Gapping and VP Ellipsis in a Unification-Based Grammar

Claire Gardent

Ph.D.
University of Edinburgh
1991
For Henriette and Jean Gardent
Declaration

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


Claire Gardent
Acknowledgements

I would like to thank my first supervisor, Ewan Klein and my employer, Gabriel Bes. To both of them, I am grateful for their patience, their support and the richness of their teaching; the little I know about linguistics and research, I owe mainly to them. Thanks also to my second supervisor, Robin Cooper for many helpful comments and suggestions concerning various drafts of this thesis.

On a more historical note, I am forever grateful to Margaret King for introducing me to the joys of research and encouraging me to come and do a PhD in Edinburgh. Without her, I would probably never have written this thesis.

The Centre for Cognitive Science has been a great place to be in, during the four years of my PhD. For reading chapter drafts and discussing ideas, I am grateful to Guy Barry, David Beaver, Patrick Blackburn, Kathrin Cooper, Robert Dale, Judy Delin, Martin Emms and Alex Lascarides. For making life bearable and even fun during the last stages of the writing up, thanks to Jerry, Melanie, Kathrin, David, Stephane, Lex, Martin, Jan, Dave and the infamous Gugnunk. For making me smile again as fig, \LaTeX{} and X were being smart, thanks to Lex Holt whose technical competence and unfailing cheerfulness saved me more than once from approaching despair.

Finally and most importantly, I thank my mother, Henriette Gardent, for her limitless love and support throughout the years. This thesis is dedicated to her, and to the memory of my father whose strength, simplicity and “joie de vivre”, I loved and admired. Pour ma première banana split et pour tout le reste, merci.
Abstract

The object of this thesis is to investigate the linguistic phenomena of gapping and VP ellipsis. These phenomena have been approached from a variety of perspectives in the literature: syntactic, semantic, pragmatic and computational. The aim of this thesis is to examine gapping and VP ellipsis from all these perspectives, and to attempt to develop an analysis which integrates syntactic, semantic and pragmatic information in a framework that is at least in principle computationally tractable.

At the syntactic level, a usual assumption about elliptical constructs is that they are characterised by an incomplete syntax. This raises a first question: how is this incomplete syntax licensed by the grammar? Another fact to be accounted for is that VP ellipsis, and more markedly gapping, are subject to strong syntactic constraints. This in turn raises a second question: how is the elliptical clause licensed? I show how the unification-based categorial framework provided by Unification Categorial Grammar (UCG) can be extended to answer these two questions.

At the semantic level, the following questions arise. First, how is the missing semantics of the ellipsis to be licensed? — this is the resolution problem. Second, how is the interpretation of the ellipsis to be determined? — this is the interpretation problem. Third, are there any semantic constraints on the distribution of VP and verbal ellipsis, and if so, how can they be accounted for? Here, the answers I provide are set within the framework of Dynamic Intensional Logic (DIL). First, I give a detailed answer to the resolution and to the interpretation problem. Second, I compare DIL with Discourse Representation Theory (DRT) and show that DRT is better equipped to account for Sag’s alphabetic variance constraint on VP ellipsis. I then show how DIL can be extended to mimic DRT with respect to this particular problem.

At the discourse level, I argue that VP ellipsis resolution is sensitive to discourse structure and give a tentative explanation of why this is so (the tentative character of the explanation is due to the fact that it is based on the notion of discourse relation, for which no good theory is presently available).

At the computational level, I show how the various insights developed in this thesis can be implemented in PROLOG, thus integrating them into a computational system. The resulting framework is then compared with existing algorithms for ellipsis resolution.
Finally, I discuss the relationship of VP and verbal ellipsis to other types of anaphors such as Null Complement Anaphora, *it*-anaphora and sluicing. Because of the multi-perspectival approach taken to the complex problem of VP and verbal ellipsis, it is hoped that some of the insights developed in this thesis can be used to deal with these related phenomena.
# Contents

## 1 Introduction

## 2 Linguistic Framework

2.1 Unification-based grammar formalisms ................................. 10

2.1.1 Feature structures ........................................... 11

2.1.2 Unification .................................................. 16

2.1.3 Possible extensions .......................................... 17

2.2 A unification approach to categorial grammar ......................... 19

2.2.1 The sign .................................................... 20

2.2.2 The grammar rules .......................................... 27

2.2.3 The lexicon ................................................. 31

2.3 The semantic representation language ................................. 34

2.3.1 Predication versus Unification ................................ 35

2.3.2 Dynamic Intentional Logic .................................. 37

2.3.3 Dynamic meaning .......................................... 40

2.3.4 Abbreviations ............................................. 43
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>vi</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.5 Example derivation</td>
<td>43</td>
</tr>
<tr>
<td>2.4 Summary</td>
<td>45</td>
</tr>
<tr>
<td>3 Syntax</td>
<td>47</td>
</tr>
<tr>
<td>3.1 Gapping</td>
<td>48</td>
</tr>
<tr>
<td>3.1.1 Syntactic properties of Gapping</td>
<td>48</td>
</tr>
<tr>
<td>3.1.2 Syntactic analysis of gapping</td>
<td>54</td>
</tr>
<tr>
<td>3.1.3 Analysis and data</td>
<td>77</td>
</tr>
<tr>
<td>3.1.4 Previous approaches to gapping</td>
<td>81</td>
</tr>
<tr>
<td>3.2 Verb Phrase Ellipsis</td>
<td>87</td>
</tr>
<tr>
<td>3.2.1 Syntactic properties of Verb Phrase Ellipsis</td>
<td>87</td>
</tr>
<tr>
<td>3.2.2 Analysis and data</td>
<td>91</td>
</tr>
<tr>
<td>3.2.3 Previous approaches to VPE</td>
<td>98</td>
</tr>
<tr>
<td>3.3 Summary and concluding remarks</td>
<td>101</td>
</tr>
<tr>
<td>4 Semantics I</td>
<td>103</td>
</tr>
<tr>
<td>4.1 Semantics of elliptical constructs</td>
<td>105</td>
</tr>
<tr>
<td>4.2 Sloppy identity</td>
<td>111</td>
</tr>
<tr>
<td>4.2.1 Properties of sloppy pronouns</td>
<td>111</td>
</tr>
<tr>
<td>4.2.2 Analysis</td>
<td>115</td>
</tr>
<tr>
<td>4.2.3 Comparison with other approaches</td>
<td>125</td>
</tr>
<tr>
<td>4.2.4 Remaining puzzles</td>
<td>131</td>
</tr>
<tr>
<td>4.3 Summary</td>
<td>135</td>
</tr>
</tbody>
</table>
5 Semantics II

5.1 Sag's account of VPE
5.2 VPE and Discourse Representation Theory
5.3 VPE and Dynamic Intensional Logic
5.4 A Sample Semantic Analysis
5.5 Summary

6 Discourse

6.1 Data
6.2 Discourse structure and Anaphora resolution
   6.2.1 Discourse theory: the state of the art
   6.2.2 Polanyi and Scha revisited
6.3 Explaining the data
6.4 Summary

7 Computation

7.1 Existing algorithms for ellipsis resolution
   7.1.1 Pragmatic approaches
   7.1.2 A linguistic approach
7.2 A UCG-based approach to ellipsis resolution
   7.2.1 Resolution
   7.2.2 A sample derivation
   7.2.3 Shortcomings and comparison with other approaches
CONTENTS

8 Conclusion 200

8.1 Ellipses and anaphors ........................................... 201

8.2 Further directions ............................................... 208

8.2.1 Semantic aspects ........................................... 208

8.2.2 Pragmatic aspects ........................................... 210

A Implementation 212

A.1 Implementing UCG(2) ........................................... 212

A.1.1 The lexicon ................................................. 213

A.1.2 The organisation of the lexicon ....................... 214

A.1.3 Compiling the lexicon .................................... 216

A.1.4 The grammar rules ........................................ 217

A.1.5 The parser ................................................ 218

A.2 Extending UCG(2) ............................................ 218

A.2.1 The discourse parser ..................................... 218

A.2.2 The resolver ............................................... 220

A.3 Possible improvements ....................................... 223

Bibliography 225
Chapter 1

Introduction

Anaphora is a linguistic process by which a semantically under-specified item, the anaphor, relates to some other item, the antecedent, which provides the basis for its semantic interpretation. Pronouns are paradigmatic examples of anaphors. Consider the following example.

(1.1) Jon walks. He whistles.

Here the pronoun he is an anaphor whose antecedent is the NP Jon. The denotation of this antecedent (an individual named Jon who walks) is also that of the pronoun so that the discourse in (1.1) means that there is an individual named Jon who walks and whistles.

Ellipsis can be thought of as a special sort of anaphoric process where the anaphor is syntactically 'incomplete' in some sense. Thus consider the differences between the sentences in (1.2) and the constructions in (1.3) below:

(1.2) a. Jon likes movies and Bill likes concerts.
    b. Jon gave chocolate to Jeff and Peter gave three roses to Mary.

(1.3) a. Jon likes movies and Bill concerts. (Gapping)
    b. Jon gave something to Jeff but not to Bill. (Stripping)
c. Jon likes movies and Bill does too. *(VP Ellipsis)*

d. Jon gave something to Jeff but I don’t know what. *(Sluicing)*

The constructions in (1.2) are complete in the sense that their function-argument requirements are satisfied. Thus, the verb *likes* requires two arguments, and these are provided by the NPs *Jon* and *movies* in the first clause, and *Bill* and *concerts* in the second clause. Similarly, *gave* requires three arguments and these are also provided. In contrast, the sentences in (1.3) cannot be said to be complete in this sense — they are syntactically incomplete, or ‘under-specified’, and are referred to in the literature as *ellipsis*. In each case, elements are missing which are felt to be required for the sentence to be complete. In the gapping example (1.3a), the verb *(likes)* is missing from the second conjunct. In the stripping construction, the verb *(gave)* is missing together with most of the arguments *(Jon, something)*. In the case of VP ellipsis, a VP *(likes movies)* is missing from the second clause whereas in the sluicing construction, a whole sentence *(Jon gave to Jeff)* is lacking.

The study of ellipsis then involves several sub-tasks. Firstly, some account must be provided of how the ‘incomplete’ syntax of elliptical constructs is licensed. Secondly, the conditions under which a given ellipsis may be linked (or *anaphorically related*) to a potential antecedent must be determined. Thirdly, the exact process by which the interpretation of the ellipsis obtains must be made precise.

In this thesis, I concentrate on two cases of ellipsis: VP ellipsis (or VPE) and gapping (also called, verbal ellipsis). I develop an analysis for these two phenomena which is cast within an extension of Unification Categorial Grammar *(UCG)*, called *UCG(2)*. As the name indicates, this particular framework is characterised by two main facts: it is categorial and it is unification-based. As a categorial theory of language, it supports the view that constituents are either functors or arguments which combine together under application of a limited set of rules such as backward and forward functional application. For instance, the rule of forward functional application can be represented as follows:

\[ X/Y \rightarrow X \]

where \( X \) and \( Y \) are variables ranging over categories and \( X/Y \) is a functor category which takes an argument of category \( Y \) to yield a constituent of category \( X \).

As a unification-based theory of grammar, *UCG(2)* models constituents with partially-specified feature structures containing information from different linguistic levels such as phonology, syn-
tax and semantics. Unification is then used to combine feature structures into more fully specified objects and to form relationships between the various information levels of distinct constituents. There are two main motivations for adopting a unification-based approach to grammar. First, it allows for a declarative grammar and second, it provides a simple model of how different types of linguistic information interact. With a declarative grammar, linguistic phenomena can be examined and described independently of any procedural consideration — that is, the long-standing separation introduced by Chomsky between competence and performance can be preserved. With a good model of how different levels of information interact, a more complete account of linguistic phenomena can be provided. As we shall see, this is particularly relevant when dealing with ellipsis where not only syntax, but also semantics and even pragmatics intervene in determining the acceptability and the interpretation of the elliptical construct. Finally note that a monostatal approach (that is one in which the different types of linguistic information operate in parallel) is more in line with the psychological reality of human sentence processing because it allows for on line interactions between say syntax and semantics. This has been shown (cf. [Swinney 1979; Tyler/Maralen-Wilson 1977 and Altmann 1986]) to be a more appropriate model of human sentence processing than an autonomous system in which all syntactic analysis would have to be done before semantics analysis can be attempted.

