domains for anaphor binding. Although our account is used in this section, the developed version of Moortgat's account could equally well be used as the two approaches differ little in terms of their empirical ramifications.

Icelandic long distance reflexivization

Reflexivization in Icelandic is in general clause bounded, but under certain conditions unbounded or 'long distance' reflexivization (LDR) is possible. Icelandic LDR is subject oriented, where the relevant notion of 'subject' includes possessive NP determiners.[15] Thrainsson (1976) points to a relationship between the occurrence of LDR and the presence of subjunctive mood. LDR arises only when the reflexive appears in a complement possessing

[15] Non-subject antecedents are allowed in some dialects of Icelandic, but only with *clause bounded* binding. We do not attempt to treat these cases here.

subjunctive mood, as shown by the following minimal pair:

(5.116) a. *Jón$_i$ veit aδ María elskar sig$_i$
Jon$_i$ knows that Maria loves(I) REFL$_i$

b. Jón$_i$ segir aδ María elski sig$_i$
Jon$_i$ knows that Maria loves(S) REFL$_i$

(we use *I* and *S* in the glosses of these examples to indicate a verb in indicative or subjunctive mood, respectively.) In (5.116a), where the embedded clause has indicative mood, the reflexive cannot be bound by the matrix subject. (Note that the example is ungrammatical only under the intended reading — the embedded subject *is* a possible antecedent for *sig*). In (5.116b), the embedded clause is in subjunctive mood, and long binding by the matrix subject is allowed. The next example, where each subject is a possible antecedent, shows that binding may be made across any number of clause boundaries provided that the intervening clauses are all in subjunctive mood:

(5.117) Jón$_i$ segir aδ María$_j$ telji aδ Haraldur$_k$ vilji aδ Billi$_l$ heimsæki sig$_{i/j/k/l}$
Jon knows that Maria believes(S) that Harold wants(S) that Billy visit(S) REFL

Subjunctive mood in Icelandic is introduced by particular items, such as subordinating conjunctions (e.g. *nema* 'unless' and *ef* 'if') and nonfactive verbs. For example, a nonfactive verb such as *segir* ('say') may introduce subjunctive mood, whereas a factive verb like *víta* ('know') does not, the latter generally taking an indicative complement. However, what has been called a *domino effect* is observed (first noted by Thrainsson, 1976) such that a subjunctive mood introduced onto some complement may be propagated down through embedded complements, where subjunctive mood might not otherwise appear. In (5.117), the verb *segir* introduces the subjunctive mood onto its complement, from whence subjunctive 'trickles down' onto the complements embedded under it.

An account of Icelandic long distance reflexivization

We assume the following version of the RIR for Icelandic LDR which allows only subject antecedents, and the type shown in (5.119) for long distance reflexives.

(5.118)
$$\dfrac{\begin{array}{c}[\ominus X]^{i}\\ \vdots\\ Y\end{array}}{Y}\,\mathrm{RIR}^{i}$$
Where Y is of the form $W \obslash X$ or $W \oslash X$ and $Y \in \{vp, \overline{N}\}$
(i.e. where $\overline{N}$ ranges over the set $\{np \obslash det,\ np \oslash det,\ np \obslash np_p,\ np \oslash np_p\}$)

(5.119) $\Box(\boxed{u}\, np/\ominus np)$

Note that the value subtype of (5.119) is marked with a modality $\boxed{\mathrm{u}}$, whose presence, as we saw earlier, will make syntactic binding insensitive to the presence of any modal boundaries that have a modality $\boxed{\alpha}$ such that $\mathcal{L}_{\mathrm{u}} \subseteq \mathcal{L}_{\alpha}$.

Our account of Icelandic LDR is based on the following condition:

(5.120) Subjunctive Boundaries Condition:
where a lexical functor seeks a complement of type $\boxed{\alpha}X$, then:
$\mathcal{L}_{\mathrm{u}} \subseteq \mathcal{L}_{\alpha}$ holds *if and only if* the type X is marked [+subjunctive]

We assume that this requirement applies in the lexicon, constraining the possibilities for lexical type assignment, with the consequence that modal boundaries marked on [+subjunctive] complements do not present barriers to binding of LD reflexives. The requirement is biconditional, so that binding of LD reflexives is insensitive *only* to boundaries specified on [+subjunctive] complements.[16] This gives the observed dependence of LDR on the presence of subjunctive mood.

Concerning the 'domino effect' observed of the distribution of subjunctive mood, we assume a simple lexical account based on a generalization about the distribution of the [+subjunctive] feature on lexical types, which can be stated as a 'licensing condition' as follows:

(5.121) 'Domino effect' Condition:
Any functor that is itself marked as [+subjunctive] may specify the feature [+subjunctive] on any of its complements.

This licensing condition has the effect that there is no limitation on the 'downward transmission' of the [+subjunctive] feature, with the consequence that the [subjunctive] feature may be specified on many constituents where it has no morphological realization. Note that the Subjunctive Boundaries Condition (SBC) in (5.120) refers only to whether an argument of a functor is marked [+subjunctive], no reference is made to the feature markings on the functor itself. Hence, the SBC applies equally to cases where an argument is marked [+subjunctive] under the domino effect as where a functor actually *introduces* subjunctive mood. This characteristic of the SBC (i.e. that it refers only to the featural status of the argument of a functor) means that the approach accounts quite straightforwardly for what might otherwise seem a puzzling characteristic of Icelandic LDR, i.e. why the mood of the clause in which the *antecedent* appears is irrelevant to the possibility of LDR.

[16] Maling (1984, p236, fn.5) notes dialect variation whereby some speakers allow cases of LDR that violate the standard subjunctive mood requirement. Accommodating such dialects would seem to require that one side of the biconditional in (5.120) be weakened, i.e. so that boundaries which do not block binding of a LD reflexive can be specified on a [−subjunctive] complement. Later, we will discuss reasons for which we may wish to weaken the other side of the biconditional.

Let us briefly note an alternative possibility for how we might handle Icelandic LDR using the apparatus of our approach. The possibility exists that we could use the box modality marked on each lexical type, which licenses its appearance within modal domains, to limit its distribution. In general, a lexical type $\boxed{\alpha}X$ may appear within a domain with boundary modality $\boxed{\beta}$ only if $\mathcal{L}_\alpha \subseteq \mathcal{L}_\beta$. (So far, lexical types have been assumed to be marked with the 'strong' modality $\Box$, having defining language $\mathcal{L}_\emptyset$, so that no such distributional restrictions arise, i.e. since $\mathcal{L}_\emptyset$ is a subset of all other defining languages used.) We could use modalities marked on lexical types to implement constraints on their distribution which would otherwise be handled using features, for example the distribution of subjunctive mood. In that case, we might be able to give an account of Icelandic LDR where the Subjunctive Boundaries Condition was replaced with a much simpler condition relating the modality marked on LD reflexives (which allows evasion of boundary constraints) to the modality (or modalities) that determine the *distribution* of subjunctive mood. Various fascinating possibilities arise for how we might develop the account in this direction, but investigation of these possibilities is beyond the scope of the present work.

Possible counterarguments to the account

This account of Icelandic LDR does not differ at all in its essential view of what is important in determining binding possibilities from the account as it was developed in relation to English, i.e. binding requires that the antecedent o-commands the anaphor, and that no 'blocking' boundaries intervene between the two. Maling (1984) argues that c-command is irrelevant for Icelandic LDR. These arguments apply also against o-command, and so warrant our close attention.

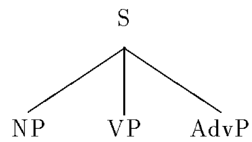
Maling's arguments for the irrelevance of c-command to Icelandic LDR are based around two classes of 'counterexample' which arise with adverbial clauses and certain possessive constructions. We consider first the case of adverbials. As noted by Thrainsson (1976), a matrix subject may not bind a reflexive occurring in an adverbial clause, e.g:

(5.122) a. *Jón$_i$ yrδi glaδur ef Sigga byδi sér$_i$
John$_i$ would-be(S) glad if Sigga invited(S) REFL$_i$

b. *Jón$_i$ kemu ekki nema Sigga bjóδi sér$_i$
John$_i$ comes(I) not unless Sigga invites(S) REFL$_i$

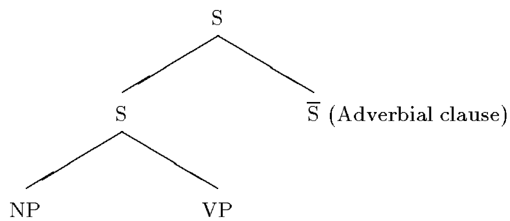
Given that the adverbial clause is in subjunctive mood, the impossibility of LDR seems puzzling under the structure that Maling assumes, shown in (5.123), in which the matrix subject c-commands the adverbial and hence the anaphor within it.

(5.123)



The obvious manoeuvre to make in response to this argument is to claim that these examples have a different structure to (5.123), one in which the matrix subject does not command the adverbial clause, such as the following possible structure that Maling discusses:

(5.124)



Maling argues that to assume this alternative structure simply to ensure that c-command does not hold, without independent motivation for it, would be to rob c-command of its predictive power. From a categorial perspective, however, where a central underlying assumption is that syntactic structure is essentially just a refinement of semantic 'structure', it is Maling's structure (5.123) that seems in need of motivation. Since an adverbial clause such as *nema Sigga bjóði sér*$_i$ (*unless Sigga invites(S) REFL*) semantically relates *propositions* rather than *predicates*, it is appropriate that it receives a clause modifier type s\s. This type for the adverbial has the consequence that the matrix subject does not o-command the adverbial clause, explaining the impossibility of LDR in this case.

Maling points out that a reflexive *can* take the matrix subject as antecedent if it appears in an adverbial that is "phrasal (i.e. not sentential)", as in (5.125):

(5.125) Jón$_i$ kemur ekki án konu sinnar$_i$/*hans$_i$
Joh comes not without wife(G) REFL$_i$/*his[−refl]$_i$
'John won't come without his wife'

Maling argues that if we seek to block LDR in the sentential adverb cases by assuming the structure (5.124), then cases like (5.125) present a problem, obliging us to assign a different structural position for nonsentential adverbials. However, under the categorial assumption that syntactic structure reflects semantic structure, different attachment for these adverbials is precisely what we expect, i.e. since they modify the predicate semantics of the VP, we expect a type such as vp\vp. With this type, the subject argument *does* o-command the anaphor inside the adverbial, and so the grammaticality of (5.125) is predicted.

Maling argues in favour of her assumed structure (5.123) that it has provided for an explanation of some disjoint reference data in terms of Binding Principle C (stated in (5.1)).

The relevant facts are the same for Icelandic and English, and so we consider the latter:

(5.126) a. John_i won't come unless he_i is invited.

b. *He_i won't come unless John_i is invited.

Under Maling's structure, the matrix subject c-commands the adverbial clause, and so (5.126b) is a Principle C violation, explaining its ungrammaticality. Under the structure in (5.124), the command relation does not obtain, and so this explanation is not available.

We take the data in (5.126) to result from a pragmatic/discourse linear ordering requirement on referential (as opposed to bound) use of pronouns not commanded by their antecedent (as obtains under the structure we assume), even though this contradicts what is almost 'received wisdom' in GB. Thus, we explain the data in (5.126) in the same way as the infelicity of the (b) cases in (5.127) and (5.128):

(5.127) a. John_i arrived at the house and he_i ate a bagel.

b. *He_i arrived at the house and John_i ate a bagel.

(5.128) a. John_i arrived at the house. He_i ate a bagel.

b. *He_i arrived at the house. John_i ate a bagel.

This linear order requirement appears to be limited in its application, perhaps to the use of full NPs and pronouns in non-overlapping clauses (which seem sensible sized units for a discourse rule to apply to), since it does not rule out coreference of a pronoun with a subsequent full NP in cases such as *Every man who knows* $\textit{him}_i$ *thinks that* $\textit{John}_i$ *is a fool*.

However, a problem arises with the following cases, where the adverb appears initially:

(5.129) a. Unless he_i is invited, John_i won't come.

b. Unless John_i is invited, he_i won't come.

Maling assumes that such adverbials are fronted and Chomsky-adjoined to S. Consequently, the subject NP does not c-command the pronoun, so Principle C does not apply, explaining why neither sentence is ungrammatical (although, of course, this view does not explain the unacceptability of (5.127)). We might attempt to explain the acceptability of both orderings in (5.129) in terms of the adverbial being fronted, e.g. the ordering constraint might not apply to moved clauses, or it might allow a sentence provided that *either* the observed word order meets the requirement *or* the word order in the corresponding canonical sentence would (a sort of pragmatic variant of a 'reconstruction' explanation). We shall not attempt to develop such an account here.

Maling presents some examples involving possessive constructions that, unlike the adverbial cases, do present a problem for our account. For example:

(5.130) Skoδun Siggu$_i$ er aδ sig$_i$ vanti hæfileika.
opinion Sigga$_i$'s is that REFL$_i$ lacks(S) talent

The antecedent *Siggu* does not o-command the reflexive, and so LDR should not be possible. However, this apparent counterexample to the command requirement is not specific to Icelandic LDR, but seems rather to be linked to possessive constructions in general. Reinhart (1983) notes a comparable c-command violation in examples such as:

(5.131) Every boy$_i$'s mother loves him$_i$.

This example admits a bound variable interpretation suggesting that the quantifier *every boy* binds the object pronoun even though the former does not c-command the latter under any obvious structure.

The problems that arise with possessive constructions do not indicate specifically that command is irrelevant for Icelandic LDR, but rather reveal a problem for the use of command restrictions in the treatment of binding in general. Given that command restrictions such as c-command and o-command have proved of great value in explaining a broad range of binding phenomena, we do not take the existence of this exception to indicate that we should discard command based explanations, but rather that we have further to go in developing the concepts needed to handle binding phenomena.

Logophoricity

Maling (1984) suggests that Icelandic LDR is subject to a further constraint, namely that it is *logophoric*. This requires that the antecedent of a LD reflexive must be the individual whose point of view is being presented, i.e. whose speech, thought, belief or feelings are being reported (a so-called *logocentric entity*). This requirement is illustrated by the following sentences, from Maling (1984):

(5.132) a. Hann$_i$ sagδi aδ sig$_i$ vantaδi hæfileika.
He$_i$ said that REFL(A)$_i$ lacked(S) ability
'He said that he lacked ability'

b. *Honum$_i$ var sagδt aδ sig$_i$ vantaδi hæfileika.
Him$_i$ was said that REFL(A)$_i$ lacked(S) ability
'He was told that he lacked ability'

(5.133) a. Hún$_i$ sagði að mér líkaði vel nýja bókin sin$_i$.
She$_i$ said that me(D) liked(S) well new book REFL$_i$
'She said that I like her new book a lot.'

b. *Henni$_i$ var sagðt að mér líkaði vel nýja bókin sin$_i$.
She$_i$ was said that me(D) liked(S) well new book REFL$_i$
'She was told that I like her new book a lot.'

Maling attributes the ungrammaticality of the (b) examples to the fact that the matrix subject of the passive verb is not the source of the reported discourse. Note that the unacceptability of these sentences cannot be attributed to the presence of passive *per se* given the acceptability of the passive sentence (5.134) in which the logophoricity requirement is satisfied, nor to the oblique case marking of the subject given the acceptability of (5.135).

(5.134) Honum$_i$ var talin trú um, að sig$_i$ vantaði hæfileika.
Him(D) was convinced belief about that REFL$_i$ lacked(S) ability
'He was made to believe that he lacked ability.'

(5.135) Honum$_i$ finnst að sig$_i$ vanti hæfileika.
Him(D)$_i$ finds that REFL(A)$_i$ lacks(S) ability
'He thinks that he lacks ability.'

The contrasts illustrated in (5.132) and (5.133) do suggest that something like a logophoricity constraint applies to Icelandic LDR. A problem arises, however, for examples such as (5.117), repeated here:

(5.136) Jón$_i$ segir að María$_j$ telji að Haraldur$_k$ vilji að Billi$_l$ heimsæki sig$_{i/j/k/l}$
Jon knows that Maria believes(S) that Harold wants(S) that Billy visit(S) REFL

It is generally assumed in discourse research that a sentence or clause may have *at most one* point of view, which does not seem to fit with the observation that any of a number of NPs may bind the reflexive in an example such as (5.136). It may be, however, that a sentence such as (5.136) is ambiguous as to whose point of view is dominant in the discourse models that may be provided for it (with the selection of an antecedent for the reflexive placing a constraint that eliminates this ambiguity). An alternative possibility is that the requirement in operation is that the semantic characteristics of the antecedent are such that it is *potentially* an individual whose point of view is presented, rather than the individual whose point of view actually *is* presented in the given use of the sentence. This would follow from the lexical semantics of the verb which takes the subject, i.e. if the verb's semantics has 'logocentric entailments' with respect to its subject argument.

If in the logophoric requirements on Icelandic LDR we see a discourse requirement that applies directly in determining acceptability, then the treatment of this requirement would fall outside the scope of our essentially syntactic framework. However, if we are correct in our conjecture that the requirement actually concerns the semantic characteristics that are entailed for the antecedent by its governing verb, the possibility arises for 'grammaticalizing' this aspect of the semantics as a basis for a syntactic treatment. For example, we might assume a feature [±log] ('logocentric') marked on NPs, such that verbs which have the appropriate entailments with respect to their subject mark it as [+log], other NPs being marked [−log]. Then, assigning LDR reflexives the following type would restrict these to taking only the appropriate antecedents.

$$\Box(\boxed{\scriptstyle u}\,np/\ominus np_{[+log]})$$

An alternative possibility for explaining the unavailability of the matrix subject as an antecedent in examples such as (5.132b) is that this results from the boundary that the matrix verb specifies on its complement, i.e. that the boundary does not license LDR. To allow this explanation, we might alter the SBC so that a verb would be able to specify a non-blocking modality on its complement *if and only if* that complement was both [+subjunctive] *and* the verb's semantics had the appropriate characteristics (i.e. 'logocentric entailments'). Further research is required to allow evaluation of this novel view of how a logophoricity requirement might arise.

5.3 Conclusion

We have addressed the treatment of binding phenomena within the Lambek calculus framework. An account has been developed which exploits the polymodal treatment of linguistic boundaries and the new account of word order developed in the preceding chapters as a basis for handling locality and command constraints on binding. In addition, an alternative Lambek account, due to Moortgat (1990b), has been developed to incorporate the same treatments of locality and command constraints. The two accounts differ little in their empirical ramifications, although we have discussed some practical and methodological issues which provide a basis for favouring each account above the other.

We have addressed both syntactic binding of anaphors and the discourse governed referential behaviour that anaphors exhibit in some languages. We have argued for a view of these phenomena in English in which the possibilities for syntactic and discourse binding overlap. The Lambek account presented accords with this view of the data.

A considerable amount of attention has been given to the phenomenon of long distance reflexivization in Icelandic. The account we have developed hopefully demonstrates the value of the polymodal treatment of linguistic boundaries in handling phenomena involving complicated locality behaviour such as the constrained extension of binding domains exhibited by Icelandic LDR.

Chapter 6

Efficient Lambek Theorem Proving

6.1 Introduction

This chapter addresses natural language parsing for Lambek calculus grammars, and in particular, how we may improve its efficiency. Since our linguistic framework is also a logical calculus, parsing may be treated as an instance of automated theorem proving. As we have seen, there is more than one formulation of the Lambek calculus. For the purpose of parsing, the Gentzen-style sequent formulation of the Lambek calculus is most useful since it is demonstrably decidable and allows for a very simple theorem proving method. However, a problem arises even for unextended Lambek calculus which threatens to undermine efficient parsing. This is that the calculus typically allows many distinct proofs of a given sequent that assign the same meaning. Since search for proofs of a sequent must be exhaustive to be sure of finding all different 'readings' that may be assigned, a naive theorem prover must do a lot of unnecessary work constructing proofs that assign the same meaning, wasted effort which radically reduces the efficiency of Lambek calculus theorem proving.

In this chapter, we consider the Gentzen-style sequent formulation of the Lambek calculus, and automated theorem proving based on it, as well as the problem presented by the occurrence of multiple equivalent proofs. Then we consider some attempts to deal with this problem. These solutions are all based around defining a notion of *normal form* for Lambek proofs, as a basis for limiting the search space for proofs of any sequent that must be addressed during theorem proving. Firstly. we consider the proposals of König (1989). After that we consider the proposals of Moortgat (1990b) and Hepple (1990d), which can be seen to develop the basic idea of König's method but which attempt to avoid some problems that arise for it.

6.2 The sequent formulation of the Lambek calculus

In this section, the Gentzen-style sequent formulation of the Lambek calculus is stated. Much of this has already been outlined in earlier chapters, but it seems appropriate to restate it here given that the chapter relies so heavily upon it. The Gentzen formulation of the Lambek calculus is a calculus of *sequents*. A sequent is an object of the form:

$$\Gamma \Rightarrow x$$

where $\Rightarrow$, the derivability relation, indicates that the *succedent* x can be derived from the *antecedents* Γ. Specific to the Lambek calculus, we require that each sequent has a non-empty antecedent sequence and precisely one succedent type. In this chapter, we address only the implicational fragment of the Lambek calculus (the 'product-free' calculus),[1] which can be formulated with the following rules (due to Lambek, 1958):

(6.1) Axiom: $x \Rightarrow x$

Right rules:

$$\frac{\Gamma, y \Rightarrow x}{\Gamma \Rightarrow x/y}[/R] \qquad \frac{y, \Gamma \Rightarrow x}{\Gamma \Rightarrow x\backslash y}[\backslash R]$$

Left rules:

$$\frac{\Delta \Rightarrow y \quad \Gamma, x, \Lambda \Rightarrow z}{\Gamma, x/y, \Delta, \Lambda \Rightarrow z}[/L] \qquad \frac{\Delta \Rightarrow y \quad \Gamma, x, \Lambda \Rightarrow z}{\Gamma, \Delta, x\backslash y, \Lambda \Rightarrow z}[\backslash L]$$

Cut rule:

$$\frac{\Delta \Rightarrow x \quad \Gamma, x, \Lambda \Rightarrow y}{\Gamma, \Delta, \Lambda \Rightarrow y}[cut]$$

We refer to this formulation as **L**. The axiom schema expresses the notion that any type derives itself. The left rules are so-called since they govern the behaviour of a connective of an antecedent type (i.e. a type occurring to the left of the derivability relation $\Rightarrow$), whilst right rules govern the behaviour of the principal connective of the succedent type. We refer to the type whose connective is the focus of an inference rule as the *active type* for that inference. In contrast to the left and right rules, the cut rule is non-logical in that it does not manipulate a logical connective. Rather it is *structural,* admitting flexibility in the construction of proofs. The cut rule expresses a fundamental property of the system, namely the transitivity of derivability. The use of these inference rules was illustrated in Chapter 2.

Lambek (1958) demonstrates that this formulation has the property of *cut elimination,*

[1] The *implicational* fragment of the Lambek calculus, sometimes called the product-free calculus, uses only the type-forming operators / and \, and is so-called because these operators correspond to directional versions of material implication ($\rightarrow$).

i.e. that every proof based on these rules can be transformed into one having the same end-sequent but containing no uses of the cut rule. It follows that any sequent which is a theorem under **L** may be proven without the cut rule. Top-down search for proofs of any sequent using just the remaining rules is guaranteed to terminate. This is because, firstly, the premise(s) of each rule are (in sum) simpler than the conclusion (under a metric which counts the number of connectives), and secondly, premises are constructed using only subformulas of the types occurring in the conclusion. Consequently, the search space for proofs is finite. Since proof search is guaranteed to terminate, theoremhood is *decidable.*

Moortgat (1988) shows how this system may be extended to give the semantics for valid type combinations. To this purpose, the antecedents and succedent of a sequent are taken to consist of a pair TYPE:TERM, where TERM is a lambda expression, sometimes called a *proof term*, giving the rule system in (6.2). The left rules correspond semantically to functional application, the right rules to functional abstraction. The proof term for the succedent gives its meaning as a combination of the meanings of the antecedent types. This will be loosely referred to as the 'meaning (or reading) assigned by the proof'. For the purposes of this chapter, we assume that each antecedent type of the root sequent of any proof is assigned a distinct variable for its semantics. This is because we are interested in the equivalence or otherwise of distinct proofs for some sequent aside from contingent equivalences that may arise given particular semantic assignments to antecedents.