A third characteristics of the linguistic framework used in this thesis is that the semantics is a dynamic one, that is one that takes the meaning of a natural language expression to be defined by its information state update potential rather than by its truth-conditions. In the particular semantics used (i.e., Dynamic Intensional Logic or Dil), an information state contains information about potential antecedents for pronouns, very much like in Discourse Representation Theory (cf. [Kamp 1981]). As we shall see later, this has to be taken into account when defining the interpretation of elliptical constructs.

Consider then, from this perspective, the problems raised by ellipsis. The first sub-task to be dealt with concerns the syntactic licensing of elliptical constructs. Now, recall that in a categorial grammar constituents are either functors or arguments and that they combine under some simple rule schema to form higher level constituents. In UCG(2), the only binary rule used is that of functional application. This raises a problem for gapping and VPE because the chain of functor/arguments needed to derive a normal sentence is ‘broken’ by the absence of the main functor (i.e., the verb) in the case of gapping, and by the absence of the argument of the auxiliary (i.e., a VP) in the case of VP ellipsis.
Interestingly, this structural difference between VPE and gapping is accompanied by distributional and interpretive differences between the two types of ellipsis. For instance, a gapped clause must be root, whereas this is not true of VPE. Furthermore, the antecedent of a VP ellipsis may either precede or contain the ellipsis, whereas in the case of gapping, the antecedent must always precede the elliptical construct (at least in English).

Following [Chao 1987], I will take this set of structural, distributional and interpretive differences between VPE and gapping to be sufficient motivation for thinking of the two types of ellipsis as belonging to two distinct classes and consequently, motivate the fact that the two phenomena are treated differently in the grammar.

Consider gapping first. As already mentioned, gapping is characterised by the absence of the principal verbal functor. This raises the problem first of how the overt constituents appearing in the elliptical construct are to be licensed (constituents are usually licensed by their relation to some functor) and second of how the elliptical construct itself is to be licensed (higher-level, phrasal constituents are usually the 'projection' of some functor in that they result from this functor combining with one or more argument(s)). The account of gapping proposed here has two components: (i) the introduction of the product operation into the set of combinatory rules which allows for the generation of gapping structures and (ii) a reconstruction process for the derivation tree of gapped clauses which constrains the output of the product operation and provides the semantic interpretation of the elliptical clause. This provide for the licensing of gapped clauses, and for their interpretation. Constraints on the relation between ellipsis and the antecedent simply fall out of the category for coordination: a gapped clause is only licensed when coordinated with a fully-fledged sentence. The first conjunct is the antecedent clause (or a-clause).

VP ellipsis is treated very differently. First, recall that VPE is characterised in the syntax by the absence of an argument for the VPE auxiliary. This is so because typically, a categorial syntax will assign to auxiliaries the category VP/VP (i.e., a functor which takes a VP as argument and yields another VP as result category). However, in this thesis, I argue that this needs not be the case and that there are in fact two categories for auxiliaries: VP/VP and VP. That is, some auxiliaries are VP in their own right. As a result, no particular step needs to be taken to ensure the licensing of VPE clauses. To account for the interpretive and distributional properties of VP ellipsis, I develop an account which amounts to treating VPE much like ordinary pronouns in that their interpretation is provided by some entity available from the current context. Moreover, I argue that the antecedent of the VPE must be provided by
a clause in the discourse which is related by some discourse relation to the VPE-clause. Finally, following [Sag 1980], I contend that the relation between VPE and its antecedent is strongly constrained by a semantic constraint, called the alphabetic variance constraint and show how this constraint simply falls out of a dynamic semantics.

More generally, I argue (i) that VPE and gapping are very different types of ellipsis and (ii) that in both cases the relation between ellipsis and antecedent is constrained not only by syntax, but also by semantics and even pragmatics. The importance of each of this grammar component varies with the type of the ellipsis. Typically, the role of syntax is prominent in the case of gapping, whereas VPE is shown to be constrained mainly by semantics and discourse structure. If one thinks of gapping as a verbal equivalent to reflexive pronouns and of VP ellipsis as more like ordinary pronouns, these differences between the two types of ellipsis becomes understandable: like reflexives, gapping is strongly constrained by syntax, whereas like ordinary discourse pronouns, VP ellipsis is restricted mainly by discourse structure (and by semantics). To give this claim a more general support, an overall classification of how the linguistic properties of anaphors relate to a particular anaphoric class is given in the final chapter thus bringing together ellipses and anaphors. A typology is proposed which explains the differences and resemblances between gapping and VPE on the one hand and ellipses and anaphors on the other.

The chapter-by-chapter plan of the thesis is as follows.

Chapter 2 introduces the general framework within which my analysis of gapping and VP ellipsis is couched. It falls into two main parts: the first part describes the merits of unification-based grammar formalisms in general, and of UCG(2), in particular, and the second part introduces the dynamic semantics used i.e., DIL.

Chapter 3 discusses the syntax of VPE and gapping. For each type of ellipsis, the discussion falls into three distinct parts. Firstly, the cluster of properties which characterises the ellipsis is discussed. Secondly, an analysis is developed which strives to account (i) for the 'incomplete' syntax of the elliptical clause and (ii) for the syntactic constraints which may hold between ellipsis and antecedent. Thirdly, the proposed approach is compared with other existing approaches, and remaining problems and shortcomings are pointed out.

Chapter 4 concentrates on the semantics of ellipses. This time, VPE and gapping are treated on a par. First, a general proposal is put forward which indicates how elliptical constructs are to be assigned an interpretation within the framework proposed. Here, two main points are
drawn to attention. The first concerns the modelling of discourse information: if, as is the case, the semantics used is a dynamic one which encodes information about potential antecedents for pronouns, then special care must be taken when computing the interpretation of the ellipsis to preserve this information. The solution proposed is to create an alphabetic variant of the antecedent in order to interpret the ellipsis. A second point of concern when dealing with the semantics of ellipsis stems from the phenomenon of sloppy/strict ambiguity. This is best illustrated by an example. Consider the following sentence:

(1.4) Jon\textsubscript{1} washes his\textsubscript{1} car and Peter does too.

where the intended reading for the first conjunct is such that Jon washes his own car. The sloppy/strict ambiguity lies in that (1.4) has two possible readings: either Jon washes Jon's car and Peter washes Jon's car or Jon washes Jon's car and Peter washes Peter's car. The first reading is called strict, whereas the second is called sloppy. In order to give a correct account of the semantics of VPE and gapping, the strict/sloppy ambiguity of pronouns must be dealt with. This is the subject of the second part of the chapter where I argue that sloppy pronouns are subject to strong parallelism constraints, and develop a theory of parallelism which combined with the theory of semantic interpretation proposed in the first section, yields the correct results.

Chapter 5 discusses Sag's alphabetic variance constraint on VPE and shows how this constraint simply falls out of DIL dynamic semantics.

In chapter 6, I argue that ellipsis, and more particularly VP ellipsis, is subject to strong discourse constraints. The core of the argument is that the antecedent of a VPE must occur in a clause which is related by some discourse relation to the clause containing the ellipsis.

Chapter 7 shows how the syntactic and semantic analysis developed in this thesis can be integrated in a computational framework. I start by presenting the basic algorithm for resolution, and run through a simple example to illustrate how the different levels of linguistic information interact in determining the acceptability and the interpretation of VPE and gapping. I then compare this system with existing resolution algorithms for ellipsis, and show that the present algorithm yields a more thorough account of the linguistic properties characteristic of VP and verbal ellipsis. I conclude by pointing out some shortcomings in the present implementation, and suggest some directions for future research.
Finally, chapter 8 concludes the thesis by (i) taking stock of what has been achieved, (ii) giving a more general perspective of how gapping and VPE differ from each other, and from other anaphors and (iii) considering what remains to be done. A classification of ellipses and anaphors is proposed which combines ideas from [Sag/Hankamer 1984] on the one hand and [Chao 1987] on the other. This yields some more support for the claim that VPE and gapping are to be treated differently within the grammar, and provides a more general explanation of why and how, VPE and gapping differ from other types of ellipses.
Chapter 2

Linguistic Framework

Theorizing about a particular object involves constructing a formal model which reflects the interesting properties of the object under study. If it fails to yield sufficiently accurate predictions, the model is inadequate and must be revised. With respect to language, a linguistic theory is to strive for an adequate model of linguistic information. A first step towards developing such a theory involves choosing a particular framework (called *grammar formalism*) within which to work. One grammar formalism that has become increasingly popular among computational and theoretical linguists is the unification-based grammar formalism. In such a formalism, the basic theoretical objects are feature structures, and the basic combining operation is unification. Not only is this formalism theoretically sound in that it yields a precise descriptive tool for language analysis (each object and operation is mathematically well-defined and as such has an unambiguous meaning) but it is also linguistically and computationally attractive. The assets of unification-based grammar formalisms can be summarised as follows.

First, the use of feature structures as basic linguistic objects allows for all levels of linguistic information (phonologic, morphosyntactic, syntactic, semantic etc.) to be represented within a single linguistic unit. That is, the notion of category traditional from generative grammar can be broadened to include not only morphosyntactic and syntactic information but also information about e.g., phonology and semantics. Such an approach provides for an integrated model of grammar in line with such psychological investigations as [Swinney 1979; Marslen-Wilson/Tyler 1980 and Altmann 1986] where natural language processing is shown to be a multi-dimensional incremental process of information building. Grammatical theories (such as Head-driven Phrase
Structure Grammar (HPSG), Unification Categorial Grammar (UCG) and Categorial Unification Grammar (CUG)) which take advantage of this possibility and posit linguistic objects to extend over several 'dimensions' are said to be monostratal.

Second, the choice of unification as a merging operation provides a simple account of how linguistic information combines. Moreover, it means that this combination process is declarative and monotonic. It is monotonic in that information can be neither lost nor overwritten. It is declarative in that unification is order-independent: the grammar formalism can thus be used to state the linguistic knowledge embodied in the grammar, but not how it is to be used. Concretely, this means that a unification-based grammar may be used both for parsing and for generation.

Third, the adoption of a unification-based grammar formalism as a metalanguage within which to couch a particular linguistic theory means that this theory provides a computer-interpretable characterisation of natural language. Computationally, feature-structures are easy to model (e.g., as record structures in Pascal or as terms in Prolog) and unification is a natural operation: it has an underpinning in graph-theory and is for example, the fundamental operation of the Prolog interpreter.

UCG partakes of the general trend to use unification-based grammar formalisms and thus inherits of the properties just described. However as the name suggests, it also takes inspiration from categorial grammar. As such it exhibits two additional characteristics. A first characteristic is that linguistic expressions are taken to behave as functors and arguments whereby combination may only occur between a functor and some appropriate argument. This has two main advantages. First, this yields a simple account of compositionality where the properties of a phrasal sign are a function of the properties of its subconstituents. Second, when enriched with unification, it offers an elegant treatment of the merging and propagation of information within the derivation tree. For instance as we shall see, UCG allows for a direct encoding of the head feature principle familiar from GPSG, whereby the head features of the head daughter are shared by the mother node. A second characteristics of categorial grammars resides in the tight integration of syntax with semantics. This is of course fully compatible with the monostratal approach permitted by unification-based grammar formalisms.

The aim of this chapter is to introduce the reader to the framework within which the analysis of verbal and VP ellipsis proposed in this thesis is couched. It is structured as follows. First (section 2.1), I describe in some detail the various components of a unification-based grammar
formalism. The aim here is first, to indicate what the expressive power of such a formalism is and second, to give some more motivations for its adoption as a general grammar framework. In section 2.2, I describe UCG and give some examples of sentence derivations. Finally, section 2.3 presents the semantic representation language used in this thesis i.e., Dynamic Intensional Logic (or DIL). The syntax and semantics of DIL are investigated, and a sample derivation is given which illustrates the workings of DIL.

2.1 Unification-based grammar formalisms

As discussed above, there are several motivations for adopting a unification-based formalism as a basic grammar framework within which to develop a theory of language. First, it provides a precise descriptive tool for linguistic analysis. Second, it yields an appropriate language within which to describe natural language expressions as bearers of multi-dimensional partial information. Third, it is amenable to computation. In this section, I concentrate on showing how unification-based formalisms allow for a precise definition of the objects and operations used in the grammar. That is, I demonstrate that these formalisms do in fact provide a precise descriptive tool for linguistic analysis. This contrasts with a lot of work in traditional linguistics where because no precise meta-language is given, objects and operations often turns out to be ill-defined and misunderstood. The problem with this can be illustrated by a short anecdote borrowed from [Feynman 1987]: Richard Feynman, the famous physicist, sits in a philosophy class where the lecturer has been talking about the notion of crucial object for the last two hours. Suddenly, Feynman asks: Is a brick a crucial object? He received a different answer from almost every student in the assembly. The point is that the concept was so ill-defined that it was in fact, undefined. Now the situation in linguistics is perhaps not as bad, but examples of ill-defined terms and operations are not hard to find. From a theoretical point of view, this is very problematic because it means that the theory is not a theory because it is too vague to be verifiable (or falsifiable). On this particular point, unification-based grammar formalisms are better equipped than most traditional linguistic frameworks because they come with a sound mathematical foundation.

There is another theoretical advantage in resorting to unification-based formalisms to do linguistics: they are amenable to computation on the one hand, and to the theoretical investigation of their computational properties on the other. One might ask: what has computation to do with linguistics? There are two answers to this question. First, there is a theoretical point.
CHAPTER 2. LINGUISTIC FRAMEWORK

If the theory can be tested on a computer, then it means that all the terms used have to be defined in a once-and-for-all fashion simply because this is the way a computer functions: it relies on some data (e.g., an implementation of the grammar) to produce more data (e.g., an analysis of a string). The point is, the grammar a computer uses does not change depending on the user. So the problem of having ill-defined terms disappear, or at least is considerably alleviated. Note also that generally speaking, a computer provides a better way of testing a theory that any group of human beings ever will simply because it can test a lot of data very fast (providing the program embodying the theory is efficient). A second answer to the question of the relation between linguistic and computation has to do with language processing. Traditionally, a large part of theoretical linguistics is devoted to the study and modelling of linguistic knowledge. Language however is a process: we understand utterances and we produce utterances. Therefore modelling language naturally implies producing some model for language processing. It does not matter if the model has nothing to do with reality — what matters is that a model be produced which makes the right predictions about the way in which we produce and understand utterances. Clearly, computers and computer program provide as good a potential for developing such a model as any other tool. Furthermore, note that here again the fact the unification-based formalisms have a mathematical foundation comes useful. Consider for instance Johnson's work on the logic of feature structures (cf. [Johnson 1988]). The basic idea there is that feature structures are formulae of first-order logic, and therefore that parsing is just a theorem proving process. Just another model of language understanding. But because of this mathematical foundation, it becomes possible to estimate parameters such as the complexity of parsing.

Clearly, work on the semantics of unification-based formalisms and on their meta-theoretical properties is only beginning, and to date, the bulk of our knowledge about language still comes from traditional linguistics. Nevertheless, I hope to have given some motivation of why using a computer interpretable framework is important. In what follows, I describe in more detail the basic components of a unification-based formalisms and show how they are equipped with a precise mathematical foundation.

2.1.1 Feature structures

Unification-based formalisms use feature structures as basic objects. The idea behind this is that a linguistic expression conveys some partial information and that feature structures adequately model this information. So what are feature structures? They are sets of attribute-value pairs
where attributes and values form a finite set of predefined terms\(^1\). From a more linguistic perspective, feature structures are used to describe or represent the information associated with grammar objects. For instance, in a monostratal grammar we can think of constituents as sets of attribute-value pairs specifying a value for each of say, three attributes e.g., \textit{pho}, \textit{synt} and \textit{sem}.

There are basically two ways to represent feature-structures: as Attribute-Value Matrices (AVM for short) or as graphs. By way of example, suppose that the proper noun \textit{Sarah} is assigned a feature structure with values for the three attributes just mentioned, \textit{Sarah}, \textit{np} and \textit{sarah} respectively. The corresponding AVM will be:

\[
\begin{array}{c}
\text{pho} : \text{Sarah} \\
\text{synt} : \text{np} \\
\text{sem} : \text{sarah}
\end{array}
\]

Alternatively, this information can be represented in a graph as in (2.2) where attributes label the arcs of the graph and values, the nodes. In this thesis, only the AVM notation will be used.

In modelling linguistic information, three main properties make feature structures particularly attractive.

First, values need not be atomic — they can also be complex. A complex value is itself a feature structure. That is, feature structures are recursive structures (a feature structure may contain another feature structure). As an example, consider the feature structure in (2.3).

\(^1\)For a discussion of the mathematical properties of feature structures, see [Rounds/Kasper 1986; Kasper/Rounds 1986; Moshier/Rounds 1987 and Ait-Kaci 1984].
In this feature structure the value of the \textit{synt} attribute is itself a feature structure with attributes \textit{cat} and \textit{agr} whose values are \textit{np} and \textit{3sg} respectively. Given this potential for recursive embedding of feature structures within feature structures, it is convenient to have a way to refer to a particular subpart of a feature structure. This is what \textit{path attributes} (or \textit{paths} for short) are for. A path is just a finite sequence of attributes (notated e.g, \textit{synt:agr}). For example, in (2.3) the value of the path \textit{synt:agr} is the atomic value 3sg.

Another interesting property of feature structures is their potential for structure sharing: two (or more) distinct attributes (or paths) may share the same value. Here it is important to understand that "same" is to be taken as denoting token rather than type identity. This is best illustrated by an example. Consider the following structures:

\begin{align*}
\text{(2.3)} \\
\begin{array}{c}
\text{\textit{pho} : } \text{Sarah} \\
\text{\textit{synt}} : \\
\quad \text{\textit{cat}} : \text{np} \\
\quad \text{\textit{agr}} : \text{3sg} \\
\text{\textit{sem}} : \text{Sarah}
\end{array}
\end{align*}

\begin{align*}
\text{(2.4)} \\
\begin{array}{c}
\text{\textit{subj}} : \\
\quad \text{\textit{agr}} : 1
\end{array} \\
\begin{array}{c}
\text{\textit{vp}} : \\
\quad \text{\textit{agr}} : 1 \\
\quad \text{\textit{gender}} : \text{fem} \\
\quad \text{\textit{number}} : \text{sg}
\end{array}
\end{align*}
In (2.5), the two attributes subj:agr and vp:agr have two distinct values of the same type whereas in (2.4), they share a single value. The difference comes into play when some additional information further specifies the content of these feature structures. When adding information to the subj:agr attribute of (2.5), the value of vp:agr remains unchanged. By contrast, adding information to the subj:agr attribute of (2.4) means that the same information is added to the value of the subj:agr value (because there is in fact one single value shared by two attributes). A feature structure in which two attributes share the same value is said to be reentrant. As illustrated in (2.4), reentrancy is notated by using tags on values. The single value is written once and preceded by a tag. Attributes that share this value are then assigned this tag as value.

A third fundamental fact about feature structures is their varying degree of informativeness. Linguistically, a feature structure conveys information about some linguistic object. This information is often partial. By way of example, consider the following feature structures.

(2.6)  
\[
\text{sem} : \text{sarah}
\]

(2.7)  
\[
\text{syst} : \text{np}  \\
\text{sem} : \text{sarah}
\]

(2.7) is more informative than (2.6) because it contains all the information contained in (2.6) plus some more information. More generally, one can define an ordering relation on feature
structures according to their degree of informativeness. We say that a feature structure $A$ \textit{subsumes} a feature structure $B$ (written $A \sqsubseteq B$) if $B$ is at least as informative as $A$; conversely, $B$ is said to \textit{extend} $A$.

Not all feature structures can be ordered according to their degree of informativeness. There are basically two cases where two feature structures cannot be so ordered. The first occurs when the two feature structures contain incompatible information (i.e., contradictory information), the second when they contain no common information. The first case is illustrated in (2.8) and the second in (2.9).

(2.8) \[
\begin{array}{c}
\text{sem : sarah} \\
\end{array}
\begin{array}{c}
\text{sem : jeremy}
\end{array}
\]

(2.9) \[
\begin{array}{c}
\text{synt : np} \\
\text{sem : sarah}
\end{array}
\begin{array}{c}
\text{pho : Sarah} \\
\text{order : post}
\end{array}
\]

This means that subsumption is a partial (rather than a total) ordering on feature structures. More precisely, subsumption is in fact a reflexive partial ordering on feature structures which provides the feature structure domain with a natural lattice structure (we shall see when discussing unification that this is important). It is defined as follows. Let $V_X (a)$ be the value of the attribute $a$ in the feature structure $X$ for any attribute $a$ and feature structure $X$. Then we have that,

1. A variable subsumes any feature structure (because it contains no information at all).
2. A constant subsumes and is only subsumed by itself.
3. A feature structure $A$ subsumes a feature structure $B$ if for any attribute $a$ in $A$, $V_A(a) \subseteq V_B(a)$ and $V_A(p) \subseteq V_B(q)$ for all paths $p$ and $q$ such that $p = q$ in $A$ where $=$ indicates token identity i.e., reentrancy.
2.1.2 Unification

A central issue in modelling natural language processing is to determine how the linguistic information carried by the different subparts of an utterance is assembled into a structured representation as a result of parsing an utterance. Within a unification-based framework, this process is characterised via unification. Unification is a combining operation on feature structures which is based on the notion of subsumption described above. Intuitively, it is very much like set union in that unifying two feature structures requires taking the set union of the attributes contained in the two structures and combining (i.e., unifying) their values recursively. For instance, the unification of the two feature structures in (2.9) is as given in (2.10).

\[
\begin{align*}
\text{synt} : \text{np} & \land \text{pho} : \text{Sarah} \\
\text{sem} : \text{sarah} & \land \text{order} : \text{post} \\
\end{align*}
\]

\[
\begin{align*}
\text{synt} : \text{np} \land \text{pho} : \text{Sarah} \\
\text{sem} : \text{sarah} \land \text{order} : \text{post} \\
\end{align*}
\]

More generally, the unification of two feature structures A and B yields a new structure \( A \land B \) which contains all the information contained in both A and B providing this information is consistent. Mathematically, \( A \land B \) is the greatest lower bound of A and B in the lattice defined by the subsumption relation on the feature structure domain such that the top element (T) corresponds to null information and the bottom element (⊥) to contradictory information.

Alternatively, one can think of unification in terms of substitution operations. A substitution is a function assigning terms to variables, that is, a function of the form \( \sigma : \text{VAR} \rightarrow \text{TERM} \) where \( \sigma \) is the name of the substitution, \( \text{VAR} \) is a set of variables and \( \text{TERM} \) a set of terms. The application of a substitution \( \sigma \) to a feature structure \( A \), written \( [A]\sigma \) is the result of uniformly replacing all occurrences of \( v_i \) by \( \sigma(v_i) \), where \( v_i \) is any variable occurring within the domain of \( \sigma \). To unify two feature structures \( A \) and \( B \), we must find the most general substitution \( \sigma \) such that \( [A]\sigma = [B]\sigma \). \( \sigma \) is called a unifier and \( A \) and \( B \) are said to unify.