(6.2) Axiom: $\mathrm{x}{:}f \Rightarrow \mathrm{x}{:}f$

Right rules: $\dfrac{\Gamma,\ \mathrm{y}{:}i \Rightarrow \mathrm{x}{:}f}{\Gamma \Rightarrow \mathrm{x/y}{:}\lambda i.f}[/\mathrm{R}]$ $\qquad \dfrac{\mathrm{y}{:}i,\ \Gamma \Rightarrow \mathrm{x}{:}f}{\Gamma \Rightarrow \mathrm{x\backslash y}{:}\lambda i.f}[\backslash\mathrm{R}]$

Left rules: $\dfrac{\Delta \Rightarrow \mathrm{y}{:}g \qquad \Gamma,\ \mathrm{x}{:}fg,\ \Lambda \Rightarrow \mathrm{z}{:}h}{\Gamma,\ \mathrm{x/y}{:}f,\ \Delta,\ \Lambda \Rightarrow \mathrm{z}{:}h}[/\mathrm{L}]$ $\qquad \dfrac{\Delta \Rightarrow \mathrm{y}{:}g \qquad \Gamma,\ \mathrm{x}{:}fg,\ \Lambda \Rightarrow \mathrm{z}{:}h}{\Gamma,\ \Delta,\ \mathrm{x\backslash y}{:}f,\ \Lambda \Rightarrow \mathrm{z}{:}h}[\backslash\mathrm{L}]$

Cut rule: $\dfrac{\Delta \Rightarrow \mathrm{x}{:}f \qquad \Gamma,\ \mathrm{x}{:}f,\ \Lambda \Rightarrow \mathrm{y}{:}g}{\Gamma,\ \Delta,\ \Lambda \Rightarrow \mathrm{y}{:}g}[\mathrm{cut}]$

Moortgat (1990a) demonstrates that cut elimination preserves the *strong recognizing capacity* of the calculus in the sense that the systems with and without cut yield precisely the same readings for any theorem modulo logical equivalence.

We will note a few terminological points. Observe that the left rules have been stated so that the value subtype of the active type appears as an antecedent of the right hand premise and the argument subtype of the active type appears as the succedent of the left hand premise. Given this, we call the right hand premise of a left rule its *major* premise, and the left hand premise its *minor* premise. We call the bottom-most sequent of any proof the *root*, and the inference that has the root as its conclusion the *root inference*. Where the root sequent of some proof or subproof is a left inference, we refer to the proofs of its major and minor premises as the *major subproof* and *minor subproof* of that (sub)proof, respectively. We call the *main branch* of any cut-free proof that (unique) path s_1, s_2,..,s_n through the proof such that s_1 is the root sequent, s_n corresponds to an axiom instance, and none of s_2,..,s_n is the minor premise of a left inference. We call the axiom leaf that lies on a proof's main branch, the *main branch axiom*. In addition, we use symbols **P**, **Q**, **R**, etc to designate proofs, and the notation **P**(S) to represent a proof **P** of the sequent S.

6.3 Equivalence and the lambda calculus

In this section, we briefly consider the lambda calculus, particularly some points that are of relevance to the equivalence of proof terms assigned by Lambek sequent proofs.

Firstly, the equivalence of lambda expressions is itself addressed in terms of a normal form system, based on the method of *reduction*. This method involves defining a *contraction* relation ($\rhd_1$) between terms, which is stated as a number of contraction rules, each of the form X $\rhd_1$ Y (in which the form on the left is termed a *redex* and the form on the right its *contractum*). Each contraction rule allows that any term containing an occurrence of a redex may be transformed into a term in which that occurrence is replaced by its contractum. A term is said to be in normal form (NF) if and only if it contains no occurrences of any redex. The contraction relation generates a *reduction* relation ($\rhd$) which is such that X *reduces* to Y (X $\rhd$ Y) if and only if Y is obtained from X by a finite series (possibly zero) of contractions. A term Y is a NF of X if and only if Y is a NF and X $\rhd$ Y.

For the lambda calculus, *extensional equivalence* is given by the $\beta\eta$-reduction system, which is based around the following two contraction rules:

(6.3) (β) $(\lambda x.\mathrm{M})\mathrm{N} \rhd_1 [\mathrm{N}/x]\mathrm{M}$
(η) $\lambda x.\mathrm{M}x \rhd_1 \mathrm{M}$ (where x not free in M)

(The notation [N/x]M stands for the result of substituting N in place of every free occurrence of the variable x in M.) The reduction system that results given these two contraction rules exhibits a significant property called the *Church-Rosser* property, from which a number of consequences follow. (We will state the content of the Church-Rosser property later in the chapter.) Firstly, if a term has *any* NF (and not all terms do in the general case), then it has a *unique* NF. Secondly, if any two terms P and Q are extensionally equivalent, then provided they have any NF, they must have the *same* (unique) NF.

The completely general lambda calculus (the so-called λK-calculus) allows that a lambda operator may bind any number of occurrences of a variable in a term, including zero. The proof terms assigned by proofs in **L**, however, involve only a subsystem of the full lambda calculus which we might call the *single-bind* lambda calculus, where each lambda operator binds exactly one variable occurrence. For the single-bind lambda calculus, contraction strictly decreases the size of a term, so that reduction exhibits *strong normalization*, the property that every term reduces to a NF in a finite number of steps. This has the corollary *normalization* that every term has a NF. Thus, the proof terms given by two Lambek proofs are equivalent just in case they reduce to a common NF in a finite number of steps.

An important property of the system **L** is that *cut free* proofs assign proof terms that are in β-NF. This point is well known, but given its significance for what follows, we provide the following proof. The main lemma of our proof is Lemma A. For convenience, we adopt a predicate BNF, such that BNF(T) is true iff T is in β-NF, and a predicate BNF* which applies to a sequence of type/term pairs, such that BNF*(Δ) is true iff Δ is empty or the term of each type/term pair in Δ is in β-NF. Lemma A states that if the proof terms of the antecedents of some sequent are all in β-NF, then so also is the proof term for the succedent under any cut-free proof. Since we assume that root sequent antecedents are assigned unique variables for their proof terms, it follows from Lemma A that any cut-free proof in **L** assigns a proof term in β-NF. Lemma A rests on Lemma B, given below.

(6.4) Lemma A: for any proof $\mathbf{P}(\Delta \Rightarrow \mathrm{x}{:}f)$, $\quad \textsc{bnf}^*(\Delta) \rightarrow \textsc{bnf}(f)$.

PROOF: proof is by induction on **P**:

CASE 1: $\mathbf{P} \equiv \mathrm{x}{:}f \Rightarrow \mathrm{x}{:}f$. The induction hypothesis obviously holds.

CASE 2: $\mathbf{P} \equiv \dfrac{\mathbf{R}(\Lambda \Rightarrow \mathrm{y}{:}g) \qquad \mathbf{Q}(\Delta,\, \mathrm{x}{:}fg,\, \Gamma \Rightarrow \mathrm{z}{:}h)}{\Delta,\, \mathrm{x/y}{:}f,\, \Lambda,\, \Gamma \Rightarrow \mathrm{z}{:}h}[/\mathrm{L}]$

(i)	$\textsc{bnf}^*(\Lambda) \rightarrow \textsc{bnf}(g)$	by induction hypothesis on **R**,
(ii)	$\textsc{bnf}^*(\Delta,\, \mathrm{x}{:}fg,\, \Gamma) \rightarrow \textsc{bnf}(h)$	by induction hypothesis on **Q**,
(iii)	$\textsc{bnf}^*(\Delta,\, \mathrm{x/y}{:}f,\, \Lambda,\, \Gamma)$	assumption,
(iv)	$\textsc{bnf}(g)$	by (i), (iii),
(v)	$\textsc{bnf}(fg)$	by (iii), (iv) and Lemma B,
(vi)	$\textsc{bnf}(h)$	by (ii), (iii), (v).

Hence, $\textsc{bnf}^*(\Delta,\, \mathrm{x/y}{:}f,\, \Lambda,\, \Gamma) \rightarrow \textsc{bnf}(h)$ and the induction hypothesis is true for **P**.

CASE 3: Root inference of **P** is [\L]. As for CASE 2.

CASE 4: $\mathbf{P} \equiv \dfrac{\mathbf{Q}(\Gamma,\, \mathrm{x}{:}v \Rightarrow \mathrm{y}{:}f)}{\Gamma \Rightarrow \mathrm{y/x}{:}\lambda v.f}[/\mathrm{R}]$

Induction hypothesis on **Q** gives $\textsc{bnf}^*(\Gamma,\, \mathrm{x}{:}v) \rightarrow \textsc{bnf}(f)$. $\textsc{bnf}(f)$ implies $\textsc{bnf}(\lambda v.f)$. Hence $\textsc{bnf}^*(\Gamma) \rightarrow \textsc{bnf}(\lambda v.f)$, so induction hypothesis is true for **P**.

CASE 5: Root inference of **P** is [\R]. As for CASE 4.

(6.5) Lemma B: In any cut-free **L** proof, no antecedent proof term is of the form $(\lambda x.M)$.

PROOF: Examining the left and right rules of **L**, it is obvious that, if in the conclusion of an inference, no antecedent has a proof term of the form $(\lambda x.M)$, then the same will hold for the premise antecedents. Obviously then, the same is true for *all* antecedents in a cut-free proof whose root sequent meets this requirement. Recall that root sequent antecedents are assigned simple variables for their proof terms.

Since cut free proofs assign terms that are in β-NF, it follows that the proof terms assigned by two cut free proofs are extensionally equivalent only if they reduce to some common NF term under finite sequences of η-contractions.

6.4 Lambek calculus theorem proving

6.4.1 Theorem proving

In this subsection, we briefly discuss Lambek calculus theorem proving. We will not go into great detail, since we require only the general idea for the purposes of this chapter. The interested reader should consult Moortgat (1988), where this topic is discussed thoroughly.

The task of theorem proving is to establish the theoremhood of any given sequent (the *goal sequent*). To do this, we need only construct a proof of the sequent using the rules of the calculus. Proof construction proceeds 'top-down', i.e. from the root sequent out to the axiom leaves of the proof.[2] This is done by non-deterministically selecting a rule to be matched[3] against the goal sequent. If a goal sequent matches the axiom rule, it is proven. Where the rule selected is a left or right inference rule, the goal sequent is matched against the conclusion of the rule. If the matching process succeeds, it provides a specification ('instantiation') of the premise(s) of the inference rule, and which then constitute subgoals that must in turn be proven. For the reasons noted above in relation to decidability, the search space for proofs is finite and proof search is guaranteed to terminate.

6.4.2 The problem of multiple equivalent proofs

The calculus **L** typically allows more than one cut-free proof for each possible reading of a sequent. We shall consider some illustrative examples. The two proofs shown in (6.6) without semantics, and in (6.7) with, assign the same reading:

[2] An unfortunate characteristic of this terminology is that, though top-down proof construction runs from root sequent to axioms, for proofs on the page, the root is at the bottom and the axiom leaves at the top.

[3] This notion of 'matching' should be intuitively clear to the reader. We leave it undefined.

(6.6)

$$\dfrac{\dfrac{z \Rightarrow z \qquad y \Rightarrow y}{y/z,\ z \Rightarrow y}[/L] \qquad x \Rightarrow x}{x/y,\ y/z,\ z \Rightarrow x}[/L] \qquad\qquad \dfrac{z \Rightarrow z \qquad \dfrac{y \Rightarrow y \qquad x \Rightarrow x}{x/y,\ y \Rightarrow x}[/L]}{x/y,\ y/z,\ z \Rightarrow x}[/L]$$

(6.7)

$$\dfrac{\dfrac{z{:}h \Rightarrow z{:}h \qquad y{:}gh \Rightarrow y{:}gh}{y/z{:}g,\ z{:}h \Rightarrow y{:}gh}[/L] \qquad x{:}fgh \Rightarrow x{:}fgh}{x/y{:}f,\ y/z{:}g,\ z{:}h \Rightarrow x{:}fgh}[/L] \qquad \dfrac{z{:}h \Rightarrow z{:}h \qquad \dfrac{y{:}gh \Rightarrow y{:}gh \qquad x{:}fgh \Rightarrow x{:}fgh}{x/y{:}f,\ y{:}gh \Rightarrow x{:}fgh}[/L]}{x/y{:}f,\ y/z{:}g,\ z{:}h \Rightarrow x{:}fgh}[/L]$$

Here we see that the operations involved in 'building' some argument type (here y) may be performed either before or after the left inference on the functor requiring that argument (here x/y).[4]

Another case of distinct proofs assigning the same meaning is shown in (6.8) and (6.9):

(6.8)

$$\dfrac{y \Rightarrow y \qquad \dfrac{\dfrac{z \Rightarrow z \qquad \dfrac{x \Rightarrow x \qquad w \Rightarrow w}{x,\ w\backslash x \Rightarrow w}[\backslash L]}{x,\ w\backslash x/z,\ z \Rightarrow w}[/L]}{x,\ w\backslash x/z \Rightarrow w/z}[/R]}{x,\ y,\ w\backslash x/z\backslash y \Rightarrow w/z}[\backslash L] \qquad \dfrac{\dfrac{y \Rightarrow y \qquad \dfrac{z \Rightarrow z \qquad \dfrac{x \Rightarrow x \qquad w \Rightarrow w}{x,\ w\backslash x \Rightarrow w}[\backslash L]}{x,\ w\backslash x/z,\ z \Rightarrow w}[/L]}{x,\ y,\ w\backslash x/z\backslash y,\ z \Rightarrow w}[\backslash L]}{x,\ y,\ w\backslash x/z\backslash y \Rightarrow w/z}[/R]$$

(6.9)

$$\dfrac{y{:}g \Rightarrow y{:}g \qquad \dfrac{\dfrac{z{:}i \Rightarrow z{:}i \qquad \dfrac{x{:}f \Rightarrow x{:}f \quad w{:}kgif \Rightarrow w{:}kgif}{x{:}f,\ w\backslash x{:}kgi \Rightarrow w{:}kgif}[\backslash L]}{x{:}f,\ w\backslash x/z{:}kg,\ z{:}i \Rightarrow w{:}kgif}[/L]}{x{:}f,\ w\backslash x/z{:}kg \Rightarrow w/z{:}\lambda i.kgif}[/R]}{x{:}f,\ y{:}g,\ w\backslash x/z\backslash y{:}k \Rightarrow w/z{:}\lambda i.kgif}[\backslash L] \qquad \dfrac{\dfrac{y{:}g \Rightarrow y{:}g \qquad \dfrac{z{:}i \Rightarrow z{:}i \qquad \dfrac{x{:}f \Rightarrow x{:}f \quad w{:}kgif \Rightarrow w{:}kgif}{x{:}f,\ w\backslash x{:}kgi \Rightarrow w{:}kgif}[\backslash L]}{x{:}f,\ w\backslash x/z{:}kg,\ z{:}i \Rightarrow w{:}kgif}[/L]}{x{:}f,\ y{:}g,\ w\backslash x/z\backslash y{:}kg,\ z{:}i \Rightarrow w{:}kgif}[\backslash L]}{x{:}f,\ y{:}g,\ w\backslash x/z\backslash y{:}k \Rightarrow w/z{:}\lambda i.kgif}[/R]$$

These proofs show a case where a right inference can equally well be made either before or after some unrelated left inference. A final example:

(6.10)

$$x/y \Rightarrow x/y \qquad\qquad \dfrac{\dfrac{y \Rightarrow y \qquad x \Rightarrow x}{x/y,\ y \Rightarrow x}[/L]}{x/y \Rightarrow x/y}[/R]$$

(6.11)

$$x/y{:}f \Rightarrow x/y{:}f \qquad\qquad \dfrac{\dfrac{y{:}g \Rightarrow y{:}g \qquad x{:}fg \Rightarrow x{:}fg}{x/y{:}f,\ y{:}g \Rightarrow x{:}fg}[/L]}{x/y{:}f \Rightarrow x/y{:}\lambda g.fg}[/R]$$

Here we see that where the type instantiating an axiom is a function, it is also possible to 'unpack' the type using right and left inferences, and then 'equate' subformulas of the original function in further axiom instances. The proof terms assigned by these two proofs are equivalent under η-reduction.

[4] The meaning of 'before' and 'after' as used here should be clear. Proof construction proceeds from root sequent to axiom leaves. Some inference A occurs 'after' some other B if A occurs in a subproof of one of the premises of B. Similarly, A occurs 'before' B if B occurs in a subproof of one of the premises of A.

The existence of multiple equivalent proofs presents a problem for efficient theorem proving based on the calculus **L**. Search for proofs must be exhaustive to ensure that all different readings for a given sequent are found, and a naive theorem prover will expend considerable effort constructing proofs that assign the same meaning. This radically reduces the efficiency of Lambek calculus theorem proving.

As a concrete example, consider the case of sequents of the form:

$$\mathrm{x}_1/\mathrm{x}_2,\ \mathrm{x}_2/\mathrm{x}_3,\ \mathrm{x}_3/\mathrm{x}_4,\ \cdots,\ \mathrm{x}_n \Rightarrow \mathrm{x}_1$$

where each is x_i is atomic. Such sequents are semantically unambiguous, i.e. there is only one reading for the combination. Proofs of such sequents involve only [/L] inferences and axioms. However, ambiguity for the selection of the active type for each left inference is such that the number of equivalent proofs grows rapidly with increasing n. In fact for this particular case, the number of equivalent proofs is given by the Catalan series, which is such that $catalan(n) \geq 2^{n-2}$, and for $n > 6$, $catalan(n) > 2^n$:

n	=	2	3	4	5	6	7	8	9
$catalan(n)$	=	1	2	5	14	42	132	429	1430

6.5 Solutions to the problem of multiple equivalent proofs

We will consider three approaches to handling the problem of multiple equivalent proofs as it arises for parsing the implicational fragment of the Lambek calculus. As we shall see, the three approaches are closely related in a number of regards. Firstly, each solution is based around deriving a notion of *normal form* for proofs, an approach to handling the problem of multiple equivalent proofs for flexible CG parsing first proposed by Hepple and Morrill (1989). Secondly, the notions of NF proof given in the three approaches are related in that the structure of NF proofs can be seen to mirror the structure of the lambda expression that specifies the meaning assigned by the proof.

The problem of multiple equivalent proofs is not restricted to the Lambek calculus. It arises for other flexible CG approaches also, including the Combinatory Categorial Grammar (CCG) approach developed by Steedman (1987), Szabolcsi (1987b) and others. Hepple and Morrill (1989) suggest a solution to this problem as it arises for CCG which is based around the use of NFs. It is a common state of affairs for some terms of a language (in the broadest sense of the word) to be equivalent under some relevant notion of equality. Under such circumstances it can be useful to specify NF terms to act as privileged representatives of their equivalence class. For the case of CG parsing, the 'terms' that concern us are proofs for combinations, and the relevant notion of equivalence is the semantic equivalence of the readings assigned by these proofs, or to speak more precisely, extensional equivalence of the lambda expressions that state the reading assigned by proofs. Hepple and Morrill develop a notion of NF for CCG proofs, using the method of *proof reduction*. This notion of NF is used in parsing by adapting the parsing method so that only returns NF proofs are returned. Hepple and Morrill demonstrate that this method is *safe* in the sense that every proof has a semantically equivalent NF proof, so that a parser which generates all and only the NF proofs for a combination must thereby return all possible readings that can be assigned.

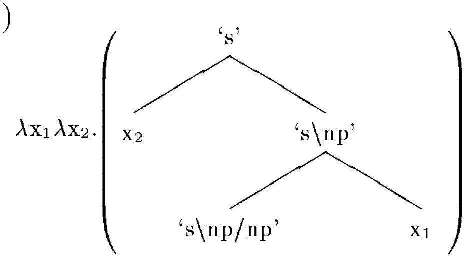
6.6 König's method

König (1989) describes a solution to the problem of multiple equivalent proofs for the Lambek calculus which, as we have noted, involves defining a NF for sequent proofs, and then adopting a parsing method which is intended to generate only such NF proofs.

König's method for defining NF proofs involves firstly mapping from sequent proofs to objects which are termed *syntax trees*. Proofs which assign the same proof term map into the same syntax tree, so this first mapping, called Syntax Tree Construction, serves the purpose of defining the equivalence classes of proofs. A second mapping, called Proof Reconstruction, generates for each syntax tree a unique proof, which is the NF exemplar of the relevant equivalence class.

The syntax tree that is constructed for any proof has essentially the same structure as the lambda expression that states the meaning of that proof, with subparts that correspond to each functional application and abstraction in the lambda expression. Lexical types are assigned 'partial' syntax trees. For a basic lexical type, its syntax tree consists of a single node labelled with the name of that type (all such names being put in single quotes). Thus, a lexical type np has a syntax tree consisting of the single node: 'np'. Functional lexical types are mapped onto a binary tree corresponding to what we might call a 'natural projection' of the function, with lambda abstractions over the argument positions. For example, a lexical type s\np/np has the following partial syntax tree:

(6.12)



Using partial syntax trees associated with the lexical types of the goal sequent, the syntax tree associated with some proof of the sequent can be built up alongside the construction of the proof. We saw earlier how the inference rules of **L** can be augmented to construct the proof term for a proof, with each antecedent or succedent element being a pair TYPE:TERM. König's method for syntax tree construction involves stating a version of the **L** inference rules where each element is a pair TYPE:ST (where ST stands for syntax tree), and where syntax trees are combined in a manner precisely analogous to the proof term combination steps in the 'semantic' versions of the **L** rules (where t, t_a, t_v, t_f are partial syntax trees):

(6.13) Axiom: $\text{x}{:}t \Rightarrow \text{x}{:}t$

Right rules:

$$\frac{\Gamma,\ \text{y}{:}t_a \Rightarrow \text{x}{:}t_v}{\Gamma \Rightarrow \text{x/y}{:}\text{`x/y'}(t_v)}[/\text{R}] \qquad \frac{\text{y}{:}t_a,\ \Gamma \Rightarrow \text{x}{:}t_v}{\Gamma \Rightarrow \text{x}\backslash\text{y}{:}\text{`x}\backslash\text{y'}(t_v)}[\backslash\text{R}]$$

Left rules:

$$\frac{\Delta \Rightarrow \text{y}{:}t_a \qquad \Gamma,\ \text{x}{:}t_f[t_a],\ \Lambda \Rightarrow \text{z}{:}t}{\Gamma,\ \text{x/y}{:}t_f,\ \Delta,\ \Lambda \Rightarrow \text{z}{:}t}[/\text{L}] \qquad \frac{\Delta \Rightarrow \text{y}{:}t_a \qquad \Gamma,\ \text{x}{:}t_f[t_a],\ \Lambda \Rightarrow \text{z}{:}t}{\Gamma,\ \Delta,\ \text{x}\backslash\text{y}{:}t_f,\ \Lambda \Rightarrow \text{z}{:}h}[\backslash\text{L}]$$

The step associated with left rules involves 'applying' one syntax tree to the other (where the notation $t_f[t_a]$ corresponds to application, with an implicit step of β-reduction, as required). For the right rule, instead of performing lambda abstraction as in the proof term construction step, the syntax tree 'x/y'(t_v) (or 'x\y'(t_v)) is formed, recording information of the succedent type that would not be encoded by the lambda abstraction. Thus, the structure of the syntax tree assigned by a proof is closely analogous to the proof term it would assign, the former differing from the latter, however, in that it records information of the types combined and their linear ordering. Hence, different proof terms correspond to structurally distinct syntax trees. The additional information that syntax trees provide is required by the Proof Reconstruction procedure, being sufficient to allow a complete proof for the original sequent to be constructed, which is the NF member of the equivalence class.

The Proof Reconstruction procedure generates NF proofs whose structure closely mirrors the structure of the input syntax tree, and hence in turn, given the correspondence of syntax trees and proof terms, mirrors the structure of the proof term which gives the reading assigned by the proof. In Proof Reconstruction, the NF proof is built during a recursive traversal of a syntax tree. We shall not consider the details of the the Proof Reconstruction method, or the relation between the form of a syntax tree and its corresponding NF proof. We shall instead discuss the form of NF proofs directly in relation to proof terms. We noted that cut-free proofs in **L** assign proof terms that are in β-NF. A β-NF lambda expression is always of the form:

$$\lambda v_1..v_n.(hU_1..U_m) \qquad (n, m \geq 0)$$

where h is a variable and each U_i is in turn of this stated form (which follows given the absence of β-redexes). The structure of a NF proof **P** mirrors the structure of its proof term as follows: Looking at the root sequent of **P**, proceeding upwards, we find that the proof begins with a series of n right inferences, corresponding semantically to the lambda abstraction of the variables $v_1..v_n$. At the end of these inferences is a subproof which assigns the reading $(hU_1..U_m)$, whose main branch consists of m left inferences, terminating in an axiom instance. The first of these left inferences has for its active type the term whose semantics is the variable h and semantically performs the application of h to U_1 (the term U_1 being constructed in the subproof for the minor premise). The next left inference on the main branch has for its active type the value subpart of the active type for the previous left inference, and semantically performs the application of hU_1 to U_2. The ensuing left inferences proceed similarly until the axiom is reached. The subproof for the minor premise of each left inference is also of this general form, mirroring the structure of the subterm U_i returned for its meaning.

König's parsing method is based around the following set of *nesting constraints*, which are conditions on rule use during top-down theorem proving, and which are derived by examining the general form of NF proofs.

(6.14) NESTING CONSTRAINTS:

1. Right Rule Preference: Complex succedent types are selected as active type (or in König's terms, *current functor*) in preference to antecedent types.
2. Current Functor Unfolding: Once an antecedent is chosen as active type, it should be completely 'unfolded', i.e. its value subtype should be the active type for the next inference, unless it is an atomic type.
3. Goal Criterium: An antecedent may be selected as active type only if its 'innermost' value type is the same as ('is unifiable with') the succedent of the sequent.

The theorem proving algorithm is informally stated as follows:

- A sequent is proved if it is an instance of the axiom scheme.
- Otherwise, choose an inference rule in accordance with the nesting constraints, and try to prove its premise(s).

6.6.1 Discussion of König's method

We next consider the problems that arise for König's method. These problems arise both in respect of König's NF system and in the parsing method that König derives from it. Note that in judging the adequacy of König's NF system, and the alternatives described later, we must bear in mind that the intention is to use it as a basis for a parsing/theorem proving method that returns just these proofs. In judging the associated parsing method, the criterion is simply whether it is successful in returning all and only the NF proofs.

Separately from any particular NF based approach, we can see that there are certain requirements that a NF system should satisfy to be suitable for use in handling the problem of multiple equivalent proofs. Firstly, and most importantly, parsing based on some NF system should be *safe* in the sense that, in limiting our attention to only NF proofs, we will still determine all the distinct readings that can be assigned to a given sequent. This is the case provided that (a) every proof has a NF, and (b) the NF(s) of any proof are semantically equivalent to it. Requirement (b) is evidently satisfied by König's method, since the NF proof generated by the Proof Reconstruction procedure from some syntax tree is one which gives rise to the same syntax tree. Hence, each proof and its NF must assign identical proof terms. However, König's method fails in respect of requirement (a). We saw above that the types in the initial goal sequent ('lexical types') are assigned *partial* syntax trees, which may involve lambda operators binding variables corresponding to unfilled nodes in the syntax tree. König's augmented **L** inference rules, which construct syntax trees, are such that some proofs will be assigned syntax trees that include lambda operators. However, the Proof Reconstruction procedure (which we have not stated) applies only to 'non-partial' syntax trees, i.e. those which do not include lambda operators, and so proofs whose syntax trees include lambda operators have *no* NF under König's method.

For example, consider the simple case of a proof consisting of just a single axiom instance instantiated with the functional type s\np/np, i.e:

$$\mathrm{s\backslash np/np} \Rightarrow \mathrm{s\backslash np/np}$$

The antecedent type of this sequent is initially assigned the partial syntax tree involving lambda operators shown in (6.12). Under the augmented inference rules in (6.13), this syntax tree is also that returned for the succedent type (and hence is the syntax tree for the entire proof). Proof Reconstruction cannot apply to this syntax tree, and so this proof has no NF. Since not all proofs have NFs, König's NF method does not guarantee that all distinct readings that can be assigned for a sequent are returned.

A strongly desirable property for a NF system used for present purposes is that there should be *exactly one* NF proof corresponding to each set of extensionally equivalent proofs, since then parsing will be *optimal* in the sense that the parser will need to generate only one proof for each distinct reading returned. König's method for distinguishing equivalence classes of proofs rests on syntax tree construction. We saw that cut free proofs in **L** return proof terms that are in β-NF, but that extensional equivalence for the lambda calculus is given by the $\beta\eta$-reduction system. Hence, proofs which assign extensionally equivalent meanings may give distinct proof terms — terms which reduce to a common term under η-reduction. Proofs which assign distinct proof terms fall into distinct equivalence classes under König's system and so have distinct NFs. Thus, König's NF system fails in respect of a criterion of uniqueness of proofs with respect to readings.

We next consider the parsing method derived from König's notion of NF. It appears that this method is effective when used together with a unification based version of the Lambek calculus that König describes. However, as König points out, the method allows the derivation of non-NF proofs when used with standard (propositional) versions of the Lambek calculus. We can see this problem by considering the following two proofs of the sequent x/x, x/y, y ⇒ x:

(6.15) (a)

$$\dfrac{\dfrac{\mathrm{y} \Rightarrow \mathrm{y} \qquad \mathrm{x} \Rightarrow \mathrm{x}}{\mathrm{x/y,\ y} \Rightarrow \mathrm{x}}[/L] \qquad \mathrm{x} \Rightarrow \mathrm{x}}{\mathrm{x/x,\ x/y,\ y} \Rightarrow \mathrm{x}}[/L]$$

(b)

$$\dfrac{\mathrm{y} \Rightarrow \mathrm{y} \qquad \dfrac{\mathrm{x} \Rightarrow \mathrm{x} \qquad \mathrm{x} \Rightarrow \mathrm{x}}{\mathrm{x/x,\ x} \Rightarrow \mathrm{x}}[/L]}{\mathrm{x/x,\ x/y,\ y} \Rightarrow \mathrm{x}}[/L]$$

The first proof is in NF. The second isn't but is still admitted given the Nesting Constraints. For the non-NF proof, we can see that starting with the root sequent, the Goal Criterium allows that either x/x or x/y is chosen as the active type, and the choice of x/y is not compatible with a NF proof. König's parsing rules allow that after an inappropriate antecedent current functor has been chosen, and some part of the proof constructed, an alternative current functor can be selected, allowing completion of the proof.

We saw earlier that the main branch of a NF proof should begin with a sequence of right inferences (corresponding to the outermost lambda abstractions in the proof term). This is followed by some sequence of left inferences, terminating in a axiom instance, where the active type for $(n{+}1)$th inference must be the value subpart of the functor that was active type for the nth inference. König's Nesting Constraints require that an antecedent active type once selected, must be completely 'unfolded', which accords with these observations about the structure of NF proofs. However, these constraints allow a new antecedent current functor to be selected once one has been completely unfolded, which clearly contradicts our observations about the structure of NF proofs, and so the Nesting Constraints are clearly too weak. Evidently, this weakness could be remedied by adding a further element of 'commitment', so that a new antecedent active type could *not* be selected for completing the main branch of a proof once another one had been completely unfolded.

A further point of note is that König's parsing method places some constraints on admitted proofs that are not present in the NF system itself. Thus, the first step in proving any goal or subgoal sequent is to check whether it is an instance of the axiom scheme, and if it is, the proof is taken to be complete. Hence, a sequent $s\backslash np/np \Rightarrow s\backslash np/np$ receives only the proof where it is taken to be an instance of the axiom scheme. The following alternative is not returned:

(6.16)
$$\dfrac{\dfrac{\dfrac{np \Rightarrow np \qquad \dfrac{np \Rightarrow np \quad s \Rightarrow s}{np,\ s\backslash np \Rightarrow s}[\backslash L]}{np,\ s\backslash np/np,\ np \Rightarrow s}[/L]}{s\backslash np/np,\ np \Rightarrow s\backslash np}[\backslash R]}{s\backslash np/np \Rightarrow s\backslash np/np}[/R]$$

This asymmetry is not present in the NF system. If anything the opposite holds. As we saw earlier, the proof of this sequent consisting of just an axiom instance has no corresponding NF under König's system, whilst the proof in (6.16) is admitted as a NF proof under König's NF system but is excluded by the parsing method.

6.7 Moortgat's method

We next consider an approach to handling the problem of multiple equivalent proofs due to Moortgat (1990a), which is based around a method of *partial deduction*. This involves performing a process of *partially executing* the types of the initial goal sequent with respect to the inference rules of **L** to arrive at a set of *derived* inference rules.

This method is most easily explained in relation to concrete examples. Partial execution involves unfolding a type with respect to the **L** inference rules. For example, a type $a\backslash b/c$ can be unfolded using the left rules by first taking it to be the active type of the [/L] rule, giving the instantiation of the rule shown in (6.17a):

(6.17)(a)
$$\dfrac{B \Rightarrow c \qquad A,\ a\backslash b,\ C \Rightarrow Z}{A,\ a\backslash b/c,\ B,\ C \Rightarrow Z}[/L]$$

(b)
$$\dfrac{C \Rightarrow c \qquad \dfrac{B \Rightarrow b \quad A,\ a,\ D \Rightarrow Z}{A,\ B,\ a\backslash b,\ D \Rightarrow Z}[\backslash L]}{A,\ B,\ a\backslash b/c,\ C,\ D \Rightarrow Z}[/L]$$

(We use letters A, B, ..., H, as variables over sequences.) Unfolding requires that we hypothesize context material around the unfolded type, here realized as the variables A, B, C. The major premise of (6.17a) specifies an antecedent type that can be further unfolded, with respect to [\L]. Adding this step, we arrive at (6.17b) (unifying through the constraints that the step adds to the requirements on context, and renaming context variables). The major premise of the second inference can be matched against the axiom scheme, requiring the context variables to be empty sequences. This gives the final partial execution structure shown in (6.18a). Eliminating irrelevant structure and any leaves that do not correspond to unresolved subgoals, we arrive at the 'derived' inference rule shown in (6.18b).

(6.18)(a)
$$\frac{B \Rightarrow c \qquad \dfrac{A \Rightarrow b \quad a \Rightarrow a}{A,\ a\backslash b \Rightarrow a}[\backslash L]}{A,\ a\backslash b/c,\ B \Rightarrow a}[/L]$$
(b)
$$\frac{B \Rightarrow c \quad A \Rightarrow b}{A,\ a\backslash b/c,\ B \Rightarrow a}$$

Unfolding an atomic type x is taken to yield a 'derived' version of the axiom scheme $x \Rightarrow x$.

Let us consider a simple example. Given an initial goal sequent:

$$b,\ a\backslash b/c,\ c \Rightarrow a$$

partial deduction of the antecedent types yields the following collection of derived inference rules:

(6.19)(a)
$$\frac{B \Rightarrow c \quad A \Rightarrow b}{A,\ a\backslash b/c,\ B \Rightarrow a}$$
(b) $b \Rightarrow b$ (c) $c \Rightarrow c$

The sequent is then proven using just these rules. Only rule (6.19a) matches the initial goal sequent. Matching gives rise to two further subgoals: $b \Rightarrow b$ and $c \Rightarrow c$, which match the derived axioms (6.19b) and (6.19c). These rules have no subgoals to be proven, and so proving the initial sequent is complete.

The method is made slightly more complicated by the need to allow for *higher order* types, which require that partial deduction be made with respect to right inferences. This is illustrated by the unfolding of the type $s\backslash(s/(s\backslash np))$ in (6.20a), which gives the derived rule shown in (6.20b).

(6.20)(a)
$$\frac{\dfrac{A,\ s\backslash np \Rightarrow s}{A \Rightarrow s/(s\backslash np)}[/R] \qquad s \Rightarrow s}{A,\ s\backslash(s/(s\backslash np)) \Rightarrow s}[\backslash L]$$
(b)
$$\frac{A,\ s\backslash np \Rightarrow s}{A,\ s\backslash(s/(s\backslash np)) \Rightarrow s}$$
(c)
$$\frac{B \Rightarrow np}{B,\ s\backslash np \Rightarrow s}$$
(d) $np \Rightarrow np$

The appearance of the type $s\backslash np$ as an antecedent of the subgoal of the derived rule (6.20b) means that partial deduction must be made also with respect to this type, giving the derived rule shown in (6.20c). For proving the sequent:

$$np,\ s\backslash(s/(s\backslash np)) \Rightarrow s$$

partial deduction yields the derived inference rules (6.20b) and (6.20c), together with the derived axiom (6.20d). A proof for the sequent is as follows (where the rule uses are labelled with (b) (c) or (d), as specified in (6.20)):

(6.21)
$$\frac{\dfrac{\dfrac{}{np \Rightarrow np}(d)}{np,\ s\backslash np \Rightarrow s}(c)}{np,\ s\backslash(s/(s\backslash np)) \Rightarrow s}(b)$$

6.7.1 Discussion of Moortgat's method

Firstly, let us consider the form of proofs that result under the partial deduction method, and their relation to the associated proof term. Note that the implicit structure of Moortgat's proofs is obscured by the step of 'flattening out' the structures that result after unfolding. This step, however, is by no means necessary. Instead of the flattened out derived rule in (6.18b), we could equally well use the structure in (6.18a) as the derived rule, which has the same two as yet unfulfilled subgoals as (6.18b). Then, proof construction using the derived rules would yield structures that precisely corresponded to proofs in **L**, except that fewer inference steps would be required since the derived rules collapse together several uses of **L** inference rules.

When we view the proofs that result under Moortgat's method in this way, we can see that they are closely related to König's NF proofs. We saw earlier that König's NF proofs are such that their main branch begins with some sequence of zero or more right inferences, followed by some sequence of left inferences, terminating in an axiom instance, where the value subpart of the active type for the nth left inference must be the active type for the $(n+1)$th inference.

Moortgat's procedure of partial deduction with respect to right inferences also has the effect that right inferences are ordered before left inferences on the main branch of any subproof. Furthermore, the unfolding of antecedent functors (as in (6.18a)) ensures that the latter part of the main branch of any subproof consists of some left inferences terminating in an axiom instance, where these steps serve to 'unfold' a single functional type. The derived inference rule in (6.18b) provides only sufficient surrounding 'context' to construct the arguments of the function, with the effect that the derived inference rule can only be used where the unfolded function has widest scope in the proof term assigned for the subproof. In these respects, Moortgat's proofs closely mimic König's. However, Moortgat's unfolding procedure for antecedent functors requires that they be completely unfolded, with the result that only axiom scheme instances that are instantiated with atomic types are allowed. This characteristic of Moortgat's 'normal form' proofs is not shared by König's NFs.

We next consider some criticisms and problems of Moortgat's method. A major criticism of Moortgat's approach is that his partial execution procedure places various restrictions on the consequences of the unfolding process which implicitly encode (intuitively sensible) ideas about NF proofs which are neither a natural consequence of the (general) partial deduction method, nor receive independent justification in their own terms.

For example, consider the partial unfolding of the type a\b/c discussed in relation to the diagrams in (6.17) and (6.18). The type was first unfolded under two left inferences to give the structure in (6.17b). Then, its unresolved major premise was matched against the axiom scheme, giving the structure in (6.18b), which is the essence of the final derived inference rule. We may ask, however, why this final step of matching the major premise against the axiom scheme should have been done. The structure (6.17b) that we had before this final step corresponds to a perfectly valid 'derived' inference rule, i.e:

(6.22) $$\frac{\text{C} \Rightarrow \text{c} \qquad \text{B} \Rightarrow \text{b} \qquad \text{A, a, D} \Rightarrow \text{Z}}{\text{A, B, a}\backslash\text{b/c, C, D} \Rightarrow \text{Z}}$$

Furthermore, either of the unresolved minor premises of (6.17b) could equally well have been matched against the axiom scheme. If, for example, we matched both of these, but not the major premise, we would arrive at the structure (6.23a), which simplifies to the derived inference rule in (6.23b). Again, the derived inference rule that results is perfectly valid.

(6.23)(a) $$\frac{\text{c} \Rightarrow \text{c} \qquad \dfrac{\text{b} \Rightarrow \text{b} \qquad \text{A, a, D} \Rightarrow \text{Z}}{\text{A, b, a}\backslash\text{b, D} \Rightarrow \text{Z}}[\backslash\text{L}]}{\text{A, b, a}\backslash\text{b/c, c, D} \Rightarrow \text{Z}}[/\text{L}]$$

(b) $$\frac{\text{A, a, D} \Rightarrow \text{Z}}{\text{A, b, a}\backslash\text{b/c, c, D} \Rightarrow \text{Z}}$$

In fact, the matching of (only) the major premise of the unfolded structure against the axiom scheme is crucial to the success of Moortgat's method. This has the effect that the resulting derived inference rule only allows inference with respect to an antecedent functor when it has widest scope semantically and ensures that the main branch of the given subproof involves only left inferences that unfold the given type. (Note that the derived inference rule in (6.22) would *not* have this effect.)

Since the completion of partial execution in this way does not follow automatically from the nature of partial execution, it appears that the requirement on partial execution has been added simply because it realizes a sensible intuition about the nature of NF proofs.

A second criticism of Moortgat's approach (as it is presented in Moortgat (1990a)) is that it is not *demonstrated* that the method is safe (i.e. that all distinct readings that can be assigned to a sequent under the rules of **L** are also returned under the partial deduction approach). Similarly, it is not demonstrated that the method is optimal in the sense that only a single proof under the method will return each distinct reading.

A final point is that the addition of a compilation phase to the parsing process, during which the types of the initial goal sequent are partially executed with respect to the inference rules, considerably complicates the overall parsing method, and distances it somewhat inelegantly from the original calculus, obscuring the character of the inference rules as simple steps which govern the behaviour of single connectives. It would seem much better if an adequate approach can be arrived at which avoids the need of a 'compilation phase'. We believe that the approach that is presented in the next section fulfils these requirements.

6.8 Hepple's method

In this section, we describe a NF based method for handling the problem of multiple equivalent proofs that was first presented in Hepple (1990d). The method centres around a *constructive* notion of NF proof, which is closely related to the NF proofs of König's system. A second notion of NF is also defined, using the method of *proof reduction*. This system is shown to be equivalent to the first, and is principally of value in that it provides a basis for demonstrating *completeness*, i.e. that every proof has a (equivalent) NF. Finally, the constructive notion of NF is used as a basis for defining a new calculus, called the *parsing calculus*, for use in theorem proving, which only allows the construction of proofs which correspond to constructive NF proofs.

6.8.1 Normal forms and proof terms

The NF proofs that arise under the NF systems to be described are of essentially the same form as König's NFs, although (like Moortgat) we add a requirement that only axiom leaves instantiated with atomic types are allowed. As with König's and Moortgat's approaches, the structure of our NFs mirrors the form of the proof term they assign. Given the significance of this relationship for what follows, we restate it here.

As we have seen, the meaning assigned by a cut-free proof in **L** is always a lambda term in β-NF, of the form:

$$\lambda v_1..v_n.(hU_1..U_m) \qquad\qquad (n, m \geq 0)$$

where h is a variable and each U_i is also of this general form. NF proofs mirror the form of such ambda terms in that, if we examine the sequence of inferences along the main branch of the proof, beginning at the root, we find firstly a series of (zero or more) right inferences, corresponding semantically to the abstraction of the variables $v_1..v_n$ in the proof term, and then zero or more left inferences, terminating in an axiom instance. These left inferences correspond semantically to the application of the variable h to its arguments $U_1..U_m$, and so the active type for these inferences is the antecedent which has widest scope in the proof term for the immediate subproof (i.e. the antecedent having semantics h).

6.8.2 The headedness of proofs

To enable the definition of NF proofs, it is important that we can identify the antecedent type of a sequent that has widest scope under the ensuing proof. We call this type the *head type* of the sequent under the given proof.[5] It turns out that we can always identify the head type of a sequent under some proof in terms of the form of that proof, i.e. without needing to look at the semantics of the types in the proof.

The main branch axiom plays an important role here, because its semantics is directly linked to the semantics of the proof as a whole. In particular, the proof term associated with the entire proof corresponds to that of the succedent (and hence also the sole antecedent) of the main branch axiom having undergone zero or more lambda abstractions. (This follows given the semantic operations associated with the **L** rules, and can be easily shown by induction.) It follows that the head type of a sequent under some proof is also that type whose semantics has widest scope in the proof term for the antecedent of the main branch axiom. This semantic relation has a syntactic counterpart: for the head H of a sequent under some proof and the sole antecedent A of the proof's main branch axiom, it is always the case that A is a value subformula of H (or is H itself). This point is illustrated by the following proof:

(6.24)

$$\dfrac{\dfrac{\mathrm{z}{:}i \Rightarrow \mathrm{z}{:}i \qquad \dfrac{\mathrm{y}{:}f \Rightarrow \mathrm{y}{:}f \qquad \mathrm{x}{:}gif \Rightarrow \mathrm{x}{:}gif}{\mathrm{y}{:}f,\ \mathrm{x}\backslash\mathrm{y}{:}gi \Rightarrow \mathrm{x}{:}gif}[\backslash L]}{\mathrm{y}{:}f,\ \mathrm{x}\backslash\mathrm{y}/\mathrm{z}{:}g,\ \mathrm{z}{:}i \Rightarrow \mathrm{x}{:}gif}[/L]}{\mathrm{y}{:}f,\ \mathrm{x}\backslash\mathrm{y}/\mathrm{z}{:}g \Rightarrow \mathrm{x}/\mathrm{z}{:}\lambda i.gif}[/R]$$

The head type here is the antecedent x\y/z since its semantic variable g has widest scope in the meaning assigned by the proof. Observe that the type x which instantiates the main branch axiom is the ultimate value subformula of the head type. Not all proofs have a head, as in (6.25), where the variable having widest scope in the meaning assigned by the proof does not correspond to the semantics of either of the root sequent antecedents. Instead, this variable originates with a type that is 'introduced' in a right inference:

(6.25)

$$\dfrac{\dfrac{\dfrac{\mathrm{y}{:}g \Rightarrow \mathrm{y}{:}g \qquad \mathrm{x}{:}fg \Rightarrow \mathrm{x}{:}fg}{\mathrm{x}/\mathrm{y}{:}f,\ \mathrm{y}{:}g \Rightarrow \mathrm{x}{:}fg}[/L] \qquad \mathrm{z}{:}i(fg) \Rightarrow \mathrm{z}{:}i(fg)}{\mathrm{x}/\mathrm{y}{:}f,\ \mathrm{y}{:}g,\ \mathrm{z}\backslash\mathrm{x}{:}i \Rightarrow \mathrm{z}{:}i(fg)}[\backslash L]}{\mathrm{x}/\mathrm{y}{:}f,\ \mathrm{y}{:}g \Rightarrow \mathrm{z}/(\mathrm{z}\backslash\mathrm{x}){:}\lambda i.i(fg)}[/R]$$

This syntactic relation between the head type of a sequent under some proof and the sole antecedent type of the main branch axiom (i.e that the latter is a value part subformula

[5] This use of the term *head* should not be confused with the linguistic use of the same term in the preceding chapters.

of the former) is exploited in the definition in (6.26), which identifies the head type of any proof in terms of the form of the proof. This makes use of some new notational conventions. Firstly, we use numerically subscripted proof symbols (e.g. $\mathbf{P}_3$, $\mathbf{P}_n$, $\mathbf{Q}_m$, etc) to refer to the headedness of proofs. In particular, a proof $\mathbf{P}_n$, $n \geq 1$, is a headed proof, with the head being the nth member of the antecedent sequence of the root sequent. If $n = 0$, then the proof is unheaded. In addition, we use superscripted Greek letters to indicate sequences of given numbers of types, i.e. π^n corresponds to a sequence of n types.