Note that the unification of two feature structures \( A \) and \( B \) does not always exists. More specifically, if \( A \) and \( B \) contain inconsistent information, then \( A \land B \) does not exist. Intuitively, this is because we do not want two information units which contain incompatible information to merge. Mathematically, this comes out of the fact that subsumption is a partial rather than
a total ordering and that on the resulting lattice structure, A and B may not have any greatest lower bound (or rather, they have a greatest lower bound but it is the 'bottom' of the lattice that is, the object in the lattice which represents inconsistent information). In terms of substitution, this follows from the fact that if two feature structures ‘disagree’ on the value of any given attribute then trivially, no substitution can be found which makes A and B equal. An example of feature structures which cannot unify is given in (2.11).

\[
\begin{array}{c|c}
\text{synt} : \text{np} & \text{synt} : \text{pp} \\
\text{sem} : \text{sarah} & \text{sem} : \text{sarah}
\end{array}
\]

To summarise, feature structures can be ordered according to a subsumption relation \( \sqsubseteq \) such that \( A \sqsubseteq B \) if B is at least as informative as A. The unification \( A \land B \) of two feature structures A and B is then the least informative feature structure which is subsumed both by A and B. Mathematically, the subsumption relation on the feature structure domain forms a meet semilattice with top (\( T \)) indicating contradictory information and bottom (\( L \)), null information. Unification corresponds to the meet operation of this lattice.

### 2.1.3 Possible extensions

So far, we have only considered feature structures whose values are either atoms or feature structures. However, linguists typically resort to information which may be either negative (e.g., barlevel \( \neq 2 \)), disjunctive (e.g., case = (nom \( \lor \) acc)) or conditional (e.g., clitic\( (X) \rightarrow \) precedes\( (X, \text{verb}) \)). This suggests that the basic unification framework presented above is lacking both expressive power and linguistic felicity. Following [Pollard/Sag 1987], I now discuss how this basic framework can be extended to include negative, disjunctive and conditional information. To start with, we extend the set of feature structures to include disjunctive information. Let us call this new domain of feature structures \( FS_2 \). A disjunctive feature value is notated \( A_1 \lor \ldots \lor A_n \) where \( A_1, \ldots, A_n \) are basic feature structures i.e., feature structures exempt of disjunctive values. Subsumption is then generalised to apply to \( FS_2 \) feature structures as follows.

**Definition 2.1 (Subsumption\(_2\))**

Let A and B be feature structures such as \( A = A_1 \lor \ldots \lor A_n \) and \( B = B_1 \lor \ldots \lor B_m \), then A subsumes\(_2 \) B iff for all \( A_i \) in A, there exists \( B_j \) in B such that \( A_i \sqsubseteq B_j \).
We call this new version of subsumption subsume\textsubscript{2} or \( \sqsubseteq_2 \) (as opposed to \( \sqsubseteq \) which is the subsumption relation on basic feature structures). Equipped with this new subsumption relation, we can now define unification on feature structures containing disjunctive specifications of values. Let us call this new version of unification \( \Lambda_2 \). Intuitively, \( A \Lambda_2 B \) will simply be the least informative feature structure such that both \( A \) and \( B \) subsume\textsubscript{2} \( A \Lambda_2 B \). Formally, \( A \Lambda_2 B \) is the meet operation of the new lattice \( L_2 \) formed by \( \sqsubseteq_2 \) on the extended set of feature structures \( FS_2 \). Conversely, the disjunction \( A \lor B \) of two feature structures will be the most informative feature structure which subsumes both \( A \) and \( B \). That is, \( A \lor B \) is the join operation of \( L_2 \).

As it turns out, \( L_2 \) is a special kind of distributive lattice called Heyting algebra. On such a lattice, the relative pseudo-complement operation is well-defined. This can be used to model conditional feature structures: the feature structure \( A \rightarrow B \) is the relative pseudo-complement of \( A \) and \( B \). More intuitively, the content of a conditional feature structure can be explained as follows. Suppose we have the conditional feature structure \( A \rightarrow B \). What we want is that if some feature structure \( X \) unifies with \( A \rightarrow B \), the result is a new feature structure \( B' \) such that \( B' \) contains (i) all the information contained in \( B \) (because it is implied by \( A \)) plus (ii) any additional information contained in \( X \) (because it unifies with \( A \)). In other words, we want \( A \rightarrow B \) to be such that the unification of \( A \rightarrow B \) with some other feature structure \( X \) is subsumed by \( B \).

Finally negation can be defined in terms of implication: the negation \( \neg A \) of a feature structure \( A \) is encoded as \( A \rightarrow \bot \).

Another possible direction in which unification-based formalisms can be extended concerns the nature of the basic values. So far, values were either atoms, feature structures or logical combinations thereof. However, one could think of some linguistic applications where values may best be set values, list values or even functionally (or relationally) dependent values. In what follows, I briefly describe how such objects might be introduced in a unification-based framework.

A list value, written \( \langle A_1, \ldots, A_n \rangle \), is simply a list of values (whose type as atoms, feature structures etc. I leave here open). Subsumption and unification for list values are defined pointwise as follows:

\[
\langle A_1, \ldots, A_n \rangle \sqsubseteq \langle B_1, \ldots, B_n \rangle \text{ iff } A_i \sqsubseteq B_i \text{ with } i = 1, \ldots, n
\]

\[
\langle A_1, \ldots, A_n \rangle \land \langle B_1, \ldots, B_n \rangle = \langle A_1 \land B_1, \ldots, A_n \land B_n \rangle
\]
Note that only lists of equal length may unify.

A set value, written \{A_1, \ldots, A_n\}, consists of a set of values. Unification of sets is notoriously complex, the problem being that there is no way to guess which object in the first set corresponds to which object in the second, so that all possible ways of matching them up must be considered.

Finally, functionally dependent values are values which functionally depends on other values in the same feature structure. As an example, consider the semantics of a pronoun within a dynamic type approach to natural language semantics. It could be assigned the value \textsc{member}(dmlist) where \textsc{dmlist} is the list of accessible discourse markers (see e.g., [Kamp 1981]). That is, it could be assigned a functional value which depend on the function \textsc{member} and on the value \textsc{dmlist}.

This concludes the description of the formal apparatus available within a unification-based formalism. The next section presents the linguistic framework used throughout this thesis i.e., ucg.

### 2.2 A unification approach to categorial grammar

Categorial grammar (cg) began with Adjukiewicz in 1935, and was later revived by Bar-Hillel in 1953. The basic intuition behind categorial grammar is (i) that linguistic constituents behave as functors and arguments and (ii) that they can be combined using a restricted set of rules which includes functional application. Typically, a functor category will be of the form \( X/Y \) where \( X \) and \( Y \) are variables ranging over categories. Functional application then ensures that functors combine with arguments of the appropriate category to yield a new category, the result category. More precisely, functional application can be represented as follows:

\[
X/Y, \ Y \Rightarrow X
\]

This is the forward version of functional application i.e., the version in which the functor \((X/Y)\) precedes the argument \((Y)\). There is also a backward version for when the functor follows the argument. This backward version is:

\[
Y, \ X\backslash Y \Rightarrow X
\]
The meaning of the notation used to represent functor categories should now become clear: a functor category of the form $X/Y$ is a category which can combine by forward application with a constituent of category $Y$ immediately following it to yield a constituent of category $X$. Similarly, a functor with category $X\backslash Y$ can combine by backward application with a constituent of category $Y$ which immediately precedes it, to yield a new constituent of category $X$.

During the last decade, this basic version of CG has been extended in two distinct but compatible directions. On the one hand, there has been proposals to extend the set of combinatory rules to be used (see for instance [Steedman 1986; Moortgaat 1988 and Morrill 1989]) and on the other hand, there has been some interest in developing versions of CG in which the basic building block of the grammar are feature structures rather than monadic categories (see [Pollard, 1985; Karttunen 1986 and Uszkoreit 1986]). UCG falls within the second paradigm.

2.2.1 The sign

Following [Pollard/Sag 1987], each lexical item in UCG is assigned a sign as category. A sign is a feature structure containing phonological, semantic, syntactic and ordering information as a conjunction of attribute-value pairs. The structure of a sign and its attributes is specified by means of type declarations. That is, UCG is a typed system. This notion of typing is akin to the notion of typing used in sorted logic and data typing of programming languages: every object is typed and the set of types form a closed system\(^2\). Intuitively, type specifications indicate restrictions on the values of attributes; they express generalizations over the structure of the grammar objects manipulated by the theory. Independent of this theoretical point, there are also some practical advantages in having a linguistic system whose objects are typed. One advantage is that it permits a consistency check on the grammar: data typing usually comes with a type-checking algorithm. This algorithm can be used to verify that the grammar written by a particular linguist is consistent with the generalizations over the structure of linguistic objects stated in their typing. Another practical advantage is that it forces one to be explicit about the structure of the object used in the theory. This might help in spotting inefficiencies or inadequacies in the overall design of the grammar.

Types are specified by type declarations. Following [Smolka 1988], I represent type declarations

\(^2\)This means that any attribute that is not explicitly defined in a particular type declaration is held to be undefined for that type. In effect, this means that type declaration express universal generalizations about the structure of the grammar objects used in UCG.
using a matrix notation where type symbols appear in bold face and attribute names in italics. In specifying type assignment, I will resort to the two types nil and basic where nil denotes a type which is only compatible with itself and basic indicates an atomic type. I will also use ⊥ to represent null information (for instance, if the value of the case attribute of some sign \( X \) is unknown or unspecified, this will be specified as case: ⊥).

A sign in UCG is assigned the following type:

\[
\begin{align*}
\text{sign} : & \quad \begin{cases} 
\text{pho : basic} & \lor \text{phonseq} \\
\text{synt : basiccat} & \lor \text{complexcat} \\
\text{sem : dilformula} \\
\text{order : basic}
\end{cases}
\end{align*}
\]

That is, an object of type sign is a feature structure with four attributes: pho, synt, sem and order. Each of this attribute is constrained by its own type declaration. Note that because the type system is closed any object of type sign which does not consist of exactly these four features is ill formed.

I now turn to the type specification of the four attributes of a sign: pho, synt, sem and order.

Phonology. Here I will take the phonological information contained in a UCG sign to be simply the orthographic realization of a word or sequence of words. A word will have type basic and thus conform to the first disjunct of the type specification for pho. By contrast, a sequence of words is of type phonseq with type specification:

\[
\begin{align*}
\text{phonseq : nil} & \lor \begin{cases} 
\text{head : basic} \\
\text{rest : + phonseq}
\end{cases}
\end{align*}
\]

where + denotes string concatenation (see [Hopcroft/Ullman 1979]). This means that the value of a phonology attribute is a sequence of one or more words. Of course, it would be desirable to have a type for phonological values which allows for richer information about phonological structure to be encoded within the sign. Recent work ([Bird 1990; Scobbie 1990]) within the field of attribute-value phonology suggests that this is possible. However, since phonology is not
the primary focus of this thesis, I will continue to conflate here phonology with orthography.