(6.26) Each proof $\mathbf{P}_n$ is of one of the forms:

a. $\mathrm{x} \Rightarrow \mathrm{x}$ where $n = 1$

b. $$\frac{\mathbf{Q}_m(\mathrm{y},\ \pi \Rightarrow \mathrm{x})}{\pi \Rightarrow \mathrm{x}\backslash\mathrm{y}}[\backslash R]$$ where $((m > 0)\ \&\ (n = m - 1))$ or $((m = 0)\ \&\ (n = 0))$

c. $$\frac{\mathbf{Q}_m(\pi^k,\ \mathrm{y} \Rightarrow \mathrm{x})}{\pi^k \Rightarrow \mathrm{x}/\mathrm{y}}[/R]$$ where $((m = k+1)\ \&\ (n = 0))$ or $((m \leq k)\ \&\ (n = m))$

d. $$\frac{\mathbf{R}(\pi^k \Rightarrow \mathrm{y}) \qquad \mathbf{Q}_j(\phi^m,\ \mathrm{x},\ \psi \Rightarrow \mathrm{z})}{\phi^m,\ \pi^k,\ \mathrm{x}\backslash\mathrm{y},\ \psi \Rightarrow \mathrm{z}}[\backslash L]$$ where $((j \leq m)\ \&\ (n = j))$ or $((j > m)\ \&\ (n = j + k))$

e. $$\frac{\mathbf{R}(\pi^k \Rightarrow \mathrm{y}) \qquad \mathbf{Q}_j(\phi^m,\ \mathrm{x},\ \psi \Rightarrow \mathrm{z})}{\phi^m,\ \mathrm{x}/\mathrm{y},\ \pi^k,\ \psi \Rightarrow \mathrm{z}}[/L]$$ where $((j \leq m+1)\ \&\ (n = j))$ or $((j > m+1)\ \&\ (n = j+k))$

The base case for the definition is where a subproof consists of only an axiom inference, in which case the head of the proof is the single antecedent member (and hence, $n = 1$). From there, the position of the head can be kept track of by counting the number of antecedents added in beside the head as subproofs are combined. Note that *every* cut-free proof in **L** is a proof $\mathbf{P}_n$ for *some* value of n.

6.8.3 A constructive notion of normal form proof

We next give a *constructive* definition of NF proof, i.e. we recursively define a set of proofs for the cut-free product-free Lambek calculus which are all and only the *constructive normal form* (CNF) proofs. For this purpose, it is useful to distinguish two subtypes of proof: τ_1 and τ_2. The set of τ_1 proofs is precisely the set of CNF proofs. The τ_2 proofs are a subset of the τ_1 proofs — those whose main branch includes no right inferences.

(6.27) The set of Constructive NF proofs ($\mathcal{T}_1$) is the smallest set such that:

a. *if* x is an atomic type *then* $x \Rightarrow x \ \in \mathcal{T}_2$

b. *if* $\mathbf{P}_{n+1}(\pi^n,\ x,\ \psi \Rightarrow z) \in \mathcal{T}_2$ *and* $\mathbf{Q}(\phi \Rightarrow q) \in \mathcal{T}_1$

$$\textit{then}\ \ \frac{\mathbf{Q}(\phi \Rightarrow q)\quad \mathbf{P}_{n+1}(\pi^n,\ x,\ \psi \Rightarrow z)}{\pi^n,\ x/q,\ \phi,\ \psi \Rightarrow z}[/L] \quad \in \mathcal{T}_2$$

c. *if* $\mathbf{P}_{n+1}(\pi^n,\ x,\ \psi \Rightarrow z) \in \mathcal{T}_2$ *and* $\mathbf{Q}(\phi \Rightarrow q) \in \mathcal{T}_1$

$$\textit{then}\ \ \frac{\mathbf{Q}(\phi \Rightarrow q)\quad \mathbf{P}_{n+1}(\pi^n,\ x,\ \psi \Rightarrow z)}{\pi^n,\ \phi,\ x\backslash q,\ \psi \Rightarrow z}[\backslash L] \quad \in \mathcal{T}_2$$

d. *if* $\mathbf{P} \in \mathcal{T}_2$ *then* $\mathbf{P} \in \mathcal{T}_1$

e. *if* $\mathbf{P}(\pi,\ x \Rightarrow y) \in \mathcal{T}_1$ *then* $\dfrac{\mathbf{P}(\pi,\ x \Rightarrow y)}{\pi \Rightarrow y/x}[/R] \quad \in \mathcal{T}_1$

f. *if* $\mathbf{P}(x,\ \pi \Rightarrow y) \in \mathcal{T}_1$ *then* $\dfrac{\mathbf{P}(x,\ \pi \Rightarrow y)}{\pi \Rightarrow y\backslash x}[\backslash R] \quad \in \mathcal{T}_1$

Given (6.27a), CNF proofs contain only axiom leaves that are instantiated with atomic types. (6.27b) and (6.27c) allow for the occurrence of a left inferences in CNF proofs, and require that a left inference must always be made with respect to the head of the major subproof. Given (6.27d), every $\mathcal{T}_2$ proof is a $\mathcal{T}_1$ proof also (but note that the converse does not hold). Given (6.27e) and (6.27f), only $\mathcal{T}_1$ proofs may have a right rule as the root inference—no $\mathcal{T}_2$ proof is of this form. Since the major subproof of a left inference *must* be a $\mathcal{T}_2$ proof, a right inference may never occur above a left inference on the main branch of a CNF proof. Thus, the main branch of a CNF proof always consists of zero or more right inferences, followed by zero or more left inferences, and terminates in an axiom instance. The minor subproofs of left inferences are themselves $\mathcal{T}_1$ proofs, and so are also of this general form.

As we discussed earlier, for a NF system to be useful for parsing, it must satisfy certain requirements. Most importantly, we want generating only CNF proofs to be *safe* in the sense that our theorem prover returns every distinct reading that can be assigned to a given sequent. Assuming that the theorem prover returns all and only the CNF proofs of a sequent, this is the case if for every proof of a sequent, there exists a semantically equivalent CNF proof. To demonstrate this, a second notion of NF is next defined, based on the method of proof reduction, which is provably equivalent to the constructive notion of NF, and which is also provably complete.

6.8.4 A reductive notion of normal form proof

Reduction and normal forms

Earlier, in relation to the lambda calculus, we discussed how NFs can be defined using the method of reduction. We here restate the main points of this method, as a prelude to defining a reductive notion of NF proof. Firstly, a *contraction* relation ($\triangleright_1$) is defined in terms of a number of contraction rules, each of the form X $\triangleright_1$ Y (in which the form on the left is termed a *redex* and the form on the right its *contractum*). Each contraction rule allows that any term containing an occurrence of a redex may be transformed into a term in which that occurrence is replaced by its contractum. A term is said to be in NF if and only if it contains no occurrences of any redex. The contraction relation generates a *reduction* relation ($\triangleright$) which is such that X *reduces to* Y (X $\triangleright$ Y) if and only if Y is obtained from X by a finite series (possibly zero) of contractions. A term Y is a NF of X if and only if Y is a NF and X $\triangleright$ Y.

Proof reduction and the Lambek calculus

We shall next consider a set of contraction rules stated on Lambek sequent proofs, which together define a reductive notion of NF. A total of eighteen contraction rules are required, which fall into four groups of essentially similar rules. For convenience, we introduce the contraction rules by considering only one member of each group. The full set of eighteen contraction rules is listed at the end of the section.

An instance of the first group of contraction rules is shown in (6.28). We shall refer to these contraction rules as Group A contraction rules (this identity being marked as a subscript on the contraction relation).

$$(6.28) \qquad \mathrm{x/y \Rightarrow x/y} \qquad \triangleright_{1A} \qquad \cfrac{\cfrac{\mathrm{y \Rightarrow y} \qquad \mathrm{x \Rightarrow x}}{\mathrm{x/y,\ y \Rightarrow x}}[/L]}{\mathrm{x/y \Rightarrow x/y}}[/R]$$

This contraction rule expands an axiom leaf instantiated with a functional type to give a subproof containing two axiom leaves, each of which is instantiated with a 'simpler' type than the original axiom (under some metric of simplicity). There is a second rule in this group for which the functional type in the redex is leftward directional.

A Group B contraction rule is shown in (6.29). In the redex, a right inference is applied to the major premise of a left inference. In the contractum, the subproof has been restructured so that the right inference is applied to the root sequent, and the left inference to its sole premise.

$$(6.29)\quad \dfrac{\mathbf{P}(\pi \Rightarrow \mathrm{y}) \qquad \dfrac{\mathbf{Q}(\phi,\ \mathrm{x},\ \psi,\ \mathrm{w} \Rightarrow \mathrm{z})}{\phi,\ \mathrm{x},\ \psi \Rightarrow \mathrm{z/w}}[/R]}{\phi,\ \mathrm{x/y},\ \pi,\ \psi \Rightarrow \mathrm{z/w}}[/L] \quad \rhd_{1B} \quad \dfrac{\dfrac{\mathbf{P}(\pi \Rightarrow \mathrm{y}) \qquad \mathbf{Q}(\phi,\ \mathrm{x},\ \psi,\ \mathrm{w} \Rightarrow \mathrm{z})}{\phi,\ \mathrm{x/y},\ \pi,\ \psi,\ \mathrm{w} \Rightarrow \mathrm{z}}[/L]}{\phi,\ \mathrm{x/y},\ \pi,\ \psi \Rightarrow \mathrm{z/w}}[/R]$$

There are four contraction rules in this group which arise with the directionality of the connectives for the two inferences.

A Group C contraction rule is shown in (6.30). This rule makes use of the subscripted proof notation, allowing us to recognise where a left inference has the head of the subproof as its active type. In the subproof $\mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \psi,\ \mathrm{v},\ \sigma \Rightarrow \mathrm{z})$, the head is the type x. It follows that the type x/y is the head of the entire redex proof, and that v/w is not. We can see that in the redex, a head left inference (i.e a left inference with respect to the head of the given subproof) is applied to the major premise of non-head left inference. In the contractum, the subproof has been restructured so that the head inference is applied to the root sequent, and the non-head inference to its major premise.

$$(6.30)\quad \dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \psi,\ \mathrm{v},\ \sigma \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x/y},\ \pi,\ \psi,\ \mathrm{v},\ \sigma \Rightarrow \mathrm{z}}[/L]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \psi,\ \mathrm{v/w},\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]$$

$$\rhd_{1C} \quad \dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}) \qquad \dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \psi,\ \mathrm{v},\ \sigma \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x},\ \psi,\ \mathrm{v/w},\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \psi,\ \mathrm{v/w},\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]$$

There are eight Group C contraction rules, which arise with the directionality of the two functors and their relative order.

A Group D contraction rule is shown in (6.31). In the redex, a head left inference is applied to the major premise of a non-head left inference, where the latter can be seen to in-part serve the purpose of 'building' the argument required by the head functor. In the contractum, the inferences have been reordered so that the head inference applies to the root sequent, and the non-head inference applies to the minor premise of the head inference.

$$(6.31)\quad \dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{Q}(\pi,\ \mathrm{v},\ \gamma \Rightarrow \mathrm{y}) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \sigma \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v},\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v/w},\ \psi,\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]$$

$$\rhd_{1D} \quad \dfrac{\dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \qquad \mathbf{Q}(\pi,\ \mathrm{v},\ \gamma \Rightarrow \mathrm{y})}{\pi,\ \mathrm{v/w},\ \psi,\ \gamma \Rightarrow \mathrm{y}}[/L] \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \sigma \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v/w},\ \psi,\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]$$

We require four contraction rules of this pattern, which arise with the directionality of the two functors. (Extra cases do not arise with the relative order of the two functors since it is constrained by the directionality of the head functor).

We call any proof that contains no redexes (which is consequently a NF under the reduction system) an *irreducible normal form* (INF). A complete list of the eighteen contraction rules follows:

(6.32)(a)
$$\mathrm{x/y \Rightarrow x/y} \quad \rhd_{1A} \quad \dfrac{\dfrac{\mathrm{y \Rightarrow y} \qquad \mathrm{x \Rightarrow x}}{\mathrm{x/y,\ y \Rightarrow x}}[/L]}{\mathrm{x/y \Rightarrow x/y}}[/R]$$

(b)
$$\mathrm{x\backslash y \Rightarrow x\backslash y} \quad \rhd_{1A} \quad \dfrac{\dfrac{\mathrm{y \Rightarrow y} \qquad \mathrm{x \Rightarrow x}}{\mathrm{y,\ x\backslash y \Rightarrow x}}[\backslash L]}{\mathrm{x\backslash y \Rightarrow x\backslash y}}[\backslash R]$$

(6.33)(a)
$$\dfrac{\mathbf{P}(\pi \Rightarrow \mathrm{y}) \qquad \dfrac{\mathbf{Q}(\phi,\ \mathrm{x},\ \psi,\ \mathrm{w} \Rightarrow \mathrm{z})}{\phi,\ \mathrm{x},\ \psi \Rightarrow \mathrm{z/w}}[/R]}{\phi,\ \mathrm{x/y},\ \pi,\ \psi \Rightarrow \mathrm{z/w}}[/L] \quad \rhd_{1B} \quad \dfrac{\dfrac{\mathbf{P}(\pi \Rightarrow \mathrm{y}) \qquad \mathbf{Q}(\phi,\ \mathrm{x},\ \psi,\ \mathrm{w} \Rightarrow \mathrm{z})}{\phi,\ \mathrm{x/y},\ \pi,\ \psi,\ \mathrm{w} \Rightarrow \mathrm{z}}[/L]}{\phi,\ \mathrm{x/y},\ \pi,\ \psi \Rightarrow \mathrm{z/w}}[/R]$$

(b)
$$\dfrac{\mathbf{P}(\pi \Rightarrow \mathrm{y}) \qquad \dfrac{\mathbf{Q}(\phi,\ \mathrm{x},\ \psi,\ \mathrm{w} \Rightarrow \mathrm{z})}{\phi,\ \mathrm{x},\ \psi \Rightarrow \mathrm{z/w}}[/R]}{\phi,\ \pi,\ \mathrm{x\backslash y},\ \psi \Rightarrow \mathrm{z/w}}[\backslash L] \quad \rhd_{1B} \quad \dfrac{\dfrac{\mathbf{P}(\pi \Rightarrow \mathrm{y}) \qquad \mathbf{Q}(\phi,\ \mathrm{x},\ \psi,\ \mathrm{w} \Rightarrow \mathrm{z})}{\phi,\ \pi,\ \mathrm{x\backslash y},\ \psi,\ \mathrm{w} \Rightarrow \mathrm{z}}[\backslash L]}{\phi,\ \pi,\ \mathrm{x\backslash y},\ \psi \Rightarrow \mathrm{z/w}}[/R]$$

(c)
$$\dfrac{\mathbf{P}(\pi \Rightarrow \mathrm{y}) \qquad \dfrac{\mathbf{Q}(\mathrm{w},\ \phi,\ \mathrm{x},\ \psi \Rightarrow \mathrm{z})}{\phi,\ \mathrm{x},\ \psi \Rightarrow \mathrm{z\backslash w}}[\backslash R]}{\phi,\ \mathrm{x/y},\ \pi,\ \psi \Rightarrow \mathrm{z\backslash w}}[/L] \quad \rhd_{1B} \quad \dfrac{\dfrac{\mathbf{P}(\pi \Rightarrow \mathrm{y}) \qquad \mathbf{Q}(\mathrm{w},\ \phi,\ \mathrm{x},\ \psi \Rightarrow \mathrm{z})}{\mathrm{w},\ \phi,\ \mathrm{x/y},\ \pi,\ \psi \Rightarrow \mathrm{z}}[/L]}{\phi,\ \mathrm{x/y},\ \pi,\ \psi \Rightarrow \mathrm{z\backslash w}}[\backslash R]$$

(d)
$$\dfrac{\mathbf{P}(\pi \Rightarrow \mathrm{y}) \qquad \dfrac{\mathbf{Q}(\mathrm{w},\ \phi,\ \mathrm{x},\ \psi \Rightarrow \mathrm{z})}{\phi,\ \mathrm{x},\ \psi \Rightarrow \mathrm{z\backslash w}}[\backslash R]}{\phi,\ \pi,\ \mathrm{x\backslash y},\ \psi \Rightarrow \mathrm{z\backslash w}}[\backslash L] \quad \rhd_{1B} \quad \dfrac{\dfrac{\mathbf{P}(\pi \Rightarrow \mathrm{y}) \qquad \mathbf{Q}(\mathrm{w},\ \phi,\ \mathrm{x},\ \psi \Rightarrow \mathrm{z})}{\mathrm{w},\ \phi,\ \pi,\ \mathrm{x\backslash y},\ \psi \Rightarrow \mathrm{z}}[\backslash L]}{\phi,\ \pi,\ \mathrm{x\backslash y},\ \psi \Rightarrow \mathrm{z\backslash w}}[\backslash R]$$

(6.34)(a)
$$\dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \psi,\ \mathrm{v},\ \sigma \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x/y},\ \pi,\ \psi,\ \mathrm{v},\ \sigma \Rightarrow \mathrm{z}}[/L]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \psi,\ \mathrm{v/w},\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]$$
$$\rhd_{1C} \quad \dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}) \qquad \dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \psi,\ \mathrm{v},\ \sigma \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x},\ \psi,\ \mathrm{v/w},\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \psi,\ \mathrm{v/w},\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]$$

(b)
$$\dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \psi,\ \mathrm{v},\ \sigma \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x/y},\ \pi,\ \psi,\ \mathrm{v},\ \sigma \Rightarrow \mathrm{z}}[/L]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \psi,\ \gamma,\ \mathrm{v\backslash w},\ \sigma \Rightarrow \mathrm{z}}[\backslash L]$$
$$\rhd_{1C} \quad \dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}) \qquad \dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \psi,\ \mathrm{v},\ \sigma \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x},\ \psi,\ \gamma,\ \mathrm{v\backslash w},\ \sigma \Rightarrow \mathrm{z}}[\backslash L]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \psi,\ \gamma,\ \mathrm{v\backslash w},\ \sigma \Rightarrow \mathrm{z}}[/L]$$

(c)

$$\dfrac{\mathbf{R}(\gamma \Rightarrow w) \quad \dfrac{\mathbf{Q}(\pi \Rightarrow y) \quad \mathbf{P}_{n+1}(\phi^n, x, \psi, v, \sigma \Rightarrow z)}{\phi^n, \pi, x\backslash y, \psi, v, \sigma \Rightarrow z}[\backslash L]}{\phi^n, \pi, x\backslash y, \psi, v/w, \gamma, \sigma \Rightarrow z}[/L]$$

$$\rhd_{1C} \quad \dfrac{\mathbf{Q}(\pi \Rightarrow y) \quad \dfrac{\mathbf{R}(\gamma \Rightarrow w) \quad \mathbf{P}_{n+1}(\phi^n, x, \psi, v, \sigma \Rightarrow z)}{\phi^n, x, \psi, v/w, \gamma, \sigma \Rightarrow z}[/L]}{\phi^n, \pi, x\backslash y, \psi, v/w, \gamma, \sigma \Rightarrow z}[\backslash L]$$

(d)

$$\dfrac{\mathbf{R}(\gamma \Rightarrow w) \quad \dfrac{\mathbf{Q}(\pi \Rightarrow y) \quad \mathbf{P}_{n+1}(\phi^n, x, \psi, v, \sigma \Rightarrow z)}{\phi^n, \pi, x\backslash y, \psi, v, \sigma \Rightarrow z}[/L]}{\phi^n, \pi, x\backslash y, \psi, \gamma, v\backslash w, \sigma \Rightarrow z}[\backslash L]$$

$$\rhd_{1C} \quad \dfrac{\mathbf{Q}(\pi \Rightarrow y) \quad \dfrac{\mathbf{R}(\gamma \Rightarrow w) \quad \mathbf{P}_{n+1}(\phi^n, x, \psi, v, \sigma \Rightarrow z)}{\phi^n, x, \psi, \gamma, v\backslash w, \sigma \Rightarrow z}[\backslash L]}{\phi^n, \pi, x\backslash y, \psi, \gamma, v\backslash w, \sigma \Rightarrow z}[\backslash L]$$

(e)

$$\dfrac{\mathbf{R}(\gamma \Rightarrow w) \quad \dfrac{\mathbf{Q}(\pi \Rightarrow y) \quad \mathbf{P}_{n+m+2}(\phi^n, v, \psi^m, x, \sigma \Rightarrow z)}{\phi^n, v, \psi^m, x/y, \pi, \sigma \Rightarrow z}[/L]}{\phi^n, v/w, \gamma, \psi^m, x/y, \pi, \sigma \Rightarrow z}[/L]$$

$$\rhd_{1C} \quad \dfrac{\mathbf{Q}(\pi \Rightarrow y) \quad \dfrac{\mathbf{R}(\gamma \Rightarrow w) \quad \mathbf{P}_{n+m+2}(\phi^n, v, \psi^m, x, \sigma \Rightarrow z)}{\phi^n, v/w, \gamma, \psi^m, x, \sigma \Rightarrow z}[/L]}{\phi^n, v/w, \gamma, \psi^m, x/y, \pi, \sigma \Rightarrow z}[/L]$$

(f)

$$\dfrac{\mathbf{R}(\gamma \Rightarrow w) \quad \dfrac{\mathbf{Q}(\pi \Rightarrow y) \quad \mathbf{P}_{n+m+2}(\phi^n, v, \psi^m, x, \sigma \Rightarrow z)}{\phi^n, v, \psi^m, x/y, \pi, \sigma \Rightarrow z}[/L]}{\phi^n, \gamma, v\backslash w, \psi^m, x/y, \pi, \sigma \Rightarrow z}[\backslash L]$$

$$\rhd_{1C} \quad \dfrac{\mathbf{Q}(\pi \Rightarrow y) \quad \dfrac{\mathbf{R}(\gamma \Rightarrow w) \quad \mathbf{P}_{n+m+2}(\phi^n, v, \psi^m, x, \sigma \Rightarrow z)}{\phi^n, \gamma, v\backslash w, \psi^m, x, \sigma \Rightarrow z}[\backslash L]}{\phi^n, \gamma, v\backslash w, \psi^m, x/y, \pi, \sigma \Rightarrow z}[/L]$$

(g)

$$\dfrac{\mathbf{R}(\gamma \Rightarrow w) \quad \dfrac{\mathbf{Q}(\pi \Rightarrow y) \quad \mathbf{P}_{n+m+2}(\phi^n, v, \psi^m, x, \sigma \Rightarrow z)}{\phi^n, v, \psi^m, \pi, x\backslash y, \sigma \Rightarrow z}[\backslash L]}{\phi^n, v/w, \gamma, \psi^m, \pi, x\backslash y, \sigma \Rightarrow z}[/L]$$

$$\rhd_{1C} \quad \dfrac{\mathbf{Q}(\pi \Rightarrow y) \quad \dfrac{\mathbf{R}(\gamma \Rightarrow w) \quad \mathbf{P}_{n+m+2}(\phi^n, v, \psi^m, x, \sigma \Rightarrow z)}{\phi^n, v/w, \gamma, \psi^m, x, \sigma \Rightarrow z}[/L]}{\phi^n, v/w, \gamma, \psi^m, \pi, x\backslash y, \sigma \Rightarrow z}[\backslash L]$$

(h)

$$\dfrac{\mathbf{R}(\gamma \Rightarrow w) \quad \dfrac{\mathbf{Q}(\pi \Rightarrow y) \quad \mathbf{P}_{n+m+2}(\phi^n, v, \psi^m, x, \sigma \Rightarrow z)}{\phi^n, v, \psi^m, \pi, x\backslash y, \sigma \Rightarrow z}[\backslash L]}{\phi^n, \gamma, v\backslash w, \psi^m, \pi, x\backslash y, \sigma \Rightarrow z}[\backslash L]$$

$$\rhd_{1C} \quad \dfrac{\mathbf{Q}(\pi \Rightarrow y) \quad \dfrac{\mathbf{R}(\gamma \Rightarrow w) \quad \mathbf{P}_{n+m+2}(\phi^n, v, \psi^m, x, \sigma \Rightarrow z)}{\phi^n, \gamma, v\backslash w, \psi^m, x, \sigma \Rightarrow z}[\backslash L]}{\phi^n, \gamma, v\backslash w, \psi^m, \pi, x\backslash y, \sigma \Rightarrow z}[\backslash L]$$

(6.35)(a)

$$\frac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{Q}(\pi, \mathrm{v}, \gamma \Rightarrow \mathrm{y}) \quad \mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \sigma \Rightarrow \mathrm{z})}{\phi^n, \mathrm{x/y}, \pi, \mathrm{v}, \gamma, \sigma \Rightarrow \mathrm{z}}[/L]}{\phi^n, \mathrm{x/y}, \pi, \mathrm{v/w}, \psi, \gamma, \sigma \Rightarrow \mathrm{z}}[/L]$$

$$\rhd_{1D} \quad \frac{\dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \quad \mathbf{Q}(\pi, \mathrm{v}, \gamma \Rightarrow \mathrm{y})}{\pi, \mathrm{v/w}, \psi, \gamma \Rightarrow \mathrm{y}}[/L] \qquad \mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \sigma \Rightarrow \mathrm{z})}{\phi^n, \mathrm{x/y}, \pi, \mathrm{v/w}, \psi, \gamma, \sigma \Rightarrow \mathrm{z}}[/L]$$

(b)

$$\frac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{Q}(\pi, \mathrm{v}, \gamma \Rightarrow \mathrm{y}) \quad \mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \sigma \Rightarrow \mathrm{z})}{\phi^n, \mathrm{x/y}, \pi, \mathrm{v}, \gamma, \sigma \Rightarrow \mathrm{z}}[/L]}{\phi^n, \mathrm{x/y}, \pi, \psi, \mathrm{v}\backslash\mathrm{w}, \gamma, \sigma \Rightarrow \mathrm{z}}[\backslash L]$$

$$\rhd_{1D} \quad \frac{\dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \quad \mathbf{Q}(\pi, \mathrm{v}, \gamma \Rightarrow \mathrm{y})}{\pi, \psi, \mathrm{v}\backslash\mathrm{w}, \gamma \Rightarrow \mathrm{y}}[\backslash L] \qquad \mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \sigma \Rightarrow \mathrm{z})}{\phi^n, \mathrm{x/y}, \pi, \psi, \mathrm{v}\backslash\mathrm{w}, \gamma, \sigma \Rightarrow \mathrm{z}}[/L]$$

(c)

$$\frac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{Q}(\pi, \mathrm{v}, \gamma \Rightarrow \mathrm{y}) \quad \mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \sigma \Rightarrow \mathrm{z})}{\phi^n, \pi, \mathrm{v}, \gamma, \mathrm{x}\backslash\mathrm{y}, \sigma \Rightarrow \mathrm{z}}[\backslash L]}{\phi^n, \pi, \mathrm{v/w}, \psi, \gamma, \mathrm{x}\backslash\mathrm{y}, \sigma \Rightarrow \mathrm{z}}[/L]$$

$$\rhd_{1D} \quad \frac{\dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \quad \mathbf{Q}(\pi, \mathrm{v}, \gamma \Rightarrow \mathrm{y})}{\pi, \mathrm{v/w}, \psi, \gamma \Rightarrow \mathrm{y}}[/L] \qquad \mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \sigma \Rightarrow \mathrm{z})}{\phi^n, \pi, \mathrm{v/w}, \psi, \gamma, \mathrm{x}\backslash\mathrm{y}, \sigma \Rightarrow \mathrm{z}}[\backslash L]$$

(d)

$$\frac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{Q}(\pi, \mathrm{v}, \gamma \Rightarrow \mathrm{y}) \quad \mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \sigma \Rightarrow \mathrm{z})}{\phi^n, \pi, \mathrm{v}, \gamma, \mathrm{x}\backslash\mathrm{y}, \sigma \Rightarrow \mathrm{z}}[\backslash L]}{\phi^n, \pi, \psi, \mathrm{v}\backslash\mathrm{w}, \gamma, \mathrm{x}\backslash\mathrm{y}, \sigma \Rightarrow \mathrm{z}}[\backslash L]$$

$$\rhd_{1D} \quad \frac{\dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \quad \mathbf{Q}(\pi, \mathrm{v}, \gamma \Rightarrow \mathrm{y})}{\pi, \psi, \mathrm{v}\backslash\mathrm{w}, \gamma \Rightarrow \mathrm{y}}[\backslash L] \qquad \mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \sigma \Rightarrow \mathrm{z})}{\phi^n, \pi, \psi, \mathrm{v}\backslash\mathrm{w}, \gamma, \mathrm{x}\backslash\mathrm{y}, \sigma \Rightarrow \mathrm{z}}[\backslash L]$$

Normalization for the reductive normal form system

In this subsection, we demonstrate that *normalization* holds for the reductive NF system, i.e. that every proof has an INF. This property follows provided that the (stronger) property *strong normalization* holds:

(6.36) Strong Normalization: Every reduction is finite.

Given strong normalization, for every proof, a finite sequence (possibly zero) of contractions must lead to an irreducible proof. Hence, every proof has an INF.

To prove that every reduction is finite, it is sufficient to give a metric that assigns to each proof a finite non-negative integer score, and under which it can be shown that every contraction reduces the score of a proof by a positive integer amount. The scoring system given in (6.38) satisfies these requirements. This makes use of a notion *atomic type count*, defined in (6.37).

(6.37) Atomic Type Count (atc):
a. atc(x) = 1 *if* x is an atomic type
b. atc(x/y) = atc(x\y) = atc(x) + atc(y)

(6.38) The score for any proof **P** (written sc(**P**)) is as follows:

a. *if* **P** is an axiom leaf instantiated with type x
then sc(**P**) = (3 atc(x))!

b. *if* **P** has a right inference at its root, with premise subproof **Q**
then sc(**P**) = sc(**Q**) + 1

c. *if* the root inference of **P** is a head left inference, with major subproof **Q** and minor subproof **R**
then sc(**P**) = (sc(**R**) + 1)sc(**Q**) + 1

d. *if* the root inference of **P** is a non-head left inference, with major subproof **Q** and minor subproof **R**
then sc(**P**) = (sc(**R**) + 1)sc(**Q**)

Note that this scoring system satisfies the requirement that every proof is assigned a non-negative integer score (i.e. an integer score i, $i \geq 0$). The minimum score for an axiom leaf is 3! (i.e. when the inference is instantiated with an atomic type). For proofs whose root is a non-axiom inference, the score assigned is a function of the score(s) of the immediate subproof(s), giving a value greater than that of the immediate subproof(s). Hence, the score for any proof must be greater than zero. Hence, also, a contraction results in a proper decrement in the score assigned to a proof just in case the score of the contractum subproof is less than that of the redex subproof. For each group of contraction rules, we shall consider a single example, and show that this scoring system does satisfy the requirement that every contraction results in a proper decrement in score.

We repeat here a Group A contraction rule:

$$\mathrm{x/y \Rightarrow x/y} \quad \triangleright_{1A} \quad \cfrac{\cfrac{\mathrm{y \Rightarrow y} \quad \mathrm{x \Rightarrow x}}{\mathrm{x/y, y \Rightarrow x}}[/L]}{\mathrm{x/y \Rightarrow x/y}}[/R]$$

Let atc(x) = x and atc(y) = y. Then, the score assigned to the redex is $(3(x+y))!$ and that assigned to the contractum $(3x)!((3y)!+1)+1$ (note that the left inference in the contractum is with respect to the head). The minimal value of x and y is 1. Hence, the contractum score is lower than the redex score provided that for all values of x and y, such that $x, y \geq 1$, it holds that:

$$(3(x+y))! > (3x)!((3y)!+1)+1$$

This is indeed the case (proof by induction).

The following repeats a Group B contraction rule:

$$\dfrac{\mathbf{P}(\pi \Rightarrow \mathrm{y}) \qquad \dfrac{\mathbf{Q}(\phi,\ \mathrm{x},\ \psi,\ \mathrm{w} \Rightarrow \mathrm{z})}{\phi,\ \mathrm{x},\ \psi \Rightarrow \mathrm{z/w}}[/R]}{\phi,\ \mathrm{x/y},\ \pi,\ \psi \Rightarrow \mathrm{z/w}}[/L] \quad \triangleright_{1B} \quad \dfrac{\dfrac{\mathbf{P}(\pi \Rightarrow \mathrm{y}) \qquad \mathbf{Q}(\phi,\ \mathrm{x},\ \psi,\ \mathrm{w} \Rightarrow \mathrm{z})}{\phi,\ \mathrm{x/y},\ \pi,\ \psi,\ \mathrm{w} \Rightarrow \mathrm{z}}[/L]}{\phi,\ \mathrm{x/y},\ \pi,\ \psi \Rightarrow \mathrm{z/w}}[/R]$$

The left inference in these proofs might be with respect to the head type or not, and so we must consider both cases. In what follows, we use p, q and r to stand for the scores assigned to subproofs labelled as **P**, **Q** and **R**, respectively. If the left inference is not with respect to the head, then the scores for the redex and contractum are $(q+1)(p+1)$ (equivalent to $(qp+q+p+1)$) and $(q(p+1)+1)$ (equivalent to $(qp+q+1)$) respectively — a decrement of p. If the left inference is with respect to the head, then the scores for the redex and contractum are $((q+1)(p+1)+1)$ (equivalent to $qp+q+p+2$) and $(q(p+1)+1+1)$ (equivalent to $qp+q+2$), respectively—again a decrement of p.

We next consider a Group C contraction rule:

$$\dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \psi,\ \mathrm{v},\ \sigma \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x/y},\ \pi,\ \psi,\ \mathrm{v},\ \sigma \Rightarrow \mathrm{z}}[/L]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \psi,\ \mathrm{v/w},\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]$$

$$\triangleright_{1C} \quad \dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}) \qquad \dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \psi,\ \mathrm{v},\ \sigma \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x},\ \psi,\ \mathrm{v/w},\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \psi,\ \mathrm{v/w},\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]$$

The redex score is $(p(q+1)+1)(r+1)$ (equivalent to $pqr+pq+pr+p+r+1$), and that for the contractum $p(q+1)(r+1)+1$ (equivalent to $pqr+pq+pr+p+1$), a decrement of r.

Finally, a Group D contraction rule:

$$\dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{Q}(\pi,\ \mathrm{v},\ \gamma \Rightarrow \mathrm{y}) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \sigma \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v},\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v/w},\ \psi,\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]$$

$$\triangleright_{1D} \quad \dfrac{\dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \qquad \mathbf{Q}(\pi,\ \mathrm{v},\ \gamma \Rightarrow \mathrm{y})}{\pi,\ \mathrm{v/w},\ \psi,\ \gamma \Rightarrow \mathrm{y}}[/L] \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \sigma \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v/w},\ \psi,\ \gamma,\ \sigma \Rightarrow \mathrm{z}}[/L]$$

The score for the redex is $(p(q+1)+1)(r+1)$ (equivalent to $pqr+pq+pr+p+r+1$). In the contractum, the left inference combining the subproofs **R** and **Q** might be a head left inference or not, so we must consider both possibilities. In the case where it is a head left inference, the score for the contractum is $p(q(r+1)+1+1)+1$ (equivalent to $pqr+pq+2p+1$), a decrement equal to $pr-p+r$, which is always a proper decrement (since $p, r > 1$). In the case where the left inference is non-head, the score for the contractum is $p(r(q+1)+1)+1$

(equivalent to $pqr + pq + p + 1$), giving a decrement of $pr + r$.

Preservation of meaning under reduction

As we noted above, for our NF system to be of use, it must be the case that every proof has a NF proof that is semantically equivalent to it. This is the case provided that (i) every proof has a NF, and (ii) a NF of any proof is always equivalent to it. Point (i) was proven in the previous subsection. We next show that point (ii) is also true.

The equivalence of proofs and their NFs follows if meaning is preserved under reduction, a property which itself follows just in case meaning is preserved under contraction. We shall consider the groups of contraction rules in turn, to verify that contraction does not change meaning.

The following Group A contraction rule is augmented with proof terms.

$$\mathrm{x/y}{:}f \Rightarrow \mathrm{x/y}{:}f \quad \rhd_{1A} \quad \dfrac{\dfrac{\mathrm{y}{:}g \Rightarrow \mathrm{y}{:}g \qquad \mathrm{x}{:}fg \Rightarrow \mathrm{x}{:}fg}{\mathrm{x/y}{:}f,\ \mathrm{y}{:}g \Rightarrow \mathrm{x}{:}fg}\,[/\mathrm{L}]}{\mathrm{x/y}{:}f \Rightarrow \mathrm{x/y}{:}\lambda g.fg}\,[/\mathrm{R}]$$

The redex and contractum proofs assign the proof terms f and $\lambda g.fg$, respectively, which are equivalent under η-reduction.

Consider the following Group B contraction rule similarly augmented with proof terms:

$$\dfrac{\mathbf{P}(\pi \Rightarrow \mathrm{y}{:}g) \qquad \dfrac{\mathbf{Q}(\phi,\ \mathrm{x}{:}fg,\ \psi,\ \mathrm{w}{:}i \Rightarrow \mathrm{z}{:}h)}{\phi,\ \mathrm{x}{:}fg,\ \psi \Rightarrow \mathrm{z/w}{:}\lambda i.h}\,[/\mathrm{R}]}{\phi,\ \mathrm{x/y}{:}f,\ \pi,\ \psi \Rightarrow \mathrm{z/w}{:}\lambda i.h}\,[/\mathrm{L}] \quad \rhd_{1B} \quad \dfrac{\dfrac{\mathbf{P}(\pi \Rightarrow \mathrm{y}{:}g) \qquad \mathbf{Q}(\phi,\ \mathrm{x}{:}fg,\ \psi,\ \mathrm{w}{:}i \Rightarrow \mathrm{z}{:}h)}{\phi,\ \mathrm{x/y}{:}f,\ \pi,\ \psi,\ \mathrm{w}{:}i \Rightarrow \mathrm{z}{:}h}\,[/\mathrm{L}]}{\phi,\ \mathrm{x/y}{:}f,\ \pi,\ \psi \Rightarrow \mathrm{z/w}{:}\lambda i.h}\,[/\mathrm{R}]$$

Observe that the same lambda expressions label the corresponding types in the root sequents of the redex and contractum subproofs, which consequently assign the same meaning. Precisely the same observation can be made regarding Group C and Group D contraction rules, of which instances labelled with proof terms are as follows:

$$\dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}{:}h) \qquad \dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}{:}i) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x}{:}fi,\ \psi,\ \mathrm{v}{:}gh,\ \sigma \Rightarrow \mathrm{z}{:}j)}{\phi^n,\ \mathrm{x/y}{:}f,\ \pi,\ \psi,\ \mathrm{v}{:}gh,\ \sigma \Rightarrow \mathrm{z}{:}j}\,[/\mathrm{L}]}{\phi^n,\ \mathrm{x/y}{:}f,\ \pi,\ \psi,\ \mathrm{v/w}{:}g,\ \gamma,\ \sigma \Rightarrow \mathrm{z}{:}j}\,[/\mathrm{L}]$$

$$\rhd_{1C} \quad \dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}{:}i) \qquad \dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}{:}h) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x}{:}fi,\ \psi,\ \mathrm{v}{:}gh,\ \sigma \Rightarrow \mathrm{z}{:}j)}{\phi^n,\ \mathrm{x}{:}fi,\ \psi,\ \mathrm{v/w}{:}g,\ \gamma,\ \sigma \Rightarrow \mathrm{z}{:}j}\,[/\mathrm{L}]}{\phi^n,\ \mathrm{x/y}{:}f,\ \pi,\ \psi,\ \mathrm{v/w}{:}g,\ \gamma,\ \sigma \Rightarrow \mathrm{z}{:}j}\,[/\mathrm{L}]$$

$$\dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}{:}h) \qquad \dfrac{\mathbf{Q}(\pi,\ \mathrm{v}{:}gh,\ \gamma \Rightarrow \mathrm{y}{:}i) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x}{:}fi,\ \sigma \Rightarrow \mathrm{z}{:}j)}{\phi^n,\ \mathrm{x/y}{:}f,\ \pi,\ \mathrm{v}{:}gh,\ \gamma,\ \sigma \Rightarrow \mathrm{z}{:}j}[/L]}{\phi^n,\ \mathrm{x/y}{:}f,\ \pi,\ \mathrm{v/w}{:}g,\ \psi,\ \gamma,\ \sigma \Rightarrow \mathrm{z}{:}j}[/L]$$

$$\triangleright_{1D} \quad \dfrac{\dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}{:}h) \qquad \mathbf{Q}(\pi,\ \mathrm{v}{:}gh,\ \gamma \Rightarrow \mathrm{y}{:}i)}{\pi,\ \mathrm{v/w}{:}g,\ \psi,\ \gamma \Rightarrow \mathrm{y}{:}i}[/L] \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x}{:}fi,\ \sigma \Rightarrow \mathrm{z}{:}j)}{\phi^n,\ \mathrm{x/y}{:}f,\ \pi,\ \mathrm{v/w}{:}g,\ \psi,\ \gamma,\ \sigma \Rightarrow \mathrm{z}{:}j}[/L]$$

Thus, meaning is preserved under contraction, and so also under reduction. Hence, a NF of any proof **P** must be equivalent to **P**.

The uniqueness of normal forms

We next consider the uniqueness of NFs. We have seen that every proof has *at least one* NF (given strong normalization). We can show that every proof has a *unique* NF by first proving the following lemma:

(6.39) Lemma A:

$$\textit{if } \mathbf{P} \triangleright_1 \mathbf{A}, \mathbf{P} \triangleright_1 \mathbf{B}, \textit{ then } (\exists \mathbf{T})\ \mathbf{A} \triangleright \mathbf{T}, \mathbf{B} \triangleright \mathbf{T}.$$

We delay the proof of this lemma for a moment. From Lemma A, we can prove the following lemma, by induction on the length of reduction from P to A:

(6.40) Lemma B:

$$\textit{if } \mathbf{P} \triangleright \mathbf{A}, \mathbf{P} \triangleright_1 \mathbf{B}, \textit{ then } (\exists \mathbf{T})\ \mathbf{A} \triangleright \mathbf{T}, \mathbf{B} \triangleright \mathbf{T}.$$

From Lemma B, we can prove that the following property — the Church-Rosser property — holds, by induction on the length of reduction from P to B:

(6.41) Church-Rosser property:

$$\textit{if } \mathbf{P} \triangleright \mathbf{A}, \mathbf{P} \triangleright \mathbf{B}, \textit{ then } (\exists \mathbf{T})\ \mathbf{A} \triangleright \mathbf{T}, \mathbf{B} \triangleright \mathbf{T}.$$

Given that the Church-Rosser property holds, we can argue that any proof must have a *unique* NF as follows:

> Let **A** and **B** be any two NFs of some proof **P**. From the Church-Rosser property, it follows that there must be some proof **T** such that $\mathbf{A} \triangleright \mathbf{T}$, $\mathbf{B} \triangleright \mathbf{T}$. Since **A** and **B** are NFs, they are, by definition, irreducible. Hence, $\mathbf{A} \equiv \mathbf{T}$ and $\mathbf{B} \equiv \mathbf{T}$, and so $\mathbf{A} \equiv \mathbf{B}$.

We next provide a proof for Lemma A. For this, we consider all of the cases that arise for any two occurrences of redexes r_1 and r_2 within a proof **P**, and show that in all of these cases, there is always some common proof that can be reached by contraction sequences which begin with contraction of either redex.

CASE 1. r_1 and r_2 are the same redex occurrence. The lemma obviously holds.

CASE 2. r_1 and r_2 are non-overlapping within **P**, in which case **P** contains a subproof of the form: $\dfrac{\mathbf{R}\quad\mathbf{Q}}{\mathrm{s}}$ where one of the redexes occurs in **R** and the other in **Q**, such that the contraction of the redexes changes **R** to **R′** and **Q** to **Q′**. Clearly, the two contractions are independent such that contracting r_1 and then r_2 gives the same result as contracting r_2 and then r_1, namely: $\dfrac{\mathbf{R}'\quad\mathbf{Q}'}{\mathrm{s}}$

CASE 3. One redex occurs in a subproof of the other redex such that the contraction of either does not change the status of the other as a redex. For example, imagine that r_1 is a Group B redex of the form: $\dfrac{\mathbf{R}\quad\dfrac{\mathbf{Q}}{\mathrm{s1}}}{\mathrm{s2}}$ where r_2 occurs within **Q** such that its contraction would yield **Q′**. If we substitute **Q′** for **Q** in r_1 to give r_1', then r_1' is also a Group B redex. Again, in this case, the contraction of r_1 and then r_2 will yield the same result as first contracting r_2 and then r_1. The same situation may also arise where one redex occurs as a subproof of the other where the latter is a Group C or D redex. Note in this context, that the contraction rules are such that contraction may never alter the headedness of a proof, i.e. if $\mathbf{P}_m \rhd_1 \mathbf{Q}_n$, it must be the case that $m = n$. Then, in the case where one redex, say r_1, is a Group C or D redex of the form: $\dfrac{\mathbf{R}\quad\dfrac{\mathbf{Q}\quad\mathbf{P}}{\mathrm{s1}}}{\mathrm{s2}}$

and where r_2 occurs within the subproof **P**, such that the contraction of r_2 changes **P** to **P′** and r_1 to r_1', it may never be the case that r_1' is not a redex as a consequence of **P′** being differently headed from **P**.

CASE 4. Finally, the two redexes may 'overlap' such that contraction of one redex effectively 'destroys' the other redex, or, to be more precise, yields a result that is not an instance of the relevant redex type. Given the form of the various redexes, this situation can only arise where the larger redex is a Group C or D redex of the form: $\dfrac{\mathbf{R}\quad\dfrac{\mathbf{Q}\quad\dfrac{\mathbf{P}}{\mathrm{s1}}}{\mathrm{s2}}}{\mathrm{s3}}$ and where a Group B redex appears as the subproof having root sequent s2.