Syntax. As we have seen, a characteristic property of categorial grammar is that constituents are viewed as functors and arguments whereby the behavioral potential of each constituent is encoded in its category: a category of the form \( X/Y \) or \( X\backslash Y \) denotes a potential functor, whereas a basic category indicates an argument. A similar insight is embodied in UCG by the type specification of the \textit{synt} attribute which is:

\[
\textit{synt}: \text{basiccat} \lor \text{complexcat}
\]

This specification states that the \textit{synt} attribute of a UCG sign is of type either basiccat or complexcat. The idea is that signs with complexcat type syntactic values are potential functors, whereas signs with basiccat type are not. The two types basiccat and complexcat are specified as follows:

\[
\begin{align*}
\text{basiccat} & : \left[ \begin{array}{c}
\text{cat}: \text{basic} \\
\text{feats}: \text{featlist}
\end{array} \right] \\
\text{complexcat} & : \left[ \begin{array}{c}
\text{res}: \text{basiccat} \lor \text{complexcat} \\
\text{active}: \text{sign}
\end{array} \right]
\end{align*}
\]

That is, basic categories consist of an atomic category (i.e., np, noun or s) and a list of features which encode morphosyntactic information (e.g., agreement, case, verb features) where the type specification for \textit{featlist} is simply:

\[
\begin{align*}
\text{featlist} & : \text{nil} \lor \\
& \left[ \begin{array}{c}
\text{first}: \text{featvalue} \\
\text{rest}: \text{featlist}
\end{array} \right]
\end{align*}
\]

and the type \textit{featvalue} is defined as:

\[
\begin{align*}
\text{featvalue} & : \left[ \begin{array}{c}
\text{attrname}: \text{basic}
\end{array} \right]
\end{align*}
\]

where \textit{attrname} is part of a finite set of predefined terms. For example, the \textit{synt} attribute of the UCG sign associated with the noun \textit{cat} is:
By contrast, if the synt value of a sign is of type complexcat, then it can be thought of as having a syntactic attribute of the form Synt/Sign where Sign is a UCG sign and Synt is either a basic or a complex category. For example, the synt value of a transitive verb will be (omitting morphosyntactic features for simplicity): s/NP where NP abbreviates a sign with basic category np. In what follows, I will use the slash notation to abbreviate the AVM one. That is the AVM in (2.18) will be abbreviated to Synt/Sign if the order value of Sign is pre, and to Synt\Sign if it is post. Sign is called the active part of the functor.

Another abbreviation used throughout this thesis concerns the representation of basic categories. To enhance readability, I will not represent the internal structure of the feature list which is the value of the feats attribute. For instance, (2.17) will be abbreviated either to:
(2.19) \[
\begin{array}{c}
cat : \text{noun} \\
\text{feats} : \\
\begin{cases}
gender : \perp \\
number : \text{sg} \\
pers : 3 \\
case : \perp
\end{cases}
\end{array}
\]

or to:

(2.20) \[
\begin{array}{c}
synt : \text{noun}[\perp, \text{sg}, 3, \perp]
\end{array}
\]

More generally, given a UCG sign \( S \) whose value for the \( \text{synt} \) attribute is:

(2.21) \[
\begin{array}{c}
cat : \text{CAT} \\
\text{feats} : \text{FEATS}
\end{array}
\]

and such that the value of the attributes listed in \( \text{FEATS} \) is \( \text{FEATVALUES} \), the abbreviated form of the syntactic specification for \( S \) will be:

(2.22) \[
\begin{array}{c}
synt : \text{CAT}[\text{FEATVALUES}]
\end{array}
\]

Given these notational conventions, we can now abbreviate the sign in (2.23) to (2.24).
CHAPTER 2. LINGUISTIC FRAMEWORK

(2.23)

\[
\text{pho} : \text{Pho} \\
\text{cat} : s \\
\text{res} : \\
\text{feats} : \begin{align*}
gender & : \bot \\
\text{number} & : \text{NB} \\
\text{pers} & : \text{PERS} \\
\text{case} & : \bot
\end{align*}
\]

\[
\text{synt} : \text{Pho}^\text{A} \\
\text{cat} : \text{np} \\
\text{active} : \\
\text{synt} : \text{feats} : \begin{align*}
gender & : \bot \\
\text{number} & : \text{NB} \\
\text{pers} & : \text{PERS} \\
\text{case} & : \text{nom}
\end{align*}
\]

\[
\text{sem} : \text{Sem}^\text{A} \\
\text{order} : \text{Order}^\text{A}
\]

(2.24)

\[
\text{pho} : \text{Pho} \\
\text{synt} : s[\bot, \text{NB}, \text{PERS}, \bot]/ \\
\text{sem} : \text{Sem} \\
\text{order} : \text{Order}
\]

\[
\text{pho} : \text{Pho}^\text{A} \\
\text{synt} : \text{np}[\bot, \text{NB}, \text{PERS}, \text{nom}] \\
\text{sem} : \text{Sem}^\text{A} \\
\text{order} : \text{Order}^\text{A}
\]
Finally, I will also refer to constituents whose sign has \textit{synt} value \textit{cat} as constituents of category \textit{CAT}. For instance, rather than saying that the category of \textit{man} is a sign of the form given in (2.19), I will simply say that \textit{man} is a constituent of category \textit{NOUN} or \textit{N}. Similarly, the category of an intransitive verb given in (2.24) will be referred to in its abbreviated form, as \textit{S/NP} sign. Given these abbreviations, the following notational conventions will be used for verbs and verb phrases:

- \textbf{IV} abbreviates a sign with \textit{synt} value \textit{iv}
- \textbf{TV} abbreviates a sign with \textit{synt} value \textit{tv}
- \textbf{DV} abbreviates a sign with \textit{synt} value \textit{dv}
- \textbf{VP} abbreviates a sign of the form \textit{S/NP}
- \textbf{TVP} abbreviates a sign of the form \textit{S/NP/NP}
- \textbf{DVP} abbreviates a sign of the form \textit{S/NP/NP/NP}

\textbf{Order.} In most versions of categorial grammars, functors are directional in that their category indicates where the required argument is to be found. Thus a functor of the form \textit{X/Y} requires its argument (\textit{Y}) to immediately follow it, whereas a functor of the form \textit{X\ Y} requires its argument to immediately precede it. The \textit{order} attribute of \textit{UCG} reflects this dichotomy as follows. As indicated in (2.12), the \textit{order} attribute is of type \textit{basic}; its possible values are either \textit{pre} or \textit{post}. Intuitively, a sign whose \textit{order} attribute has value \textit{pre} is an argument which must be preceded by its functor whereas a sign whose \textit{order} attribute has value \textit{post} requires its functor to follow it (the details of how such a semantics for the \textit{order} attribute obtains are discussed in section 2.2.2). I will illustrate the import of the \textit{order} attribute by an example.

Consider the \textit{UCG} sign associated with a transitive verb. In categorial terms, a transitive verb is a functor which takes two arguments (a subject \textit{NP} and an object \textit{NP}) to yield a sentence. Moreover whereas the subject \textit{NP} must precede the verb, the object \textit{NP} must follow it. This is encoded in \textit{UCG} by assigning the \textit{synt} attribute of a transitive verb a value of the form:

\begin{equation}
\textit{synt} : s/\text{NP[Case,Order]}[\text{nom,post}]/\text{NP[acc,pre]}$
\end{equation}

where \textit{NP[Case,Order]} abbreviates a syntactic attribute of the form:
2.2.2 The grammar rules

To combine sequences of constituents into higher level constituents, traditional categorial grammar uses the rule of functional application (FA). The question therefore arises of what the equivalent UCG rule looks like. Although the basic idea is the same, some important differences result from the fact that in UCG, grammar objects are taken to convey partial, rather than total, information (as opposed to the fully specified monadic categories used in traditional CG) and that consequently unification, rather than identity, is taken to be the basic comparison operation between categories. More concretely, suppose that we have a functor sign $Sign_1$ together with some adjacent sign $Sign_2$ where $Sign_1$ is of the form\(^3\):

\(^3\)Following the PROLOG tradition, values starting with an upper case denote variables whereas lower-case letters indicate constants.
or equivalently, Pho:(Synt/Sign):Sem:Order. A unification-based equivalent of functional application must answer two questions. First, is Sign2 an appropriate argument for the functor sign Sign1? Second, what is the result of forward applying Sign1 to Sign2?

Consider the first question first. In traditional CG, an argument is of the appropriate category if its category is identical with the active part of the functor under consideration. This is enforced by the rule of functional application which states that a functor X/Y (where X and Y range over categories) may combine with an argument only if it is of category Y. But in UCG, the information contained in a sign is partial (not all attributes need have a defined value) and we have seen in section 2.1 that a natural way of combining two partial signs consists in unifying them. So rather than requiring that the active sign (Sign) of Sign1 equates with Sign2, we require that these two signs unify. In terms of substitution, this means that we must find the most general unifier \( \sigma \) such that \([\text{Sign}]_\sigma = [\text{Sign2}]_\sigma\). If such substitution cannot be found, functional application fails. Now, an important fact about the UCG rule of FA is that the substitution \( \sigma \) applies not only to Sign and Sign2 but in fact to the whole of the functor sign Sign1. Following [Klein 1988], I will refer to this particular aspect of UCG version of functional application as the instantiation of the functor. Functor instantiation is defined as follows.

**Definition 2.2 Functior instantiation**

Given a functor sign Sign1 with active sign Sign, the instantiation of Sign1 with respect to some other sign Sign2 is \([\text{Sign1}]_\sigma\) where \( \sigma \) is such that \([\text{Sign}]_\sigma = [\text{Sign2}]_\sigma\).

Let us now come back to the second question: What is the result of forward applying Sign1 to Sign2? Again we take inspiration from traditional CG. There, the result is stated in the functor category: a functor category of the form X/Y combines with a Y to yield an X. The UCG definition of the result category is similar but reflects the fact that unification, rather than equality, is used to check the appropriateness of Sign2 as an argument. The definition of the result category is given by the following definition for functional application:
Definition 2.3 Functional Application

Given a functor sign Sign1 of the form Pho:(Synt/Sign):Sem:Order and some adjacent sign Sign2, the result of combining Sign1 with Sign2 by functional application is \([\text{Pho}' : \text{Synt} : \text{Sem} : \text{Order}]_o\) where \([\text{Sign1}]_o\) is the instantiation of Sign1 with respect to Sign2.

Note that in this definition, the result category contains a phonology \(\text{Pho}'\) which is left undefined.

I now give a more intuitive presentation of the UCG rule of functional application which remedy this imprecision. There are in fact two UCG rules of functional application: a forward one and a backward one. The forward rule of functional application can be represented as in (2.28) where \(\text{PhoA}\) is the value of the phonology attribute of the argument sign, SignA and the order attribute of SignA is \(\text{pre}\). The backward version is similar except that the argument (i.e., SignA) must precede (rather than follow) the functor and its order value must be \(\text{post}\) rather than \(\text{pre}\).

\[
\begin{align*}
\text{pho} & : \text{PhoF} \\
\text{synt} & : \text{SyntF}/\text{SignA} \\
\text{sem} & : \text{SemF} \\
\text{order} & : \text{OrderF}
\end{align*}
\]

(2.28)

To illustrate the functioning of this rule, I will now run through a simple example. Consider the VP \(\text{likes Jon}\) with signs for \(\text{likes}\) and \(\text{Jon}\) (recall that NP abbreviates a sign with syntactic category np):

\[
\begin{align*}
\text{pho} & : \text{likes} \\
\text{synt} & : \text{n}/\text{NP/} \\
\text{sem} & : \text{SemA} \\
\text{order} & : \text{pre}
\end{align*}
\]

(2.29)
Unifying the active sign of the sign for *likes* with the sign for *Jon* yields the following substitution of terms for variables:

\[ \sigma = \{(\text{PhoA, } Jon), (\text{SemA, } jon), (\text{OrderA, } pre)\} \]

By Definition 2.3, the result category is thus:

\[
\begin{align*}
\text{pho} : \text{likes } + \text{ jon} \\
\text{synt} : s/\text{NP} \\
\text{sem} : \text{like}(X, \text{ jon}) \\
\text{order} : \text{Order?}
\end{align*}
\]

(2.30)

where Sem has been instantiated to *jon* and the phonology is *Jon+walks* i.e., *Jon walks*.

It has long been observed ([Ades/Steedman 1982 and Moortgat 1988]) that functional application alone does not permit an extensive coverage of natural language. In particular, it fails to account for unbounded dependencies. As my purpose here is to give the reader a flavour of UCG rather than a full account of it, I will not go into the details of how the basic version of UCG presented so far can be extended to deal with unbounded dependencies\(^4\). The basic idea is inspired from Gazdar’s slash category: it consists in augmenting the value of the synt attribute with a new attribute called slash in which information about extracted constituent is stored. A unary rule is then introduced which maps ‘standard’ constituents into constituents whose active sign has been transferred from the synt attribute to their slash attribute (for instance, the rule can apply to a transitive verb to yield an intransitive verb with a slash attribute whose value contains the information initially stored in the active sign — i.e., the object NP — of the transitive verb to which the rule has applied). This caters for what [Gazdar et al 1985] refers to as slash binding i.e., for the licensing of unsaturated constituents (i.e., constituents missing some argument). Slash termination (i.e., the process of licensing a sequence consisting of an extracted constituent followed by an unsaturated sentence) is then enforced either at the lexical level (i.e., in the categories of wh-words) or by a binary rule which states that a constituent followed by a sentence whose slash value unifies with this constituent yields a sentence.