The following shows a schematic instance of this situation where the larger redex is a Group C redex:

$$\mathbf{T} \equiv \dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}) \qquad \dfrac{\mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \psi, \mathrm{v}, \sigma, \mathrm{u} \Rightarrow \mathrm{z})}{\phi^n, \mathrm{x}, \psi, \mathrm{v}, \sigma \Rightarrow \mathrm{z/u}}[/R]}{\phi^n, \mathrm{x/y}, \pi, \psi, \mathrm{v}, \sigma \Rightarrow \mathrm{z/u}}[/L]}{\phi^n, \mathrm{x/y}, \pi, \psi, \mathrm{v/w}, \gamma, \sigma \Rightarrow \mathrm{z/u}}[/L]$$

For proofs of this form, the following sequences of contractions are always possible:

$\mathbf{T} \rhd_{1B} \mathbf{T2} \rhd_{1B} \mathbf{T3} \rhd_{1C} \mathbf{T4}$
$\mathbf{T} \rhd_{1C} \mathbf{T2'} \rhd_{1B} \mathbf{T3'} \rhd_{1B} \mathbf{T4}$

where **T2**, **T3**, **T3′** and **T4** are shown below. Note particularly that contracting the Group B redex in **T** yields a result **T2**, which is *not* a Group C redex.

$$\mathbf{T2} \equiv \quad \dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}) \qquad \dfrac{\dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}) \quad \mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \psi, \mathrm{v}, \sigma, \mathrm{u} \Rightarrow \mathrm{z})}{\phi^n, \mathrm{x/y}, \pi, \psi, \mathrm{v}, \sigma, \mathrm{u} \Rightarrow \mathrm{z}}\text{[/L]}}{\phi^n, \mathrm{x/y}, \pi, \psi, \mathrm{v}, \sigma \Rightarrow \mathrm{z/u}}\text{[/R]}}{\phi^n, \mathrm{x/y}, \pi, \psi, \mathrm{v/w}, \gamma, \sigma \Rightarrow \mathrm{z/u}}\text{[/L]}$$

$$\mathbf{T3} \equiv \quad \dfrac{\dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}) \quad \mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \psi, \mathrm{v}, \sigma, \mathrm{u} \Rightarrow \mathrm{z})}{\phi^n, \mathrm{x/y}, \pi, \psi, \mathrm{v}, \sigma, \mathrm{u} \Rightarrow \mathrm{z}}\text{[/L]}}{\phi^n, \mathrm{x/y}, \pi, \psi, \mathrm{v/w}, \gamma, \sigma, \mathrm{u} \Rightarrow \mathrm{z}}\text{[/L]}}{\phi^n, \mathrm{x/y}, \pi, \psi, \mathrm{v/w}, \gamma, \sigma \Rightarrow \mathrm{z/u}}\text{[/R]}$$

$$\mathbf{T4} \equiv \quad \dfrac{\dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}) \qquad \dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}) \quad \mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \psi, \mathrm{v}, \sigma, \mathrm{u} \Rightarrow \mathrm{z})}{\phi^n, \mathrm{x}, \psi, \mathrm{v/w}, \gamma, \sigma, \mathrm{u} \Rightarrow \mathrm{z}}\text{[/L]}}{\phi^n, \mathrm{x/y}, \pi, \psi, \mathrm{v/w}, \gamma, \sigma, \mathrm{u} \Rightarrow \mathrm{z}}\text{[/L]}}{\phi^n, \mathrm{x/y}, \pi, \psi, \mathrm{v/w}, \gamma, \sigma \Rightarrow \mathrm{z/u}}\text{[/R]}$$

$$\mathbf{T2}' \equiv \quad \dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}) \qquad \dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \psi, \mathrm{v}, \sigma, \mathrm{u} \Rightarrow \mathrm{z})}{\phi^n, \mathrm{x}, \psi, \mathrm{v}, \sigma \Rightarrow \mathrm{z/u}}\text{[/R]}}{\phi^n, \mathrm{x}, \psi, \mathrm{v/w}, \gamma, \sigma \Rightarrow \mathrm{z/u}}\text{[/L]}}{\phi^n, \mathrm{x/y}, \pi, \psi, \mathrm{v/w}, \gamma, \sigma \Rightarrow \mathrm{z/u}}\text{[/L]}$$

$$\mathbf{T3}' \equiv \quad \dfrac{\mathbf{Q}(\pi \Rightarrow \mathrm{y}) \qquad \dfrac{\dfrac{\mathbf{R}(\gamma \Rightarrow \mathrm{w}) \quad \mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \psi, \mathrm{v}, \sigma, \mathrm{u} \Rightarrow \mathrm{z})}{\phi^n, \mathrm{x}, \psi, \mathrm{v/w}, \gamma, \sigma, \mathrm{u} \Rightarrow \mathrm{z}}\text{[/L]}}{\phi^n, \mathrm{x}, \psi, \mathrm{v/w}, \gamma, \sigma \Rightarrow \mathrm{z/u}}\text{[/R]}}{\phi^n, \mathrm{x/y}, \pi, \psi, \mathrm{v/w}, \gamma, \sigma \Rightarrow \mathrm{z/u}}\text{[/L]}$$

Similar sequences of contractions are also possible for the other cases where a Group B and C redex overlap (i.e. cases constructed using different directionalities for the left and right inferences), as the reader can easily (if laboriously) verify.

The following shows a case where Group D and Group B redexes overlap:

$$\mathbf{T} \equiv \quad \dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{Q}(\pi, \mathrm{v}, \gamma \Rightarrow \mathrm{y}) \qquad \dfrac{\mathbf{P}_{n+1}(\phi^n, \mathrm{x}, \sigma, \mathrm{u} \Rightarrow \mathrm{z})}{\phi^n, \mathrm{x}, \sigma \Rightarrow \mathrm{z/u}}\text{[/R]}}{\phi^n, \mathrm{x/y}, \pi, \mathrm{v}, \gamma, \sigma \Rightarrow \mathrm{z/u}}\text{[/L]}}{\phi^n, \mathrm{x/y}, \pi, \mathrm{v/w}, \psi, \gamma, \sigma \Rightarrow \mathrm{z/u}}\text{[/L]}$$

For proofs of the form **T**, we find that the following sequences of contractions are always possible:

$$\mathbf{T} \ \triangleright_{1B} \mathbf{T2} \ \triangleright_{1B} \mathbf{T3} \ \triangleright_{1D} \mathbf{T4}$$
$$\mathbf{T} \ \triangleright_{1D} \mathbf{T2}' \ \triangleright_{1B} \mathbf{T4}$$

where **T2**, **T2′**, **T3** and **T4** are shown below. Note that the contraction of the Group B redex in **T** yields a result **T2**, which is not a Group D redex.

$$\mathbf{T2} \equiv \dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \qquad \dfrac{\dfrac{\mathbf{Q}(\pi,\ \mathrm{v},\ \gamma \Rightarrow \mathrm{y}) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \sigma,\ \mathrm{u} \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v},\ \gamma,\ \sigma,\ \mathrm{u} \Rightarrow \mathrm{z}}[/L]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v},\ \gamma,\ \sigma \Rightarrow \mathrm{z/u}}[/R]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v/w},\ \psi,\ \gamma,\ \sigma \Rightarrow \mathrm{z/u}}[/L]$$

$$\mathbf{T3} \equiv \dfrac{\dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \qquad \dfrac{\mathbf{Q}(\pi,\ \mathrm{v},\ \gamma \Rightarrow \mathrm{y}) \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \sigma,\ \mathrm{u} \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v},\ \gamma,\ \sigma,\ \mathrm{u} \Rightarrow \mathrm{z}}[/L]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v/w},\ \psi,\ \gamma,\ \sigma,\ \mathrm{u} \Rightarrow \mathrm{z}}[/L]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v/w},\ \psi,\ \gamma,\ \sigma \Rightarrow \mathrm{z/u}}[/R]$$

$$\mathbf{T4} \equiv \dfrac{\dfrac{\dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \qquad \mathbf{Q}(\pi,\ \mathrm{v},\ \gamma \Rightarrow \mathrm{y})}{\pi,\ \mathrm{v/w},\ \psi,\ \gamma \Rightarrow \mathrm{y}}[/L] \qquad \mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \sigma,\ \mathrm{u} \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v/w},\ \psi,\ \gamma,\ \sigma,\ \mathrm{u} \Rightarrow \mathrm{z}}[/L]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v/w},\ \psi,\ \gamma,\ \sigma \Rightarrow \mathrm{z/u}}[/R]$$

$$\mathbf{T2'} \equiv \dfrac{\dfrac{\mathbf{R}(\psi \Rightarrow \mathrm{w}) \qquad \mathbf{Q}(\pi,\ \mathrm{v},\ \gamma \Rightarrow \mathrm{y})}{\pi,\ \mathrm{v/w},\ \psi,\ \gamma \Rightarrow \mathrm{y}}[/L] \qquad \dfrac{\mathbf{P}_{n+1}(\phi^n,\ \mathrm{x},\ \sigma,\ \mathrm{u} \Rightarrow \mathrm{z})}{\phi^n,\ \mathrm{x},\ \sigma \Rightarrow \mathrm{z/u}}[/R]}{\phi^n,\ \mathrm{x/y},\ \pi,\ \mathrm{v/w},\ \psi,\ \gamma,\ \sigma \Rightarrow \mathrm{z/u}}[/L]$$

Again similar sequences of contractions are also possible for the other cases where a Group B and D redex overlap (i.e. cases constructed using different directionalities for the left and right inferences).

We have seen that in all cases where a proof contains two redexes, there is always some common proof which can be reached by contraction sequences that begin with contraction of either redex. Hence, Lemma A holds, and so in turn does the uniqueness of NFs.

6.8.5 Equivalence of the two normal form systems

In this section, we demonstrate that the two NF systems are equivalent. We write CNF(**P**) and INF(**P**) to indicate that **P** is in NF under the respective systems. The two NF systems can be shown to be equivalent in the sense that they designate precisely the same sets of proofs to be in NF, i.e:

(6.42) Lemma 1: for all **P**, CNF(**P**) $\leftrightarrow$ INF(**P**)

To prove this, it is sufficient to show that Lemmas 2 and 3 hold.

(6.43) Lemma 2: for all **P**, $\neg$INF(**P**) $\rightarrow$ $\neg$CNF(**P**)

PROOF: The correctness of this lemma is obvious. Given the definition of CNF in (6.27), no Group A redex is in CNF since a CNF may not contain any axiom leaf that is instantiated with a non-atomic type. No Group B redex is in CNF since a CNF may not contain a left inference whose major subproof begins with a right inference. No Group C or D redex is in CNF since a CNF may not contain any left inference whose active type is not the head of the ensuing subproof. Hence, any reducible proof is not a CNF.

(6.44) Lemma 3: for all $\mathbf{P}$, $\neg\mathrm{CNF}(\mathbf{P}) \rightarrow \neg\mathrm{INF}(\mathbf{P})$

PROOF: by induction on the structure of $\mathbf{P}$.

CASE 1. $\mathbf{P} \equiv \mathrm{x} \Rightarrow \mathrm{x}$. Since $\neg\mathrm{CNF}(\mathbf{P})$, x must be non-atomic, but then P is a redex and so $\neg\mathrm{INF}(\mathbf{P})$.

CASE 2. $\mathbf{P} \equiv \dfrac{\mathbf{Q}}{\mathrm{S}}$. For $\neg\mathrm{CNF}(\mathbf{P})$ to hold, it must be the case that $\neg\mathrm{CNF}(\mathbf{Q})$. By induction hypothesis, it follows that $\neg\mathrm{INF}(\mathbf{Q})$, and hence that $\neg\mathrm{INF}(\mathbf{P})$.

CASE 3. $\mathbf{P} \equiv \dfrac{\mathbf{Q} \quad \mathbf{R}}{\mathrm{S}}$ where either $\neg\mathrm{CNF}(\mathbf{Q})$ or $\neg\mathrm{CNF}(\mathbf{R})$.

By induction hypothesis, either $\neg\mathrm{INF}(\mathbf{Q})$ or $\neg\mathrm{INF}(\mathbf{R})$. Hence $\neg\mathrm{INF}(\mathbf{P})$.

CASE 4. $\mathbf{P} \equiv \dfrac{\mathbf{Q} \quad \mathbf{R}}{\mathrm{S}}$ where $\mathrm{CNF}(\mathbf{Q})$ and $\mathrm{CNF}(\mathbf{R})$.

Since $\neg\mathrm{CNF}(\mathbf{P})$, it must be the case that either:

a. $\neg(\mathbf{R} \in \tau_2)$. Then, root inference of $\mathbf{R}$ must be a right inference, and so $\mathbf{P}$ is a Group B redex, or

b. $\mathbf{R} \in \tau_2$, but root inference of $\mathbf{P}$ is a non-head left inference. The minor premise of a non-head left inference cannot be an axiom instance, and so, since $\mathbf{R} \in \tau_2$, the root inference of $\mathbf{R}$ must be a head left inference. Hence, P is a Group C or D redex

6.8.6 The uniqueness of readings

We have demonstrated the equivalence of the two NF systems. Consequently, various of the properties demonstrated for the reductive NF system carry across to the constructive NF system also. In particular, it follows that every proof $\mathbf{P}$ has a unique CNF and that this CNF is equivalent to $\mathbf{P}$. However, we have argued that a NF system is *optimal* for parsing purposes just in case there is only a single NF proof assigning *each distinct reading*, which does not follow automatically given the uniqueness of NFs. It may be the case that proofs which are semantically (extensionally) equivalent have distinct NFs. Uniqueness *with respect to readings* holds only if the reduction system in some sense 'captures' full extensional equivalence for proof readings.

We have seen that the meaning assigned by any cut-free proof is always a lambda expression in β-NF, and that any two terms in β-NF are extensionally equivalent just in case they yield some common term under η-reduction. We can argue on this basis that the system does exhibit the property of uniqueness with respect to readings.

Let us consider two distinct CNF proofs **P** and **Q** of some sequent S. Assume that they differ in some regard along their main branches. The bottom part of each main branch will consist of a sequence of zero or more right inferences. The length and nature of this sequence of inferences is fully determined by the type of the succedent in the root sequent S, and so **P** and **Q** may not differ in this regard. Thus, the meanings assigned by **P** and **Q** are lambda expressions of the following forms (in which the lambda abstractions $\lambda v_1..v_n.$ correspond to the initial sequence of right inferences):

$$\mathbf{P} : \lambda v_1..v_n.(fU_1..U_m)$$

$$\mathbf{Q} : \lambda v_1..v_n.(gU_1'..U_k')$$

The remainder of each main branch consists of some sequence of head left inferences, terminating in an axiom instance. Consequently, the two CNF proofs may differ along their main branch only in two regards: (i) a different antecedent type is selected as active type for the first left inference (this choice determining the functor for the remaining left inferences on the main branch), (ii) the same active type is chosen for the first left inference, but at some point a different subsequence of antecedents is chosen to 'construct' the argument of the functor. In either case, the semantics assigned to the two proofs must be non-equivalent. In case (i), f and g are distinct variables. In case (ii), at least some of the U_i and U_i' comprise distinct sets of variables (i.e. the semantic expressions of the selected antecedents). It is evident that in either of these eventualities, no sequence of η-contractions can yield a common term from the two resulting proof terms and so the two proofs must assign distinct readings. If the two NF proofs are similar along their main branches, they must differ in the minor subproof of some left inference. The same arguments apply in turn that if these subproofs differ in form, then they must be non-equivalent. Hence, distinct CNF proofs always differ semantically, and so the uniqueness property holds.

The property of uniqueness with respect to readings gives rise to a second proof of the weaker property of the uniqueness of NFs. Since a NF of any proof must be equivalent to it, and since distinct NFs are non-equivalent, it follows that any proof can have only a single NF. It follows in turn that the reductive NF system exhibits the Church-Rosser property (given strong normalization).

6.8.7 Normal forms and theorem proving

We next consider how the NF system developed above can provide a basis for efficient theorem proving of the Lambek calculus. We have seen that for any reading that can be assigned to a given sequent under any proof, there exists a unique CNF (that is also an INF) proof of that sequent that assigns the same reading. Hence, we can safely limit our theorem prover to producing only the CNF proofs of a sequent.

It would be most convenient if the recursive definition of CNF proof directly gave rise to a set of inference rules that only allows the construction of CNF proofs. Unfortunately, things are not so straightforward since inference rules simply relate premises to conclusions whereas the admissibility of constructing one CNF proof from one or two others using some inference rule under the constructive definition of CNF crucially depends on the form of the proofs so combined, i.e. whether they meet the definition of τ_1 or τ_2 proof, and in the case of combining two proofs under a left inference, whether this left inference is with respect to the head of the major subproof.

However, we can overcome this problem by moving from the standard Gentzen-style sequent system which uses a single notion of *sequent*, to a system in which we distinguish two different types of sequent (Type 1 and Type 2), this move providing a basis for addressing purely in terms of the status of a premise, the form of any proof that may be given for it. We will refer to this alternative system that uses Type 1 and Type 2 sequents as the *parsing calculus* (PC). Recall that the definition of CNF proof distinguishes two subtypes of proof: τ_1 and τ_2, where the former corresponds precisely to the set of CNF proofs, and the latter to the subset of CNF proofs whose main branch includes no right inferences. The distinction of Type 1 and Type 2 sequent in the PC corresponds to the τ_1 and τ_2 proof distinction in the definition of CNF proofs. In particular, any proof for a Type 2 sequent must correspond to a τ_2 proof, and likewise for Type 1 sequents and τ_1 proofs. As we will see, the proofs that are given under the two systems are not identical, but correspond very directly, so that a mapping between corresponding proofs in the two systems is very easily given.

We mark the Type 1/Type 2 sequent distinction by a subscript on the derivability arrow, thus: $\phi \Rightarrow_1 \mathrm{x}$ and $\phi \Rightarrow_2 \mathrm{x}$. For Type 2 sequents, we need to distinguish the antecedent that *must* be the head under any proof of the sequent. We do this by dividing the antecedent sequence into three subparts, separated by +'s, as for example in $\phi{+}\mathrm{x}{+}\pi \Rightarrow_2 \mathrm{z}$, of which the first and last parts are (possibly empty) sequences of types, and the middle part is a single type — the type that is required to be the head of any proof of the sequent.

Using Type 1 and Type 2 sequents, we can give a set of inference rules that only allow construction of proofs of the required form as follows:

(6.45) Axiom: $e{+}\mathrm{x}{:}f{+}e \Rightarrow_2 \mathrm{x}{:}f$ where x is an atomic type (and e denotes the empty sequence)

Right rules:
$$\frac{\Gamma,\ \mathrm{y}{:}i \Rightarrow_1 \mathrm{x}{:}f}{\Gamma \Rightarrow_1 \mathrm{x/y}{:}\lambda i.f}[/R] \qquad \frac{\mathrm{y}{:}i,\ \Gamma \Rightarrow_1 \mathrm{x}{:}f}{\Gamma \Rightarrow_1 \mathrm{x\backslash y}{:}\lambda i.f}[\backslash R]$$

Left rules:
$$\frac{\Delta \Rightarrow_1 \mathrm{y}{:}g \qquad \Gamma{+}\mathrm{x}{:}fg{+}\Lambda \Rightarrow_2 \mathrm{z}{:}h}{\Gamma{+}\mathrm{x/y}{:}f{+}(\Delta,\ \Lambda) \Rightarrow_2 \mathrm{z}{:}h}[/L]$$
$$\frac{\Delta \Rightarrow_1 \mathrm{y}{:}g \qquad \Gamma{+}\mathrm{x}{:}fg{+}\Lambda \Rightarrow_2 \mathrm{z}{:}h}{(\Gamma,\ \Delta){+}\mathrm{x\backslash y}{:}f{+}\Lambda \Rightarrow_2 \mathrm{z}{:}h}[\backslash L]$$

[2↦1] rule:
$$\frac{\Delta{+}\mathrm{x}{:}f{+}\Gamma \Rightarrow_2 \mathrm{y}{:}g}{\Delta,\ \mathrm{x}{:}f,\ \Gamma \Rightarrow_1 \mathrm{y}{:}g}[2{\mapsto}1]$$

Observe that the axiom rule specifies a Type 2 sequent, with the single antecedent type being the designated head of that sequent. This corresponds to clause (a) of the definition of CNF in (6.27). The left inference rules have Type 2 conclusions, with a Type 2 sequent for the major premise and a Type 1 sequent for the minor premise. Note particularly that the active type for the inference is the designated head of the conclusion and its value subtype is the designated head of the major premise sequent. These rules correspond to clauses (b) and (c) of the CNF definition. The right rules have both premise and conclusion as Type 1 sequents, in line with clauses (e) and (f) of the CNF definition. In addition, a further rule is provided, called the [2↦1] rule. This rule corresponds to the clause (d) of the CNF definition, which says that every τ_2 proof is also a τ_1 proof.

It should be evident that the PC will allow only proofs that are (in essence) of a form corresponding to CNF proofs. PC and CNF proofs exist in 1-for-1 correspondence, as we can demonstrate by specifying a mapping between proofs, in both directions. The mapping from PC proofs to CNF proofs is trivial. This involves simply 'deleting' unwanted bits of notation such as +'s and subscripts, as well as [2↦1] rule uses:

(6.46) From PC to CNF (k):

(a) $k(e{+}\mathrm{x}{+}e \Rightarrow_2 \mathrm{x}) \quad = \quad \mathrm{x} \Rightarrow \mathrm{x}$

(b) $$k\left(\frac{\mathbf{P}}{\phi \Rightarrow_1 \mathrm{x}}[R]\right) = \frac{k(\mathbf{P})}{\phi \Rightarrow \mathrm{x}}[R]$$

(c) $$k\left(\frac{\mathbf{P}}{\mathrm{s}}[2{\mapsto}1]\right) = k(\mathbf{P})$$

(d) $$k\left(\frac{\mathbf{Q} \qquad \mathbf{P}}{\phi{+}\mathrm{x}{+}\pi \Rightarrow_2 \mathrm{y}}[L]\right) = \frac{k(\mathbf{Q}) \qquad k(\mathbf{P})}{\phi,\ \mathrm{x},\ \pi \Rightarrow \mathrm{y}}[L]$$

The mapping from CNF to PC proofs is slightly more complicated:

(6.47) From CNF to PC (h):

(h defined in terms of f and g)

(a) $h(\mathbf{P}) \quad = \quad f(g(\mathbf{P}))$

(b) $f(\mathbf{P}(\phi \Rightarrow_1 \mathrm{x})) \quad = \quad \mathbf{P}(\phi \Rightarrow_1 \mathrm{x})$

(c) $f(\mathbf{P}(\phi \Rightarrow_2 \mathrm{x})) \quad = \quad \dfrac{\mathbf{P}(\phi \Rightarrow_2 \mathrm{x})}{\phi \Rightarrow_1 \mathrm{x}}[2{\mapsto}1]$

(d) $g(\mathrm{x} \Rightarrow \mathrm{x}) \quad = \quad e{+}\mathrm{x}{+}e \Rightarrow_2 \mathrm{x}$

(e) $g\left(\dfrac{\mathbf{P}}{\phi \Rightarrow \mathrm{x}}[R]\right) \quad = \quad \dfrac{h(\mathbf{P})}{\phi \Rightarrow_1 \mathrm{x}}[R]$

(f) $g\left(\dfrac{\mathbf{Q} \quad \mathbf{P}(\phi,\, \mathrm{x},\, \psi \Rightarrow \mathrm{z})}{\phi,\, \mathrm{x/y},\, \pi,\, \psi \Rightarrow \mathrm{z}}[/L]\right) \quad = \quad \dfrac{h(\mathbf{Q}) \quad g(\mathbf{P}(\phi,\, \mathrm{x},\, \psi \Rightarrow \mathrm{z}))}{\phi{+}\mathrm{x/y}{+}(\pi,\, \psi) \Rightarrow_2 \mathrm{z}}[/L]$

(g) $g\left(\dfrac{\mathbf{Q} \quad \mathbf{P}(\phi,\, \mathrm{x},\, \psi \Rightarrow \mathrm{z})}{\phi,\, \pi,\, \mathrm{x\backslash y},\, \psi \Rightarrow \mathrm{z}}[\backslash L]\right) \quad = \quad \dfrac{h(\mathbf{Q}) \quad g(\mathbf{P}(\phi,\, \mathrm{x},\, \psi \Rightarrow \mathrm{z}))}{(\phi,\, \pi){+}\mathrm{x\backslash y}{+}\psi \Rightarrow_2 \mathrm{z}}[\backslash L]$

Note that the purpose of f is to insert [2↦1] rule uses as and when required. Since CNF and PC proofs exist in 1-to-1 correspondence, it is clearly safe to construct only the PC proofs of a sequent.

The PC as it stands is sufficient to yield all and only the proofs that we want, i.e. proofs corresponding to the CNF proofs of any sequent. However, theorem proving based on the PC can be made considerably more efficient by adding some straightforwardly justifiable conditions to the [2↦1] rule. This rule does not refer to the structure of any type in its conclusion, and so, in top-down theorem proving, can be applied whenever the goal sequent is Type 1, and most such rule uses will lead to failure.

It turns out that when the [2↦1] rule is correctly applied (i.e. when its use may allow successful proof construction), certain conditions must always obtain. In particular, the succedent of the goal sequent must be an atomic type, and the antecedent selected to be head of the ensuing proof must be (ultimately) a function into that succedent type, i.e. the 'innermost' value subtype of the head antecedent must be of the same type as the succedent (c.f. König's Goal Criterium).

The main branch of a τ_2 proof consists of a sequence of zero or more head left inferences ending in an axiom. The antecedent and succedent of the axiom have the same type, which is required to be atomic. With each head left inference, the succedent type is inherited from major premise to conclusion, and the type of the head in the conclusion is a function into the type in the head of the major premise. Hence, the head of the sequent under a τ_2 proof is a function into the atomic type of the succedent. Since these two conditions always hold when the [2↦1] rule is correctly used, they can safely be added to the statement of the rule, giving (6.48), without altering the set of proofs admitted by the system. These conditions considerably limit the cases where the rule may be applied, and so reduce the search space that must be addressed during top-down theorem proving, and thereby improving its efficiency.

(6.48)[2↦1] rule:

$$\frac{\Delta\text{+x:}f\text{+}\Gamma \Rightarrow_2 \text{y:}g}{\Delta,\ \text{x:}f,\ \Gamma \Rightarrow_1 \text{y:}g}[2\mapsto 1] \quad \text{where y is an atomic type, and x = y or x a function into y}$$

6.9 Conclusion

We have considered three approaches to handling the problem of multiple equivalent proofs as it arises for Lambek calculus theorem proving. These methods are all based around selecting NF exemplars of equivalence classes of proofs, and ensuring that the parsing method produces all and only such NFs. Furthermore, the three approaches are based around very similar notions of NF proofs, which are of note in that the structure of NF proofs can be seen to closely mirror the structure of the proof terms that they assign.

We have noted a number of criticisms of Moorgat's approach. Most significantly, we consider the addition of a compilation phrase, during which the types of the initial goal sequent are partially executed with respect to the inference rules, to be an unnecessary complication of the overall parsing method. An adequate method which does not require a compilation phase would seem preferable to Moortgat's approach.

König's approach has the advantage of not requiring a compilation phase, but, as we have seen, has a number of problems, both in respect of the NF system and the parsing method that is derived from it. Hepple's approach can be seen to develop the essential idea underlying König's method, overcoming these problems. We have demonstrated that our method is safe in the sense that all readings that can be assigned to a sequent will be returned, and optimal in that there is only a single proof which assigns each reading. The 'parsing calculus' we have specified provides for efficient Lambek theorem proving, requiring the use of only standard theorem proving techniques.

Chapter 7

Conclusion

The extended Lambek calculus is a logically based highly lexical grammatical framework which exhibits a close correspondence of syntax and semantics. The framework arises through augmenting the basic Lambek calculus (Lambek, 1958) by the inclusion of additional type forming operators whose logical behaviour allows for the characterization of some aspect of linguistic phenomena. The general approach exhibits a recognition that new type forming operators may *need* to be created to provide the fundamental primitives necessary to deal with hitherto problematic phenomena.

This thesis has been primarily concerned with developing accounts of some important linguistic phenomena within the extended Lambek framework, with the twin goals of developing the grammatical framework and attempting to cast new light on the phenomena addressed. In the course of the thesis, we have introduced a number of new type forming operators which were needed to provide a basis for the accounts developed.

Some of the accounts presented can be seen to incorporate in modified form ideas that, at some level or another, underlie existing accounts of the phenomena addressed. For example, our account of word order clearly has strong precedents in the accounts of Jacobson (1987) and Koster (1988), but also develops the basic idea, not least by providing a completely lexical account, eliminating the need to invoke movement in the syntactic domain. Similarly, we adopt a modified version of a treatment of grammatical relations and obliqueness that has been fruitfully exploited in Montagovian work, but our work sets this treatment in an approach where syntactic combination is handled in terms of a highly general calculus of type change, rather than a system which depends on rule specific stipulations.

In contrast, we would suggest that the polymodal apparatus provides an entirely novel perspective on the treatment of linguistic boundaries, one for which there is no precedent of which the author is aware. Obviously, the approach has antecedents, particularly Morrill's proposal that modality can provide for the treatment of locality constraints (Morrill, 1989). However, the system that results with the move to polymodality is of a character quite unlike any other perspective on linguistic boundaries, allowing as it does multiple notions of boundary and the statement of relations amongst modalities which enable the sensitivity of particular processes to any given class of boundaries to be explicitly encoded.

A second concern of this work has been the development of methods for efficient processing of Lambek grammars. It is an unfortunate characteristic of the range of grammatical frameworks currently available that the suitability of each framework for natural language processing applications commonly stands in inverse relation to its breadth of grammatical coverage. It is a long term goal of the current work to develop accounts within the Lambek framework that are not only of interest in linguistic terms, but also provide for the possibility of natural language processing systems which address more than the most minimal of language fragments.

Of course, a great deal of work remains to be done. This thesis, partly because of its broad scope, does little more than scratch the surface of the phenomena addressed. For the treatment of island constraints, for example, we have simply explored the possibilities for how a range of constraints might be handled. The precise formulation of an account of these constraints would require considerable further work. The same point can be made for the other linguistic accounts presented.

The work on efficient processing of Lambek grammars has so far addressed only the minimal fragment of the calculus, albeit the most important (i.e. the implicational fragment). It remains to be seen whether the normal form approach can successfully accommodate the full range of type forming operators that will be required in a fully developed extended Lambek framework.

Appendix A

Normal form theorem prover

This appendix contains a listing of a program that does theorem proving for the (product free) Lambek calculus under both the standard sequent formulation and the normal form parsing calculus system described in Chapter 6. An illustrative log of a terminal session is also given. Test examples have been selected to illustrate the occurrence of multiple equivalent proofs (and their avoidance by the normal form parsing calculus). For extensive discussion of Lambek theorem proving see Moortgat (1988).

A.1 Parser Listing

```
%  Theorem prover for (product free) Lambek calculus.
%  Has clauses for both ordinary sequent formulation and
%  for (normal form) Parsing calculus

% Syntax

:- op(400,yfx,/).
:- op(400,yfx,\).

% Semantics

:- op(450,yfx,`).           % functional application
:- op(460,xfy,@).           % `lambda' operator

% Structural

:- op(500,xfx,:).           % separates type/semantics pairs  (T:S)
:- op(600,xfx,'=>').        % ordinary derivability relation
:- op(600,xfx,'=>1').       % derivability relation for type 1 sequents
:- op(600,xfx,'=>2').       % derivability relation for type 2 sequents
```

```
% Proof clauses for (product free) Lambek calculus

% /R
prove(Goal,Proof):-
        Goal      =   Gamma0 => X/Y:B@A,
        Premise1  =   Gamma1 => X:A,
        Proof     =   [Goal,'/R   ',Proof1],
        append(Gamma0,[Y:B],Gamma1),
        prove(Premise1,Proof1).

% \R
prove(Goal,Proof):-
        Goal      =   Gamma0 => X\Y:B@A,
        Premise1  =   Gamma1 => X:A,
        Proof     =   [Goal,'\R   ',Proof1],
        append([Y:B],Gamma0,Gamma1),
        prove(Premise1,Proof1).

% /L
prove(Goal,Proof):-
        Goal      =   Gamma0 => Z:A,
        Premise1  =   Gamma1 => Z:A,
        Premise2  =   Gamma2 => Y:B,
        Proof     =   [Goal,'/L   ',Proof1,Proof2],
        appendn([Pi1,[X/Y:C],Gamma2,Pi2],Gamma0),
        appendn([Pi1,[X:(C`B)],Pi2],Gamma1),
        prove(Premise1,Proof1),
        prove(Premise2,Proof2).

% \L
prove(Goal,Proof):-
        Goal      =   Gamma0 => Z:A,
        Premise1  =   Gamma1 => Z:A,
        Premise2  =   Gamma2 => Y:B,
        Proof     =   [Goal,'\L   ',Proof1,Proof2],
        appendn([Pi1,Gamma2,[X\Y:C],Pi2],Gamma0),
        appendn([Pi1,[X:(C`B)],Pi2],Gamma1),
        prove(Premise1,Proof1),
        prove(Premise2,Proof2).

% Axiom
prove(Goal,Proof):-
        Goal      =   [X:A]  => X:A,
        Proof     =   [Goal,'axiom'].
```

```
% Proof clauses for Parsing calculus (normal form system).

% /R
prove(Goal,Proof):-
        Goal      =   Gamma0 '=>1' X/Y:B@A,
        Premise1  =   Gamma1 '=>1' X:A,
        Proof     =   [Goal,'/R   ',Proof1],
        append(Gamma0,[Y:B],Gamma1),
        prove(Premise1,Proof1).

% \R
prove(Goal,Proof):-
        Goal      =   Gamma0 '=>1' X\Y:B@A,
        Premise1  =   Gamma1 '=>1' X:A,
        Proof     =   [Goal,'\R   ',Proof1],
        append([Y:B],Gamma0,Gamma1),
        prove(Premise1,Proof1).

% 2->1
prove(Goal,Proof):-
        Goal      =   Gamma0 '=>1' Y:B,
        Premise1  =   Pi1 + (X:A) + Pi2 '=>2' Y:B,
        Proof     =   [Goal,'2->1 ',Proof1],
        appendn([Pi1,[X:A],Pi2],Gamma0),
        atomic_type(Y),
        result_type(Y,X),
        prove(Premise1,Proof1).

% /L
prove(Goal,Proof):-
        Goal      =   Pi1 + (X/Y:C) + Pi2 '=>2' Z:A,
        Premise1  =   Pi1 + (X:C`B) + Pi3 '=>2' Z:A,
        Premise2  =   Gamma '=>1' Y:B,
        Proof     =   [Goal,'/L   ',Proof1,Proof2],
        append(Gamma,Pi3,Pi2),
        prove(Premise1,Proof1),
        prove(Premise2,Proof2).

% \L
prove(Goal,Proof):-
        Goal      =   Pi1 + (X\Y:C) + Pi2 '=>2' Z:A,
        Premise1  =   Pi3 + (X:C`B) + Pi2 '=>2' Z:A,
        Premise2  =   Gamma '=>1' Y:B,
        Proof     =   [Goal,'\L   ',Proof1,Proof2],
        append(Pi3,Gamma,Pi1),
        prove(Premise1,Proof1),
        prove(Premise2,Proof2).
```

```
% Axiom
prove(Goal,Proof):-
        Goal      =   []+(X:A)+[]  '=>2' X:A,
        Proof     =   [Goal,'axiom'].

%  Test predicate. Accesses and prints numbered goal sequent.
%  Performs failure-driven exhaustive search for proofs under
%  the product free Lambek calculus, printing out each proof
%  found and the proof term for the succedent of goal sequent.
%  Then, does same under the Parsing calculus.

test(N):-
     goal_sequent(N,Sequent),
     nl,
     write('Goal sequent:  '),
     write(Sequent),
     add_sem_vars(Sequent,Gamma => Z),
     nl, nl,
     write('Non-normal form theorem proving:'),
     nl,
     \+(( prove(Gamma => Z, Proof),
          numbervars(Proof,0,_),
          print_proof_and_sem(Proof),
          fail )),
     nl,
     write('Normal form theorem proving:'),
     nl,
     \+(( prove(Gamma '=>1' Z, Proof),
          numbervars(Proof,0,_),
          print_proof_and_sem(Proof),
          fail )).

% Goal sequents for testing.

goal_sequent(1,[x/y\z] => x/y\z).
goal_sequent(2,[x/y,y/z,z] => x).
goal_sequent(3,[x/y,y/z,z/w] => x/w).
goal_sequent(4,[x/x,x,x\x] => x).
goal_sequent(5,[x,y,w\x/z\y] => w/z).
goal_sequent(6,[x/(y/z),y/w,w/z] => x).
goal_sequent(7,[x] => y/(y\x)).

% Associates each type in a sequent with a new semantic variable.

add_sem_vars(Gamma0 => X, Gamma1 => X:_):-
        add_sem_vars_list(Gamma0,Gamma1).

add_sem_vars_list([],[]).
add_sem_vars_list([H|Gamma0],[H:_|Gamma1]):-
        add_sem_vars_list(Gamma0,Gamma1).
```

```
% Test for an atomic type.  Prolog atom check sufficient for this
% implementation.

atomic_type(X):- atom(X).

% Checks if first argument is a result subtype of second argument.

result_type(X,X).
result_type(X,Y/Z):-
     result_type(X,Y).
result_type(X,Y\Z):-
     result_type(X,Y).

% Prints proof with rule names and indentation,
% then prints semantics for goal sequent separately.

print_proof_and_sem(Proof):-
   print_proof(Proof,3),
   Proof = [Goal|_],
   Goal =.. [_,_,_:S],
   nl, nl,
   write(S), nl.

print_proof([Goal,Rule|Subproofs],M):-
    nl,
    write(Rule),
    spaces(M),
    write(Goal),
    N is M + 3,
    print_subproofs(Subproofs,N).
print_subproofs([P|Ps],N):-
    print_proof(P,N),
    print_subproofs(Ps,N).
print_subproofs([],_).

spaces(0).
spaces(N):- N > 0, M is N -1, print(' '), spaces(M).

% Utilities

appendn([],[]).             % Relates a list of lists (first arg)
appendn([L1|Ls],L) :-       % to result of appending the sublists
        append(L1,L2,L),    % together (second arg)
        appendn(Ls,L2).

append([],L,L).
append([H|L1],L2,[H|L]) :-
        append(L1,L2,L).
```

A.2 Illustrative log

```
murray% prolog
C-Prolog version 1.5
| ?- [parser].
parser consulted 6992 bytes 0.65 sec.

yes
| ?- test(1).

Goal sequent:  [x/y\z] => x/y\z

Non-normal form theorem proving:

\R       [x/y\z:A] => x/y\z:B@C@A`B`C
/R          [z:B,x/y\z:A] => x/y:C@A`B`C
\L             [z:B,x/y\z:A,y:C] => x:A`B`C
/L                [x/y:A`B,y:C] => x:A`B`C
axiom                [x:A`B`C] => x:A`B`C
axiom                [y:C] => y:C
axiom             [z:B] => z:B

B@C@A`B`C

\R       [x/y\z:A] => x/y\z:B@C@A`B`C
\L          [z:B,x/y\z:A] => x/y:C@A`B`C
/R             [x/y:A`B] => x/y:C@A`B`C
/L                [x/y:A`B,y:C] => x:A`B`C
axiom                [x:A`B`C] => x:A`B`C
axiom                [y:C] => y:C
axiom          [z:B] => z:B

B@C@A`B`C

\R       [x/y\z:A] => x/y\z:B@A`B
\L          [z:B,x/y\z:A] => x/y:A`B
axiom          [x/y:A`B] => x/y:A`B
axiom          [z:B] => z:B

B@A`B

axiom    [x/y\z:A] => x/y\z:A

A
```

```
Normal form theorem proving:

\R      [x/y\z:A] =>1 x/y\z:B@C@A`B`C
/R         [z:B,x/y\z:A] =>1 x/y:C@A`B`C
2->1          [z:B,x/y\z:A,y:C] =>1 x:A`B`C
\L               [z:B]+(x/y\z:A)+[y:C] =>2 x:A`B`C
/L                  []+(x/y:A`B)+[y:C] =>2 x:A`B`C
axiom                  []+(x:A`B`C)+[] =>2 x:A`B`C
2->1                   [y:C] =>1 y:C
axiom                     []+(y:C)+[] =>2 y:C
2->1                [z:B] =>1 z:B
axiom                  []+(z:B)+[] =>2 z:B

B@C@A`B`C

yes
| ?- test(2).

Goal sequent:  [x/y,y/z,z] => x

Non-normal form theorem proving:

/L      [x/y:A,y/z:B,z:C] => x:A`(B`C)
axiom      [x:A`(B`C)] => x:A`(B`C)
/L         [y/z:B,z:C] => y:B`C
axiom         [y:B`C] => y:B`C
axiom         [z:C] => z:C

A`(B`C)

/L      [x/y:A,y/z:B,z:C] => x:A`(B`C)
/L         [x/y:A,y:B`C] => x:A`(B`C)
axiom         [x:A`(B`C)] => x:A`(B`C)
axiom         [y:B`C] => y:B`C
axiom      [z:C] => z:C

A`(B`C)

Normal form theorem proving:

2->1    [x/y:A,y/z:B,z:C] =>1 x:A`(B`C)
/L         []+(x/y:A)+[y/z:B,z:C] =>2 x:A`(B`C)
axiom         []+(x:A`(B`C))+[] =>2 x:A`(B`C)
2->1          [y/z:B,z:C] =>1 y:B`C
/L               []+(y/z:B)+[z:C] =>2 y:B`C
axiom               []+(y:B`C)+[] =>2 y:B`C
2->1                [z:C] =>1 z:C
axiom                  []+(z:C)+[] =>2 z:C

A`(B`C)
```

```
yes
| ?- test(3).

Goal sequent:  [x/y,y/z,z/w] => x/w

Non-normal form theorem proving:

/R       [x/y:A,y/z:B,z/w:C] => x/w:D@A‘(B‘(C‘D))
/L          [x/y:A,y/z:B,z/w:C,w:D] => x:A‘(B‘(C‘D))
axiom          [x:A‘(B‘(C‘D))] => x:A‘(B‘(C‘D))
/L             [y/z:B,z/w:C,w:D] => y:B‘(C‘D)
axiom             [y:B‘(C‘D)] => y:B‘(C‘D)
/L                [z/w:C,w:D] => z:C‘D
axiom                [z:C‘D] => z:C‘D
axiom                [w:D] => w:D

D@A‘(B‘(C‘D))

/R       [x/y:A,y/z:B,z/w:C] => x/w:D@A‘(B‘(C‘D))
/L          [x/y:A,y/z:B,z/w:C,w:D] => x:A‘(B‘(C‘D))
axiom          [x:A‘(B‘(C‘D))] => x:A‘(B‘(C‘D))
/L             [y/z:B,z/w:C,w:D] => y:B‘(C‘D)
/L                [y/z:B,z:C‘D] => y:B‘(C‘D)
axiom                [y:B‘(C‘D)] => y:B‘(C‘D)
axiom                [z:C‘D] => z:C‘D
axiom             [w:D] => w:D

D@A‘(B‘(C‘D))

/R       [x/y:A,y/z:B,z/w:C] => x/w:D@A‘(B‘(C‘D))
/L          [x/y:A,y/z:B,z/w:C,w:D] => x:A‘(B‘(C‘D))
/L             [x/y:A,y:B‘(C‘D)] => x:A‘(B‘(C‘D))
axiom             [x:A‘(B‘(C‘D))] => x:A‘(B‘(C‘D))
axiom             [y:B‘(C‘D)] => y:B‘(C‘D)
/L             [z/w:C,w:D] => z:C‘D
axiom             [z:C‘D] => z:C‘D
axiom             [w:D] => w:D

D@A‘(B‘(C‘D))

/R       [x/y:A,y/z:B,z/w:C] => x/w:D@A‘(B‘(C‘D))
/L          [x/y:A,y/z:B,z/w:C,w:D] => x:A‘(B‘(C‘D))
/L             [x/y:A,y/z:B,z:C‘D] => x:A‘(B‘(C‘D))
axiom             [x:A‘(B‘(C‘D))] => x:A‘(B‘(C‘D))
/L                [y/z:B,z:C‘D] => y:B‘(C‘D)
axiom                [y:B‘(C‘D)] => y:B‘(C‘D)
axiom                [z:C‘D] => z:C‘D
axiom          [w:D] => w:D

D@A‘(B‘(C‘D))
```

```
/R      [x/y:A,y/z:B,z/w:C] => x/w:D@A`(B`(C`D))
/L         [x/y:A,y/z:B,z/w:C,w:D] => x:A`(B`(C`D))
/L            [x/y:A,y/z:B,z:C`D] => x:A`(B`(C`D))
/L               [x/y:A,y:B`(C`D)] => x:A`(B`(C`D))
axiom               [x:A`(B`(C`D))] => x:A`(B`(C`D))
axiom               [y:B`(C`D)] => y:B`(C`D)
axiom            [z:C`D] => z:C`D
axiom         [w:D] => w:D

D@A`(B`(C`D))

Normal form theorem proving:

/R      [x/y:A,y/z:B,z/w:C] =>1 x/w:D@A`(B`(C`D))
2->1       [x/y:A,y/z:B,z/w:C,w:D] =>1 x:A`(B`(C`D))
/L            []+(x/y:A)+[y/z:B,z/w:C,w:D] =>2 x:A`(B`(C`D))
axiom            []+(x:A`(B`(C`D)))+[] =>2 x:A`(B`(C`D))
2->1             [y/z:B,z/w:C,w:D] =>1 y:B`(C`D)
/L                  []+(y/z:B)+[z/w:C,w:D] =>2 y:B`(C`D)
axiom                  []+(y:B`(C`D))+[] =>2 y:B`(C`D)
2->1                   [z/w:C,w:D] =>1 z:C`D
/L                        []+(z/w:C)+[w:D] =>2 z:C`D
axiom                        []+(z:C`D)+[] =>2 z:C`D
2->1                         [w:D] =>1 w:D
axiom                           []+(w:D)+[] =>2 w:D

D@A`(B`(C`D))

yes
| ?- test(4).

Goal sequent:  [x/x,x,x\x] => x

Non-normal form theorem proving:

/L      [x/x:A,x:B,x\x:C] => x:C`(A`B)
\L         [x:A`B,x\x:C] => x:C`(A`B)
axiom         [x:C`(A`B)] => x:C`(A`B)
axiom         [x:A`B] => x:A`B
axiom      [x:B] => x:B

C`(A`B)
```

```
/L        [x/x:A,x:B,x\x:C] => x:A‘(C‘B)
axiom        [x:A‘(C‘B)] => x:A‘(C‘B)
\L           [x:B,x\x:C] => x:C‘B
axiom           [x:C‘B] => x:C‘B
axiom           [x:B] => x:B

A‘(C‘B)

\L        [x/x:A,x:B,x\x:C] => x:C‘(A‘B)
axiom        [x:C‘(A‘B)] => x:C‘(A‘B)
/L           [x/x:A,x:B] => x:A‘B
axiom           [x:A‘B] => x:A‘B
axiom           [x:B] => x:B

C‘(A‘B)

\L        [x/x:A,x:B,x\x:C] => x:A‘(C‘B)
/L           [x/x:A,x:C‘B] => x:A‘(C‘B)
axiom           [x:A‘(C‘B)] => x:A‘(C‘B)
axiom           [x:C‘B] => x:C‘B
axiom        [x:B] => x:B

A‘(C‘B)

Normal form theorem proving:

2->1      [x/x:A,x:B,x\x:C] =>1 x:A‘(C‘B)
/L           []+(x/x:A)+[x:B,x\x:C] =>2 x:A‘(C‘B)
axiom           []+(x:A‘(C‘B))+[] =>2 x:A‘(C‘B)
2->1            [x:B,x\x:C] =>1 x:C‘B
\L                 [x:B]+(x\x:C)+[] =>2 x:C‘B
axiom                 []+(x:C‘B)+[] =>2 x:C‘B
2->1                  [x:B] =>1 x:B
axiom                    []+(x:B)+[] =>2 x:B

A‘(C‘B)

2->1      [x/x:A,x:B,x\x:C] =>1 x:C‘(A‘B)
\L           [x/x:A,x:B]+(x\x:C)+[] =>2 x:C‘(A‘B)
axiom           []+(x:C‘(A‘B))+[] =>2 x:C‘(A‘B)
2->1            [x/x:A,x:B] =>1 x:A‘B
/L                 []+(x/x:A)+[x:B] =>2 x:A‘B
axiom                 []+(x:A‘B)+[] =>2 x:A‘B
2->1                  [x:B] =>1 x:B
axiom                    []+(x:B)+[] =>2 x:B

C‘(A‘B)
```

```
yes
| ?- test(5).

Goal sequent:  [x,y,w\x/z\y] => w/z

Non-normal form theorem proving:

/R       [x:A,y:B,w\x/z\y:C] => w/z:D@C`B`D`A
\L          [x:A,y:B,w\x/z\y:C,z:D] => w:C`B`D`A
/L             [x:A,w\x/z:C`B,z:D] => w:C`B`D`A
\L                [x:A,w\x:C`B`D] => w:C`B`D`A
axiom                [w:C`B`D`A] => w:C`B`D`A
axiom                [x:A] => x:A
axiom             [z:D] => z:D
axiom          [y:B] => y:B

D@C`B`D`A

\L       [x:A,y:B,w\x/z\y:C] => w/z:D@C`B`D`A
/R          [x:A,w\x/z:C`B] => w/z:D@C`B`D`A
/L             [x:A,w\x/z:C`B,z:D] => w:C`B`D`A
\L                [x:A,w\x:C`B`D] => w:C`B`D`A
axiom                [w:C`B`D`A] => w:C`B`D`A
axiom                [x:A] => x:A
axiom             [z:D] => z:D
axiom       [y:B] => y:B

D@C`B`D`A

Normal form theorem proving:

/R       [x:A,y:B,w\x/z\y:C] =>1 w/z:D@C`B`D`A
2->1        [x:A,y:B,w\x/z\y:C,z:D] =>1 w:C`B`D`A
\L             [x:A,y:B]+(w\x/z\y:C)+[z:D] =>2 w:C`B`D`A
/L                [x:A]+(w\x/z:C`B)+[z:D] =>2 w:C`B`D`A
\L                   [x:A]+(w\x:C`B`D)+[] =>2 w:C`B`D`A
axiom                   []+(w:C`B`D`A)+[] =>2 w:C`B`D`A
2->1                    [x:A] =>1 x:A
axiom                      []+(x:A)+[] =>2 x:A
2->1                 [z:D] =>1 z:D
axiom                   []+(z:D)+[] =>2 z:D
2->1              [y:B] =>1 y:B
axiom                [ ]+(y:B)+[] =>2 y:B

D@C`B`D`A
```

```
yes
| ?- test(6).

Goal sequent:  [x/(y/z),y/w,w/z] => x

Non-normal form theorem proving:

/L       [x/(y/z):A,y/w:B,w/z:C] => x:A`(D@B`(C`D))
axiom       [x:A`(D@B`(C`D))] => x:A`(D@B`(C`D))
/R          [y/w:B,w/z:C] => y/z:D@B`(C`D)
/L             [y/w:B,w/z:C,z:D] => y:B`(C`D)
axiom             [y:B`(C`D)] => y:B`(C`D)
/L                [w/z:C,z:D] => w:C`D
axiom                [w:C`D] => w:C`D
axiom                [z:D] => z:D

A`(D@B`(C`D))

/L       [x/(y/z):A,y/w:B,w/z:C] => x:A`(D@B`(C`D))
axiom       [x:A`(D@B`(C`D))] => x:A`(D@B`(C`D))
/R          [y/w:B,w/z:C] => y/z:D@B`(C`D)
/L             [y/w:B,w/z:C,z:D] => y:B`(C`D)
/L                [y/w:B,w:C`D] => y:B`(C`D)
axiom                [y:B`(C`D)] => y:B`(C`D)
axiom                [w:C`D] => w:C`D
axiom             [z:D] => z:D

A`(D@B`(C`D))

Normal form theorem proving:

2->1     [x/(y/z):A,y/w:B,w/z:C] =>1 x:A`(D@B`(C`D))
/L          []+(x/(y/z):A)+[y/w:B,w/z:C] =>2 x:A`(D@B`(C`D))
axiom          []+(x:A`(D@B`(C`D)))+[] =>2 x:A`(D@B`(C`D))
/R             [y/w:B,w/z:C] =>1 y/z:D@B`(C`D)
2->1              [y/w:B,w/z:C,z:D] =>1 y:B`(C`D)
/L                   []+(y/w:B)+[w/z:C,z:D] =>2 y:B`(C`D)
axiom                   []+(y:B`(C`D))+[] =>2 y:B`(C`D)
2->1                    [w/z:C,z:D] =>1 w:C`D
/L                         []+(w/z:C)+[z:D] =>2 w:C`D
axiom                         []+(w:C`D)+[] =>2 w:C`D
2->1                          [z:D] =>1 z:D
axiom                            []+(z:D)+[] =>2 z:D

A`(D@B`(C`D))

yes
| ?-
```

Appendix B

Bibliography

Ades, A.E. and Steedman, M.J. 1982. 'On the order of words.' *Linguistics and Philosophy*, **4**. 517–558.