Notational conventions: throughout the thesis, I will use the notations indicated in (2.31) to denote UCG binary and unary rules.

\(^4\)See [Zeevat et al 1987] for a comprehensive overview of the UCG framework.
2.2.3 The lexicon

Like many modern linguistic frameworks (e.g., GPSG, HPSG, CUG), UCG is a lexically oriented theory of language in that it uses very few combinatory rules operating on fairly complex information structures. It is therefore important that the lexicon be organised in such a way that redundancies can be avoided and generalizations captured. To this effect, two devices are used: templates and lexical rules. This follows the general practice adopted in the development of unification-based grammar frameworks, a practice which can be traced back to the creation of the PATR-formalism (see [Shieber 1986]).

A template serves to describe the structural properties shared by several lexical items; it provides a means for capturing linguistic generalizations holding over classes of words. For instance, a template \texttt{@transitive}\footnote{Following [Karttunen 1986], I will use the symbol \texttt{@} to designate templates. Thus a template name will be of the form \texttt{@template name}.} might be used to represent the general properties shared by all transitive verbs which is that they all subcategorise for two NPs, one of which is in the nominative case and must precede the verb and the other is in the accusative and must follow the verb. Intuitively, a template name abbreviates a feature structure indicating certain properties. In the case of \texttt{@transitive} for instance, the template name abbreviates a feature structure of the form:

\begin{equation}
\begin{bmatrix}
\text{pho} & : & \text{PhoF} \\
\text{synt} & : & \text{s} \\
\text{sem} & : & \text{SemA1} \\
\text{order} & : & \text{post}
\end{bmatrix}
\end{equation}

Formally, templates are defined by means of path equations. For instance, the definition for

\begin{equation}
\begin{bmatrix}
\text{pho} & : & \text{PhoA1} \\
\text{synt} & : & \text{np}[\text{nominative}] \\
\text{sem} & : & \text{SemA1} \\
\text{order} & : & \text{post}
\end{bmatrix}
\end{equation}

\begin{equation}
\begin{bmatrix}
\text{pho} & : & \text{PhoA2} \\
\text{synt} & : & \text{np}[\text{accusative}] \\
\text{sem} & : & \text{SemA2} \\
\text{order} & : & \text{pre}
\end{bmatrix}
\end{equation}

\begin{equation}
\begin{bmatrix}
\text{sem} & : & \text{arglist} = [\text{SemA1}, \text{SemA2}] \\
\text{order} & : & \text{OrderF}
\end{bmatrix}
\end{equation}

\begin{equation}
\begin{bmatrix}
\text{pho} & : & \text{PhoA1} \\
\text{synt} & : & \text{np}[\text{nominative}] \\
\text{sem} & : & \text{SemA1} \\
\text{order} & : & \text{post}
\end{bmatrix}
\end{equation}
@transitive is:

(2.33) \(\text{@transitive:}\)
\[
\text{synt:res:cat} = \text{a},
\]
\[
\text{synt:active:res:synt:cat} = \text{np},
\]
\[
\text{synt:active:res:feats:case} = \text{[nom]},
\]
\[
\text{synt:active:res:order} = \text{post},
\]
\[
\text{synt:active:res:sem} = \text{X},
\]
\[
\text{synt:active:active:synt:cat} = \text{np},
\]
\[
\text{synt:active:active:feats:case} = \text{[acc]},
\]
\[
\text{synt:active:active:order} = \text{pre},
\]
\[
\text{synt:active:active:sem} = \text{Y},
\]
\[
\text{synt:sem:arglist} = \text{[X,Y]}.
\]

But templates can also be defined in terms of other templates. For instance, suppose that we have a first template \(\text{@verb}\) whose definition is simply:

(2.34) \(\text{@verb:}\)
\[
\text{synt:res:cat} = \text{a},
\]
\[
\text{synt:active:res:synt:cat} = \text{np},
\]
\[
\text{synt:active:res:feats:case} = \text{[nom]},
\]
\[
\text{synt:active:res:order} = \text{post}.
\]

Then we can modify the definition of \(\text{@transitive}\) as:

(2.35) \(\text{@transitive:}\)
\[
\text{@verb},
\]
\[
\text{synt:active:active:synt:cat} = \text{np},
\]
\[
\text{synt:active:active:feats:case} = \text{[acc]},
\]
\[
\text{synt:active:active:order} = \text{pre},
\]
\[
\text{synt:active:active:sem} = \text{Y},
\]
\[
\text{synt:active:res:sem} = \text{X},
\]
\[
\text{synt:sem:arglist} = \text{[X,Y]}.
\]

That is, templates form a hierarchy. This leads to an inheritance-based organisation of the
lexicon where a lexical entry or template will inherit the information contained in the templates that are superordinate to it. Thus, suppose that *likes* is defined in terms of the transitive template *@transitive* which itself is defined in terms of *@verb*, then the feature structure for *likes* will inherit the information associated with both *@verb* and *@transitive*.

*Lexical rules* provide another way of structuring the lexicon. A lexical rules has the following format:

(2.36) \[ \text{RuleName: } FS_1 \Rightarrow FS_2 \]

where \( FS_1 \) and \( FS_2 \) are variables ranging over feature structures. In what follows, I will refer to the feature structure occurring before the arrow as the *input category* and to the one occurring after, as the *output category*.

Lexical rules play a role similar to transformational rules in transformational grammar, lexical redundancy rules in LFG and meta-rules in GPSG: they are used to capture relationships between linguistic expressions. A typical example is that of the relationship between passive and active. The various types of rule differ as to what objects they apply to: transformational rules apply to structural descriptions of clauses, GPSG meta-rules apply to rules and LFG lexical redundancy rules apply to lexical items. As the name suggests, UCG lexical rules also apply to lexical items. For example, the rule which transform an active verb into a passive one is stated as shown below.

(2.37) \[ \text{Active-to-Passive:} \]

\[
\begin{align*}
\text{pho} & : \text{PhoF} \\
\text{synt} & : s/ \\
\text{sem} & : \text{SemA}_1 \\
\text{order} & : \text{post} \\
\text{sem} & : \text{like(SemA}_1,\text{SemA}_2) \\
\text{order} & : \text{OrderF} \\
\text{pho} & : \text{PhoA}_1 \\
\text{synt} & : \text{np}\text{[nom]} \\
\text{sem} & : \text{SemA}_1 \\
\text{order} & : \text{post} \\
\text{sem} & : \text{like(SemA}_1,\text{SemA}_2) \\
\text{order} & : \text{pre} \\
\text{pho} & : \text{PhoA}_2 \\
\text{synt} & : \text{np}\text{[acc]} \\
\text{sem} & : \text{SemA}_2 \\
\text{order} & : \text{pre} \\
\end{align*}
\]
2.3 The semantic representation language

Traditionally formal semantics has focused on determining the meaning of isolated sentences rather than of discourse. However as we shall see, both gapping and VP ellipsis may involve the semantics of several sentences. Furthermore, we shall also see that the interaction of VP ellipsis with pronominalisation requires a treatment of intersentential pronominal anaphors as these influence the permissibility of VP ellipsis. Thus to provide an adequate semantics for verbal and VP ellipsis, it is best to adopt a framework which can account for intersentential anaphors.

A first attempt to combine model theoretic semantics with discourse theory is developed in [Kamp 1981] where a treatment of intersentential anaphora (and donkey sentences) is proposed. However this approach provides a semantics for complete sentences as opposed to the more traditional rule-by-rule semantics usual from Montague’s PTQ (cf. [Montague 19??]) and his followers. With regard to computation, this is problematic because it does not fit well with the compositionality principle implicitly embodied in unification- and in particular in sign-based grammars. Under this type of approach, the semantics of a sentence stems from the semantics of the lexical items of which the sentence consists: each items is associated in the lexicon with a sign containing some semantic information and as parsing proceeds, the grammar determines how the semantics contained in the sign of each constituent combine to yield the semantics of the resulting sentence. But in a system where a semantic representation can only be attached to sentences, such an incremental scheme is clearly inadequate.

Several proposals have been put forward to provide a rule-by-rule version of Discourse Representation Theory (DRT). One of them is the semantic representation language used in ucg i.e., InL which provides a rule-by-rule version of DRT enriched with sorted variables and a Davidsonian
treatment of tense. Another rule-by-rule semantics integrating the insights of DRT is Dynamic Montague Grammar (DMG) as developed in [Groenendijk/Stokhof 1990] and whose semantic representation language is Dynamic Intensional Logic (DIL). In this thesis, I chose to use a DMG like approach and to adopt DIL to encode the meaning of each constituent\(^6\). The main reason for this is that some work had already been done on VP ellipsis within the DRT framework (see [Klein 1987]) and that it seems worthwhile to compare DRT to DIL with respect to a specific problem. As it turns out the comparison reveals some interesting differences which though they could be spotted independent of any application had not been made clear in the existing literature at the time (cf. chapter 5 of this thesis and [Gardent 1990]). Thus the version of UCG used in this thesis has a semantics based on DIL rather than InL. This new version of UCG, I refer to as UCG(2).

The aim of this section is to introduce the semantic representation language of the grammar i.e., DIL. In section 2.3.1, I discuss the use of unification versus predication as a way of building up the semantics of a constituent in a compositional fashion and opt for predication. Section 2.3.2 starts by giving an informal explanation of the ideas underlying DIL. This is followed by a formal definition of the syntax and the semantics of DIL. In section 2.3.3, I discuss the DIL notion of meaning and show how it reflects the idea that meaning is a dynamic notion. Section 2.3.4 lists some notational conventions used to assign natural language expressions, a DIL translation that is easy to read. Finally in section 2.3.5, an example derivation is given which illustrates the workings of DIL.

2.3.1 Predication versus Unification

In a rule-by-rule semantics, the sub-constituents of any given constituent may be thought of as providing some partial information about the semantics of the constituent as a whole. Within a unification-based grammar, there are two distinct ways of representing this partial information. A first possibility is to follow Montague and have objects in the semantic domain which are properties or relations. Such entities are partial in the sense that they require some argument(s) to yield some more fully specified semantic object (e.g., a proposition). For example, the semantics of runs can be taken to be a property, notated \(\lambda z.\, \text{run}(z)\), which requires one argument to yield a fully specified proposition. Another handle on partial information is given by feature structures themselves: any missing information may be signalled by some uninstantiated variable. Under

\(^6\)This is not quite accurate. In fact, as we shall see in chapter 4, the semantic representation language used here is a modified version of DIL, called DIL\(^2\). The differences between DIL and DIL\(^2\) are discussed in chapter 4.
this approach, the semantics of \textit{run} is simply \textit{run}(X) where \textit{X} is some uninstantiated variable.

These two ways of representing partial information couple with two distinct ways of combining partial information: predication and unification. Unification is as described in the preceding section; Typically, when unification is used, the semantics of a given constituent is used as a frame which will be further specified by the semantics of the other constituents. In UCG, the constituent which provides this frame is called the \textit{semantic functor} and the other constituents are called \textit{semantic arguments}. By contrast, \textit{predication} is a structure building operation which puts together a relation with its argument(s) to build a more fully specified entity e.g., a proposition. \textit{Functional application} is just a special case of predication.