Ajdukiewicz, K. 1935. 'Die syntaktische Konnexität.' *Studia Philosophica* **1**, 1–27. Translated as 'Syntactic connexion' in S. McCall (Ed), 1967, *Polish Logic: 1920–1939*, Oxford University Press, Oxford, 207–231.

Andrews, A. D. 1975. *Studies in the Syntax of Relative and Comparative Clauses.* Ph.D. Thesis, MIT. Published by Garland Publishing Inc., New York.

Bach, E. 1979. 'Control in Montague grammar.' *Linguistic Inquiry* **10**, 515–531.

Bach, E. 1980. 'In defense of passive.' *Linguistics and Philosophy* **3**, 297–341.

Bach, E. and Partee, B.H. 1980. 'Anaphora and semantic structure.' In K.J. Kreiman, and A.E. Ojeda (Eds), *Papers from the Parasession on Pronouns and Anaphora*, Chicago Linguistic Society.

Bar-Hillel, Y. 1953. 'A Quasi-arithmetical Notation for Syntactic Description.' *Language* **29**. 47–58.

Barss, A. and Lasnik, H. 1986. 'A note on Anaphora and Double Objects.' *Linguistic Inquiry*, **17**. 47–58.

Barry, G. and Pickering, M. 1990. 'Dependency and Constituency in Categorial Grammar.' In Barry, G. and Morrill, G. (Eds), *Studies in Categorial Grammar.* Edinburgh Working Papers in Cognitive Science, Volume 5. Centre for Cognitive Science, University of Edinburgh.

van Benthem, J. 1986. *Essays in Logical Semantics.* Studies in Linguistics and Philosophy, Volume 8, D. Reidel, Dordrecht. 123–150.

van Benthem, J. 1989. Categorial grammar meets unification. Ms., Institute for Language, Logic and Information, University of Amsterdam.

Bouma, G. 1987. 'A Unification-Based Analysis of Unbounded Dependencies in Categorial Grammar.' In Groenendijk, J., Stokhof, M. and Veltman, F. (Eds), *Sixth Amsterdam Colloquium*, Institute for Language, Logic and Information, University of Amsterdam, Amsterdam, 1987. 1-19.

Bouma, G. 1988. 'Topikalisatie en Konstituent Struktuur in het Duits.' Glot, **10.3**.

Bresnan, J. W. 1974. 'The position of certain clause particles in phrase structure.' *Linguistic Inquiry*, **5**. 614-619.

Chierchia, G. 1988. 'Aspects of a categorial theory of binding.' In Oehrle, R., Bach, E. and Wheeler, D. (Eds), *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht. 125–151.

Chomsky, N. 1973. 'Conditions on transformations.' In Anderson, S. and Kiparsky, P. (Eds), *A Festschrift for Morris Halle.* Holt, Rinehard and Winston, New York.

Chomsky, N. 1975. 'Questions of Form and Interpretation.' *Linguistic Analysis,* **2**. 303–351.

Chomsky, N. 1981. *Lectures on Government and Binding.* Foris, Dordrecht.

Chomsky, N. 1982. *Concepts and Consequences of the Theory of Government and Binding.* MIT Press, Cambridge, MA.

Chomsky, N. 1986. *Barriers.* MIT Press, Cambridge, MA.

Curry, H.B. and Feys, R. 1958. *Combinatory Logic*, Volume I, North-Holland, Amsterdam.

Desclés, J-P., Guentchéva, Z. and Shaumyan, S. 1986. 'Theoretical analysis of reflexivisation in the framework of applicative grammar.' *Lingvisticæ Investigationes*, **X:1**. 1–65.

Dowty, D. 1978. 'Governed transformations as lexical rules in a Montague Grammar.' *Linguistic Inquiry,* **9**. 393–426.

Dowty, D. 1980. 'Comments on the Paper by Bach and Partee.' In K.J. Kreiman, and A.E. Ojeda (Eds), *Papers from the Parasession on Pronouns and Anaphora*, Chicago Linguistic Society.

Dowty, D. 1982a. 'Grammatical relations and Montague grammar.' In P. Jacobson and G.K. Pullum (Eds), *The Nature of Syntactic Representation*, D. Reidel, Dordrecht, 79–130.

Dowty, D. 1982b. 'More on the categorial analysis of grammatical relations.' In A. Zaenen (Ed), *Subjects and Other Subjects: Proceedings of the Harvard Conference on Grammatical Relations*, Bloomington, Indiana. Also in *Ohio State University Working Papers in Linguistics*, **26**. 102–133.

Dowty, D. 1988. 'Type raising, function composition, and non-constituent conjunction.' In Oehrle, R., Bach, E. and Wheeler, D. (Eds), *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht. 153–197.

Emms, M. 1990. 'Polymorphic Quantifiers.' In Barry, G. and Morrill, G. (Eds), *Studies in Categorial Grammar.* Edinburgh Working Papers in Cognitive Science, Volume 5. Centre for Cognitive Science, University of Edinburgh.

Emonds, J. 1976. *A Transformational Approach to English Syntax.* Academic Press, New York.

Engdahl, E. 1982. 'Restrictions on Unbounded Dependencies in Swedish.' In Engdahl, E. and Ejerhed, E. (Eds), *Readings on Unbounded Dependencies in Scandinavian Languages.*

Engdahl, E. 1985. 'Non-argument Questions.' Paper presented to the Comparative Germanic workshop in Reykjavik, June 1985.

Gawron, J.M. and Peters, S. 1988. *Anaphora and Quantification in Situation Semantics.* ms, Stanford University.

Gazdar, G. 1981. 'Unbounded dependencies and coordinate structure.' *Linguistic Inquiry,* **12**. 155-184.

Gazdar, G., Klein, E., Pullum, G. and Sag, I. 1985. *Generalized Phrase Structure Grammar.* Basil Blackwell, London.

Geach, P.T. 1972. 'A program for syntax.' In Davidson, D. and Harman, G. (Eds), *Semantics of Natural Language*, D. Reidel, Dordrecht. 483–497.

Gentzen, G. 1936. 'On the meanings of the logical constants.' In Szabo (Ed), 1969, *The Collected Papers of Gerhard Gentzen*, North Holland, Amsterdam.

Girard, J-Y. 1987. 'Linear Logic.' *Theoretical Computer Science*, **50**. 1–102.

Girard, J-Y. 1989. 'Towards a geometry of interaction.' *Proceedings of the AMS Conference on Categories, Logic and Computer Science.*

Grosu, A. 1972. 'The Strategic Content of Island Constraints.' In *Ohio State University Working Papers in Linguistics*, **13**.

Haider, H. 1986. 'V-Second in German.' In Haider, H. and Prinzhorn, M. (Eds), *Verb Second Phenomena in Germanic Languages.* Foris, Dordrecht.

Hendriks, H. 1988. 'Type change in semantics: the scope of quantification and coordination.' In Klein, E. and van Benthem, J. (Eds), *Categories, Polymorphism and Unification*, Institute for Language, Logic and Information, University of Amsterdam, and Centre for Cognitive Science, University of Edinburgh. 95–119.

Hendriks, H. 1990. 'Flexible Montague grammar.' In Hendriks, H. and Moortgat, M. (Eds), *Theory of Flexible Interpretation.* Esprit DYANA Deliverable R1.2.A, Institute for Language, Logic and Information, University of Amsterdam.

Hepple, M. 1987. *Methods for Parsing Combinatory Categorial Grammar and the Spurious Ambiguity Problem.* Unpublished M.Sc. thesis, Centre for Cognitive Science, University of Edinburgh.

Hepple, M. 1989. 'Islands and Extraction in Categorial Grammar.' Ms, Centre for Cognitive Science, University of Edinburgh.

Hepple, M. 1990a. 'Grammatical relations and Lambek calculus.' In *Proceedings of the Symposium on Discontinuous Constituency*, Institute for Language Technology and Information, University of Tilburg.

Hepple, M. 1990b. 'Verb Movement in Dutch and English.' In Reape, M. and Engdahl, E. (Eds), *Parametric Variation in Germanic and Romance.* ESPRIT deliverable DYANA R.1.1.A. Edinburgh.

Hepple, M. 1990c. 'Word Order and Obliqueness in Categorial Grammar.' In Barry, G. and Morrill, G. (Eds), *Studies in Categorial Grammar.* Edinburgh Working Papers in Cognitive Science, Volume 5. Centre for Cognitive Science, University of Edinburgh.

Hepple, M. 1990d. 'Normal form theorem proving for the Lambek calculus.' In Karlgren, H. (Ed), *Proceedings of COLING 1990.*

Hepple, M. and Morrill, G. 1989. 'Parsing and derivational equivalence.' In *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, UMIST, Manchester.

Hindley, J.R. and Seldin, J.P. 1986. *Introduction to Combinators and λ-Calculus*, Cambridge University Press, Cambridge.

Hobbs, J.R. and Rosenschein, S.J. 1978. 'Making computational sense of Montague's intensional logic.' *Artificial Intelligence*, **9**. 287–306.

Hoeksema, J. 1983. Plurality and Conjunction. In ter Meulen, A. (Ed), *Studies in Model Theoretic Semantics*, Foris, Dordrecht.

Hoeksema, J. 1985. 'Wazdat? – Contracted Forms and Verb Second in Dutch.' In Farland, J.T. (Ed), *Germanic Linguistics.* Indiana University Linguistics Club, Bloomington, Indiana.

Hughes, G.E. and Cresswell, M.J. 1984. *A Companion to Modal Logic*, Methuen, London.

Jackendoff, R. 1972. *Semantic Interpretation in Generative Grammar*, MIT Press, Cambridge, Mass.

Jacobson, P. 1987. 'Phrase Structure, Grammatical Relations and Discontinuous Constituents.' In Huck, G.J. and Ojeda, A.E. (Eds), *Syntax and Semantics*, **20**: *Discontinuous Constituency*, Academic Press, New York. 27–69.

Johnson, M. 1986. 'A GPSG account of VP structure in German.' *Linguistics*, **24**. 871-882.

Kang, B. 1988a. *Functional Inheritance, Anaphora and Semantic Interpretation in a Generalized Categorial Grammar*, Ph.D. Thesis, Brown University, Providence RI.

Kang, B. 1988b. 'Unbounded Reflexives.' *Linguistics and Philosphy*, **11**. 415–456.

Kayne, R. 1981. 'Unambiguous Paths.' In May, R. and Koster, J. (Eds), *Levels of Syntactic Representation*, Foris, Dordrecht.

König, E. 1989, 'Parsing as natural deduction.' In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Vancouver.

Koster, J. 1975. 'Dutch as an SOV language.' *Linguistic analysis*, **1**. 111–136.

Koster, J. 1978. *Locality Principles in Syntax.* Foris, Dordrecht.

Koster, J. 1987. *Domains and Dynasties: The Radical Autonomy of Syntax.* Foris, Dordrecht.

Koster, J. 1988. 'The residual SOV structure of English.' *Groningen Papers in Theoretical Linguistics*, TENK 5, University of Groningen.

Kuno, S. 1987. *Functional Syntax: Anaphora, Discourse and Empathy.* University of Chicago Press, Chicago.

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

Lambek, J. 1961. 'On the calculus of syntactic types.' In *Structure of Language and its Mathematical Aspects*, Proceedings of the Symposia in Applied Mathematics XII, American Mathematical Society, Providence.

Lapointe, S. 1980. *A Theory of Grammatical Agreement.* Ph.D. dissertation. University of Massachusets, Amherst.

Larson, R. 1988. 'On the double object construction.' *Linguistic Inquiry 19.* 335–391.

Leslie, N. 1990. 'Contrasting Styles of Categorial Derivations.' In Barry, G. and Morrill, G. (Eds), *Studies in Categorial Grammar.* Edinburgh Working Papers in Cognitive Science, Volume 5. Centre for Cognitive Science, University of Edinburgh.

Levine, R. D. 1985. 'Right Node (Non-)Raising.' *Linguistic Inquiry*, **16**. 492-497.

Lightfoot, D. 1979. *Principles of Diachronic Syntax.* Cambridge University Press, Cambridge.

Maling, J. and Zaenen, A. 1982. 'The Nonuniversality of a Surface Filter.' *Linguistic Inquiry*, **9**. 475–497.

McCawley, J.D. 1982. 'Parentheticals and Discontinuous Phrase Structure.' *Linguistic Inquiry*, **13**. 91-106.

Montague, R. 1973. 'The proper treatment of quantification in ordinary English.' In Hintikka, J., Moravcsik, J.M.E. and Suppes, P. (Eds), *Approaches to Natural Language*, D. Reidel Dordrecht. Reprinted in Thomason, R.H. (Ed), 1974, *Formal Philosophy: Selected Papers of Richard Montague*, Yale University Press, New Haven. 247–270.

Moortgat, M. 1988. 'Mixed Composition and Discontinuous Dependencies'. Paper to the Conference on Categorial Grammar, Tucson, AR, June 1985. In Oehrle, R., Bach, E. and Wheeler, D. (Eds), *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht. 349–389.

Moortgat, M. 1989. *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus*, Foris, Dordrecht.

Moortgat, M. 1990a. 'Unambiguous proof representations for the Lambek calculus.' In Proceedings of the Seventh Amsterdam Colloquium.

Moortgat, M. 1990b. 'The quantification calculus.' In Hendriks, H. and Moortgat, M. (Eds), *Theory of Flexible Interpretation.* Esprit DYANA Deliverable R1.2.A, Institute for Language, Logic and Information, University of Amsterdam.

Moortgat, M. 1990c. 'The logic of discontinuous type constructors.' In *Proceedings of the Symposium on Discontinuous Constituency*, Institute for Language Technology and Information, University of Tilburg.

Morrill, G. 1987. 'Phrase Structure Grammar and Categorial Grammar.' In van Benthem, J. and Klein, E. (eds.) *Proceedings of the Conferences in Amsterdam and Stirling*, Universities of Edinburgh and Amsterdam.

Morrill, G. 1988, *Extraction and Coordination in Phrase Structure Grammar and Categorial Grammar.* Ph.D. dissertation, Centre for Cognitive Science, University of Edinburgh.

Morrill, G. 1989. 'Intensionality, boundedness, and modal logic.' Research Paper EUCCS/RP–32, Centre for Cognitive Science, University of Edinburgh.

Morrill, G. 1990a. 'Grammar and Logical Types.' In *Proceedings of the Seventh Amsterdam Colloquium*, University of Amsterdam. An extended version appears in Barry, G. and Morrill, G. (Eds), *Studies in Categorial Grammar.* Edinburgh Working Papers in Cognitive Science, Volume 5. Centre for Cognitive Science, University of Edinburgh, 1990.

Morrill, G. 1990b. 'Intensionality and boundedness.' To appear: *Linguistics and Philosophy.*

Morrill, G., Leslie, N., Hepple, M. and Barry, G. 1990. 'Categorial deductions and structural operations.' In Barry, G. and Morrill, G. (Eds), *Studies in Categorial Grammar.* Edinburgh Working Papers in Cognitive Science, Volume 5. Centre for Cognitive Science, University of Edinburgh.

Nerbonne, J. 1986. ''Phantoms' and German fronting: poltergeist constituents?' *Linguistics*, **24**. 857-870

Partee, B.H. 1976. 'Some transformational extensions of Montague Grammar.' In Partee, B.H. (Ed), *Montague Grammar*, Academic Press, New York.

Platzack, C. 1986a. 'The Position of the Finite Verb in Swedish.' In Haider, H. and Prinzhorn, M. (Eds), *Verb Second Phenomena in Germanic Languages.* Foris, Dordrecht.

Platzack, C. 1986b. 'COMP, INFL, and Germanic Word Order'. In Hellan, L. and Christensen, K.K. (Eds), *Topics in Scandinavian Syntax*, D. Reidel, Dordrecht.

Prawitz, D. 1965. *Natural Deduction: a Proof Theoretical Study*, Almqvist and Wiksell, Uppsala.

Pereira, F. 1989. 'A calculus for Semantic Composition and Scoping.' ACL Proceedings, Vancouver. 152–160.

Postal, P.M. 1974. *On Raising.* Cambridge, Mass, MIT Press.

Pollard, C. and Sag, I.A. 1987. *An Information-Based Approach to Syntax and Semantics: Volume 1 Fundamentals.* CSLI Lecture Notes, Number 13, Center for the Study of Language and Information, Stanford.

Pollard, C and Sag, I.A. 1990. 'Anaphors in English and the Scope of Binding Theory.' Unpublished manuscript, Stanford University, CA.

Reape, M. and Hepple, M. 1990. 'Word Order and Constituency in West Continental Germanic'. In Reape, M. and Engdahl, E. (Eds), *Parametric Variation in Germanic and Romance.* ESPRIT deliverable DYANA R.1.1.A. Edinburgh.

Reinhart, T. 1983. 'Coreference and Bound Anaphora: A Restatement of the Anaphora Questions.' *Linguistics and Philosophy* **6**. 47–88.

Ross, J.R. 1967. *Constraints on variables in syntax.* Ph.D. Thesis, MIT. Indiana University Linguistics Club. Published as *Infinite Syntax!* Ablex, Norton, NJ. 1986.

Sag, I.A. 1987. 'Grammatical hierarchy and linear precedence.' In Huck, G.J. and Ojeda, A.E. (Eds), *Syntax and Semantics*, **20**: *Discontinuous Constituency*, Academic Press, New York. 303–340.

Steedman, M.J. 1985. 'Dependency and Coordination in the Grammar of Dutch and English.' *Language*, **61:3**. 523–568.

Steedman, M.J. 1987. 'Combinatory Grammars and Parasitic Gaps.' *Natural Language and Linguistic Theory*, **5:3**. 403–439.

Steedman, M.J. 1988. 'Combinators and Grammars.' Paper to the Conference on Categorial Grammar, Tucson, AR, June 1985. In Oehrle, R., Bach, E. and Wheeler, D. (Eds), *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht. 349–389.

Steedman, M.J. 1989. 'Grammar, interpretation and processing from the lexicon.' In Marslen-Wilson, W. (Ed), *Lexical Representation and Process*, MIT Press, Cambridge, MA.

Steedman, M.J. 1990. 'Gapping as Constituent Coordination.' *Linguistics and Philosophy*, **13**. 207–263.

Szabolcsi, A. 1983. ECP in categorial grammar. Ms, Max Planck Institute, Nijmegen.

Szabolcsi, A. 1987a. 'Bound variables in syntax (are there any?)' In Groenendijk, J., Stokhof, M. and Veltman, F. (Eds), *Sixth Amsterdam Colloquium*, Institute for Language, Logic and Information, University of Amsterdam.

Szabolcsi, A. 1987b. 'On Combinatory Categorial Grammar.' In *Proceedings of the Symposium on Logic and Language, Debrecen,* Akadémiai Kiadó, Budapest. 151–162.

Szabolcsi, A. 1989. 'On Combinatory Categorial Grammar and Projection from the Lexicon.' In Sag, I.A. and Szabolcsi, A. (Eds), *Lexical Matters,* CSLI Stanford/Chicago University Press, Chicago.

Szabolcsi, A. and Zwarts, F. 1990. 'Semantic Properties of Composed Functions and the Distribution of WH-Phrases.' In *Proceedings of the Seventh Amsterdam Colloquium*, ITLI, Amsterdam.

Taraldsen, K.T. 1986. 'On Verb Second and the Functional Content of Syntactic Categories.' In Haider, H. and Prinzhorn, M. (Eds), *Verb Second Phenomena in Germanic Languages,* Foris, Dordrecht.

Thomason, R. 1976. 'Some Extensions of Montague Grammar.' In Partee, B.H. (Ed), *Montague Grammar*, Academic Press, New York. 77–117.

Thrainsson, H. 1976. 'Reflexives and Subjunctives in Icelandic.' *NELS*, **6**. 225–239.

Travis, L. 1989. 'Parameters of Phrase Structure.' In Baltin, M.R. and Kroch, A.J. (Eds), *Alternative Conceptions of Phrase Structure*, University of Chicago Press. 263–279.

Uszkoreit, H. 1987a. *Word order and Constituent Structure in German.* CSLI Lecture Notes.

Uszkoreit, H. 1987b. 'Linear Precedence in Discontinuous Constituents: Complex Fronting in German.' In Huck, G.J. and Ojeda, A.E. (Eds), *Syntax and Semantics*, **20**: *Discontinuous Constituency*, Academic Press, New York. 405–425.

Wexler, K. and Culicover, P. W. 1980. *Formal Principles of Language Acquisition.* MIT.

Witten, E. 1972. 'Centrality.' Technical Report, The Computation Laboratory of Harvard University, Cambridge, Mass.

Zielonka, W. 1981. 'Axiomatizability of Ajdukiewicz-Lambek Calculus by Means of Cancellation Schemes.' *Zeitschr. f. math. Logik und Grundlagen d. Math.* **27**. 215–224.