To illustrate the difference between the two options, let us consider how the semantics of the sentence \textit{Jon runs} is built. When predication is used (and assuming that NPs are treated as generalised quantifiers), the semantics of the subject NP may simply be applied to the semantics of the verb. In this case, the semantics for \textit{Jon runs} will be the proposition \textit{AP}(\textit{jon'})(\lambda \textit{z}. \textit{run'}(\textit{z})) which, by the definition of \textit{\beta}-reduction familiar from the lambda-calculus is equivalent to \textit{run'}(\textit{jon'}). Alternatively, if unification is used, the verb will provide the frame for the semantics (i.e., it will be the semantic functor) and the semantics of \textit{Jon} might just be \textit{jon}. The grammar then somehow ensures that the uninstantiated variable \textit{X} in \textit{run}(\textit{X}) gets instantiated to \textit{jon} so that the resulting semantics is \textit{run(jon)}.

On the basis of such simple examples, it may seem that the differences between predication and unification are rather thin. However when considering larger grammar fragments, problems arise with unification which are not present in predication type approaches. In particular, it is difficult to come up with an elegant account of coordination (see [Pereira/Shieber 1987] for a detailed account of the problems encountered). Nevertheless, it might be argued that within a unification-based framework, unification is a more 'natural' operation than predication. But if functional dependencies are allowed, then predication also is a 'natural' operation. All in all, it is difficult to argue for or against the use of unification as a structure building operation for constructing the semantics compositionally. In what follows, I simply integrate DIL into UCG and use predication as a combining operation. This means that the semantic attribute of semantic functors will have a functionally dependent value: the semantics of a functor sign will be a function of the semantics of its argument signs (for instance, the semantics of \textit{Jon runs} will be \textit{apply}(\textit{AP}(\textit{jon'}), \lambda \textit{z}. \textit{run'}(\textit{z})) where \textit{apply} is a function which given two arguments \textit{A} and \textit{B} returns the functional application of \textit{A} to \textit{B}). Note though that it would not be difficult to provide a unification-based version of DIL. In fact, the definition of such a language could very
closely parallel the definition of the original semantic language of UCG i.e., InL.

Notational convention: throughout the thesis, the functionally dependent value apply(I, Y) will be abbreviated to I(Y).

2.3.2 Dynamic Intentional Logic

DRT as described in [Kamp 1981], does not provide a rule-by-rule semantics as required by the incremental character of UCG: typically, the discourse representation for a whole sentence must be built before interpretation may begin. In contrast, Dynamic Montague grammar (cf. [Groenendijk/Stokhof 1990]) does. DMG can be thought of as a cross between MG and DRT. From MG, it retains the strict version of compositionality induced by a rule-by-rule semantics. From DRT, it maintains Kamp's insights regarding donkey sentences. The semantic representation language of DMG, Dynamic Intensional Logic (DIL), can be thought of as standing between Dynamic Logic (DL) as used in computer science and Montague's Intensional Logic (IL). Like IL, DIL is a many-sorted higher-order type-theoretic language. The syntax is defined as usual on the basis of the propositional connectives, lambda abstraction, application and quantification. The type theory is as in IL: basic types includes e, s and t; complex types are of the form a → b or s → b where a, b are any types. With each basic type a a non-empty domain Da is associated. To each complex type a → b the domain Da → Db is assigned consisting of all functions from Da to Db. The main differences between DIL and IL follows from the introduction into the language of discourse markers:

A Discourse Marker (DM) is an expression of type e and its meaning is an object in Dₜₑ, i.e., a function from states to individuals. The denotation of a discourse marker is as follows:

\[ [d]^{M,s}_{t} = F(d)(s) \] for every discourse marker d

But what are states? Intuitively, states contains information about the bindings occurring between discourse markers and individuals. More abstractly, States in DIL correspond to possible worlds in IL: they allow for the construction of intensional objects. Note however that contrary to MG where intensions are functions ranging over world x time indices, in DMG intensions are functions over states. Note also that in DMG, with the exception of DMs, all basic expressions are

---

7 For a more technical discussion of the non-compositional aspects of DRT the reader is referred to [Zeevat 1990].

8 For a more detailed presentation of DMG and DIL, see [Groenendijk/Stokhof 1990].
assigned extensional interpretations. That is, DIL is only intensional with respect to discourse markers. This is in contrast with Montague Grammar in which all expressions denote intensional objects. Formally, states are non-structured entities which are part of the model.

A model for DIL is a triple \((D, S, F)\) where \(D\) is a non-empty set of individuals, \(S\) is a set of states and \(F\) is the interpretation function. Although states are construed as primitives, the following postulates means that in effect they function as assignments of values to discourse markers.

Postulate 1 (Distinctness): for all \(d \in DM\) : \(F(d)(s) = F(d)(s')\), then \(s = s'\).

Postulate 2 (Update): For all \(s \in S, d \in DM, d \in D\), there exists an \(s' \in S\) such that \(F(d)(s') = d\) and for all \(d' \in DM\) if \(d' \neq d\) then \(F(d')(s) = F(d')(s')\).

That is, for any state \(s\) there exists another (unique) state \(s'\), notated \(s[d]s'\) such that \(s'\) is just like \(s\) except for the value that it assigns to \(d\). This means that states functions just like assignments of values to variables except that they provide values for discourse markers rather than for variables.

States are updated by state-switchers of the form \(\{ \alpha/d \}\), where \(\alpha\) is an expression of type \(e\) and \(d\) is a discourse marker. As their name indicate, state-switchers create a new state with respect to which the current expression will be evaluated. Their interpretation is given as follows:

\[
[(\alpha/d)\beta]^{M,s,t} = [\beta]^{M,(d \leftarrow \alpha)^{M,s,t}}
\]

where \((d \leftarrow \alpha)^{M,s,t}\) indicates a state \(s'\) just like \(s\) except that it assigns to \(d\), the value \([\alpha]^{M,s,t}\).

Given these preliminaries, we can now define DIL. The definition includes first the definition of types, then that of the syntax and finally the semantics.

Definition 1 (Types)
\(T\) is the set of types. \(e \in T. \ t \in T\). If \(a, b \in T\), then \(a \rightarrow b \in T\). If \(a \in T\), then \(s \rightarrow a \in T\).

Definition 2 (Syntax)
1. If $\alpha \in CON_a \cup VAR_a$, then $\alpha \in ME_a$

2. If $\alpha \in DM$, then $\alpha \in ME_e$

3. If $\alpha \in ME_a \rightarrow b$ and $\beta \in ME_a$, then $\alpha(\beta) \in ME_b$

4. If $\phi, \psi \in ME_t$, then $\neg\phi, [\phi \land \psi], [\phi \lor \psi], [\phi \rightarrow \psi] \in ME_t$

5. If $\phi \in ME_t$ and $v \in VAR_a$, then $\exists v\phi, \forall v\phi \in ME_t$

6. If $\alpha \in ME_a, v \in VAR_a$ then $\lambda v\alpha \in ME_{b-a}$

7. If $d \in DM, \alpha \in ME_a, \beta \in ME_a$, then $\{\alpha/d\}\beta \in ME_a$

8. If $\alpha \in ME_a$, then $^\sim \alpha \in ME_{a-a}$

9. If $\alpha \in ME_{a-a}$, then $^\sim \alpha \in ME_a$

10. Nothing else is in $ME_a$ for any type $a \in T$

**Semantics**

A model for DIL is a 3-tuple $(D, S, F)$ where $D$ is a non-empty set of individuals, $S$ is a set of states and $F$ is the interpretation function. Postulates 1 and 2 constrain the behavior of states as indicated above. The set of possible denotations of type $a$ is defined as follows.

**Definition 3 (Domains of interpretation)**

$D_e = D$, $D_a \rightarrow b = D_b^{p*}$, $D_{a \rightarrow a} = D_b^S$ and $D_t = \{0, 1\}$.

Note that the interpretation function $F$ does not always assign to expressions of $ME_a$ a denotation in $D_a$. Exceptions to the rule are DMs. If $\alpha \in DM$ then $F(\alpha) \in D^S$. For all other expressions, if $\alpha \in ME_a$ then $F(\alpha) \in D_a$. The semantics of DIL recursively defines for any expression $\alpha$ of DIL the interpretation of $\alpha$ denoted $[\alpha]^*$ ($M$ and $g$ are ignored for simplicity) as follows.

**Definition 4 (Semantics)**

1. $[c]^* = F(c)$, for every constant $c$.

2. $[d]^* = F(d)(e)$, for every discourse marker $d$.

3. $[\alpha(\beta)]^* = [\alpha]^*[\beta]^*$
4. \([-\phi]^* = 1\) iff \([\phi]^* = 0\) 
   = 0 otherwise
5. \([\phi \land \psi]^* = 1\) iff \([\phi]^* = [\psi]^* = 1\) 
   = 0 otherwise
6. \([\phi \lor \psi]^* = 1\) iff \([\phi]^* = 1\) or \([\psi]^* = 1\) 
   = 0 otherwise
7. \([\phi \rightarrow \psi]^* = 0\) iff \([\phi]^* = 1\) and \([\psi]^* = 0\) 
   = 1 otherwise
8. \([\exists \phi]^* \cdot s = 1\) iff there is a \(d \in D_\alpha\) such that \([\phi]^* . s[\alpha/d] = 1\), where \(\alpha\) is the type of \(v\).
9. \([\forall \phi]^* \cdot s = 1\) iff for all \(d \in D_\alpha\) it holds that \([\phi]^* . s[\alpha/d] = 1\), where \(\alpha\) is the type of \(v\).
10. \([\lambda \alpha]^* = \text{that function } h \in D_\alpha^S \text{ such that } h(d) = [\alpha]^*[\alpha/d] \text{ for all } d \in D_\alpha \text{ where } \alpha \text{ is the type of } \alpha, \text{ and } b \text{ the type of } v\). 
11. \([\{\alpha/d\} \beta]^* = [\beta]^* s' = (d \leftarrow [\alpha]^* )s\)
12. \('[\alpha]^* = \text{that function } h \in D_\alpha^S \text{ such that } h(s') = [\alpha]^* s' \text{ for all } s' \in S\)
13. \('[\alpha]^* = [\alpha]^*(s)\)

2.3.3 Dynamic meaning

Before going on to provide a DIL semantics for natural language expressions, I want to briefly consider the DMG notion of meaning. Like DRT, DMG is a dynamic theory of meaning, that is, one in which the meaning of a sentence is seen to reside in its information change potential rather than in its truth conditions. The general idea here is that language is used to convey information and that as discourse proceeds, the information states of discourse participants change according to the content of the utterances produced.

This very vague description of the concept of information update finds several instantiations in the work of semanticists. Here the definition of information state is a crucial one. For example, in a possible world semantics, one can think of an information state as a set of worlds: the proposition expressed by a sentence is then the set of all worlds with respect to which this sentence is true. Furthermore, each utterance restricts this set. For example, suppose that we start with an information state containing two worlds: one in which it is true that Jon drinks...
and one in which it is false. Suppose further that someone utters the sentence *Jon drinks* then, I can eliminate from my information state, the world in which *Jon drinks* is false.

In DMG, the notion of information state (and information update) is exclusively concerned with the notion of anaphoric potential. Information states contain information about the possible referents of pronouns. This information is modelled by introducing into the model a set of discourse markers and a set of states which functions as mapping from discourse markers into individuals. An information state then can be seen as a set of states. For example, suppose we have the following model:

\[
\begin{align*}
F(\text{man}) &= \{x\} & D &= \{x, y\} \\
F(\text{walk}) &= \{x, y\} & DM &= \{d_1\} \\
S_x : \quad &d_1 \rightarrow x \\
S_y : \quad &d_1 \rightarrow y
\end{align*}
\]

where \( D \) is the set of individuals, \( DM \) is the set of discourse markers and \( S \), the set of states. Then the proposition expressed by the sentence *He walks* is the set of states \( \{S_x, S_y\} \) with respect to which *He walks* is true. In contrast, the proposition expressed by *A man walks* is the singleton set \( \{S_x\} \). So in general, we can think of the proposition expressed by a sentence as the set of states with respect to which this sentence is true. In effect, this set of states reflects the anaphoric potential of the sentence: it contains all the bindings (between discourse markers and individuals) which make this sentence true.

To encode the information change potential of a sentence, the extension of sentences is defined in terms of valid continuations. Thus a sentence with truth-condition \( \phi \) is in fact assigned the DIL translation:

\[
\lambda p.[\phi \land \neg p]
\]

where \( p \) is a variable of type \( s \rightarrow t \) ranging over propositions. The idea here is that the translation of a sentence abstracts over the translation of the next sentence, and that sentence sequencing is modelled by functional application of the translation of the first sentence to the translation of the second sentence. So in DMG, the extension of a sentence is of type \((s \rightarrow t) \rightarrow t\). That is, it is a set of set of states or put another way, a generalised quantifier over states. Here a comparison with the semantics of generalised quantifiers over individuals will help. Consider the quantifier *a man* and its predicate logic translation (ignoring intensionality for simplicity):

\[
\lambda P \exists x[\text{man}(x) \land P(x)]
\]
Intuitively, the extension of this quantifier is the set of all properties $P$ of type $e \rightarrow t$ such that $P$ is true of the individual denoted by $x$ and $\text{man}(x)$ is also true. Applying a similar reasoning to the DMG translation

$$\lambda p[\phi \wedge \neg p]$$

of sentences, we have then that the extension of a sentence is the set of all propositions $p$ of type $s \rightarrow t$ such that $p$ is true with respect to a given state $s$ and $\phi$ is also true in $s$. In essence this means that a sentence extension consists of the set of set of states which other propositions may denote and which are consistent with the anaphoric information contained in $\phi$. For example, consider the following model:

$$D = \{z, y\}$$
$$DM = \{d_1\}$$
$$S = \{M_x, W_y\}$$
$$F(\text{whistle}) = \{y\}$$
$$F(\text{walk}) = F(\text{man}) = F(\text{talks}) = \{z\}$$
$$M_x : d_1 \rightarrow z$$
$$W_y : d_1 \rightarrow y$$

The following sentences express the propositions indicated to their right:

$$\text{A man}_1 \text{ walks } \{M_x\}$$
$$\text{He}_1 \text{ talks } \{M_x\}$$
$$\text{He}_1 \text{ whistles } \{W_y\}$$
$$\text{He}_1 \text{ is a man } \{M_x, W_y\}$$

Suppose the discourse starts with $\text{A man}_1 \text{ walks}$. A valid continuation must preserve the anaphoric information contained in this first sentence i.e., it must keep track of the fact the the discourse marker $d_1$ is bound to the individual denoted by $x$. Thus, $\text{He}_1 \text{ talks}$ is a valid continuation but $\text{He}_1 \text{ whistles}$ is not (because $W_y$ is not consistent with $M_x$). More generally, any valid continuation of $\text{A man}_1 \text{ walks}$ must have as information state a set of states which contains $M_x$. So the extension of $\text{A man}_1 \text{ walks}$ will be the set $\{\{M_x\}, \{M_x, W_y\}\}$.

The meaning of a sentence is simply obtained by abstracting over the current state. So if the extension of a sentence is $\lambda p[\phi \wedge \neg p]$, its meaning is $^*(\lambda p[\phi \wedge \neg p])$ with type $s \rightarrow ((s \rightarrow t) \rightarrow t)$. 


2.3.4 Abbreviations

This section summarises the set of abbreviations introduced by Groenendijk and Stokhof to facilitate the representation of the translations assigned in DMG to natural language expressions (For a more detailed discussions of these abbreviations, see [Groenendijk/Stokhof 1990:16]).

\[ A1 \ \models \phi = \lambda p[\phi \land \neg p] \]
\[ A2 \ \models \phi = \phi(\text{true}) \]
\[ A3 \ \neg \phi = \models \neg \phi \]
\[ A4 \ \phi; \psi = \lambda p[\phi(\neg(\psi(p)))] \]
\[ A5 \ E\phi = \lambda p \exists x(x/d)(\phi(p)) \]
\[ A6 \ \phi \Rightarrow \psi = \models [\phi; \neg \psi] \]
\[ A7 \ \phi \text{ or } \psi = \models [\neg \phi; \neg \psi] \]
\[ A8 \ Ad\phi = \models E\phi \neg \phi \]

The following equivalences are also useful:

\[ E1 \ \models \phi = \phi \]
\[ E2 \ E\phi; \psi = E\phi; \psi \]
\[ E3 \ E\phi \Rightarrow \psi = Ad[\phi \Rightarrow \psi] \]
\[ E4 \ \models \neg \phi = \neg \phi \]
\[ E5 \ \models [\phi; \models \psi] = \models [\phi \land \psi] \]
\[ E6 \ \models [\phi \Rightarrow \models \psi] = \models [\phi \Rightarrow \psi] \]
\[ E7 \ \models [\phi \text{ or } \models \psi] = \models [\phi \lor \psi] \]

2.3.5 Example derivation

To illustrate the workings of the semantic component, I now run through the semantic derivation of the discourse \( A_1 \) man walks. \( Hc_1 \) talks. First, I list the translations associated in the lexicon with the most basic words of English using the abbreviations defined in the preceding section.

1. man \( \mapsto \lambda x. \models \text{man}(x) \)
2. walk \( \mapsto \lambda x. \models \text{walk}(x) \)
CHAPTER 2. LINGUISTIC FRAMEWORK

3. love \(\mapsto \lambda P \lambda x. [\neg P(\lambda y \uparrow \text{love}(y)(x))]\)

4. \(a_i \mapsto \lambda P \lambda Q \lambda d_i [\neg P(d_i); \neg Q(d_i)]\)

5. every \(\mapsto \lambda P \lambda Q \lambda d_i [\neg P(d_i) \Rightarrow \neg Q(d_i)]\)

6. no \(\mapsto \lambda P \lambda Q \lambda d_i [\neg P(d_i) \Rightarrow \neg \neg Q(d_i)]\)

7. he \(\mapsto \lambda Q[\neg Q(d_i)]\)

8. Jon \(\mapsto \lambda P(j/d_i) P(\neg d_i)\)

Given these definitions, the semantic derivation for \(A_1 \text{ man } \text{walks}\). He talks can be summarised as follows:

1. \(a_i \mapsto \lambda P \lambda Q \lambda d_i [\neg P(d_i); \neg Q(d_i)]\)  
   Basic expression
2. man \(\mapsto \lambda x. \uparrow \text{man}(x)\)  
   Basic expression
3. \(a_i \text{ man } \mapsto \lambda P \lambda Q \lambda d_i [\neg P(d_i); \neg Q(d_i)](\lambda x. \uparrow \text{man}(x))\)  
   FA
4. \(a_i \text{ man } \mapsto \lambda Q \lambda d_i [\uparrow \text{man}(d_i); \neg Q(d_i)]\)  
   \(\lambda\)-conversion, \(\neg\) cancellation
5. walks \(\mapsto \lambda x. \uparrow \text{walk}(x)\)  
   Basic expression
6. \(a_i \text{ man } \text{walks} \mapsto E d_i [\uparrow \text{man}(d_i); \uparrow \text{walk}(d_i)]\)  
   FA, \(\lambda\)-conversion, \(\neg\) cancellation
7. He \(\mapsto \lambda Q[\neg Q(d_i)]\)  
   Basic expression
8. talks \(\mapsto \lambda x. \uparrow \text{talk}(x)\)  
   Basic expression
9. He \(\text{ talks } \mapsto \uparrow \text{talk}(d_i)\)  
   FA, \(\lambda\)-conversion, \(\neg\) cancellation
10. \(A_1 \text{ man } \text{walks}. \text{ He } \text{ talks } \mapsto E d_i [\uparrow \text{man}(d_i); \uparrow \text{walk}(d_i); \uparrow \text{talk}(d_i)]\)  
    Sentence sequencing
11. \(E d_i [\uparrow \text{man}(d_i); \uparrow \text{walk}(d_i); \uparrow \text{talk}(d_i)]\)  
    E2

To obtain the semantics of \(A_1 \text{ man } \text{walks}. \text{ He } \text{ talks}\), we then translate \(E d_i [\uparrow \text{man}(d_i); \uparrow \text{walk}(d_i); \uparrow \text{talk}(d_i)]\) back into its non-abbreviated form using the equivalences defined above.

\[
E d_i [\uparrow \text{man}(d_i); \uparrow \text{walk}(d_i); \uparrow \text{talk}(d_i)] =
E d_i [\uparrow \text{man}(d_i) \land \text{walk}(d_i) \land \text{talk}(d_i)] \quad \text{(By E5)}
\lambda p \exists (z/d_i) [\uparrow \text{man}(d_i) \land \text{walk}(d_i) \land \text{talk}(d_i)](\neg p) \quad \text{(By A5)}
\lambda p \exists (z/d_i) [\lambda q \text{man}(d_i) \land \text{walk}(d_i) \land \text{talk}(d_i) \land \neg q](\neg p) \quad \text{(By A1)}
\lambda p \exists (z/d_i) \text{man}(d_i) \land \text{walk}(d_i) \land \text{talk}(d_i) \land \neg p \quad \text{(By } \lambda\)-conversion)
This concludes the presentation of DIL. In chapter 4 and 5, it is shown how DIL can be extended to deal with VP and verbal ellipsis.

2.4 Summary

In this chapter, I have introduced the general framework within which the analysis of VP and verbal ellipsis proposed in this thesis is to be developed. This framework, called UCg(2), falls within the more general paradigm of unification-based grammar formalisms on the one hand and categorial grammars on the other. First, I gave some motivations for using a unification-based grammar formalism. These can be summarised as follows. First, it allows for a monostratal approach, that is one in which several dimensions of linguistic information may be used in parallel. Second, it yields a declarative, monotonic characterisation of linguistic knowledge — this means that the same grammar can be used both for parsing and for generation. Third, it provides a precise, computer-interpretable characterisation of natural language. I then go on to show in more detail what the expressive power of unification-based grammar formalisms is and discuss some more recent extensions that have been proposed in the literature (cf. section 2.1). The next step in the chapter consists in giving a precise description of UCg(2). This is carried out in section 2.2 where the lexicon and the grammar rules are discussed and an example derivation is given which illustrates the workings of UCg(2). Finally, section 2.3 presents the semantic representation language used (i.e., DIL). Here the particulars of a dynamic semantics are discussed, the precise syntax and semantics of DIL is presented and again, an example derivation is given which illustrates how the semantics of natural language expressions combine to yield a compound semantics reflecting the meaning of the compound expression.

This sets up the frame for the next chapters of this thesis. In the next chapter (chapter 3), the syntax of ellipsis is discussed and a UCg(2) analysis is proposed which concentrates on the syntactic aspects. The next two chapters (chapter 4 and 5) then go on to discuss the semantics of ellipsis and proposes a DIL-based analysis for it. Although these two aspects of ellipsis (syntax and semantics) are treated separately, it is to be understood that in effect they operate in tandem within the UCg(2) framework just described. In chapter 6, I show that discourse structure plays an important role in constraining VPE resolution and give a sketch of how discourse theory could be made to account for these constraints. Chapter 7 discusses the implementation and runs through an example meant to illustrate how the different sources of linguistic knowledge interact in determining the acceptability and the interpretation of gapping and VP ellipsis.
Chapter 8 concludes the thesis by discussing various shortcomings of the proposed analysis and pointing to directions for further research.
Chapter 3

Syntax

In describing a language, a grammar usually makes some assumptions about the phrasal structure of the constituents of this language. One common assumption is that a sentence results from recursively combining the categories associated with the constituting words of this sentence into a more complex constituent with category s. In a categorial grammar such as ucg(2), these two assumptions are made precise first, by positing that constituents are either functors or arguments and that they combine by functional application and second, by taking the finite verb to be the principal functor of the clause. This means that a string forms a grammatical sentence iff it contains a finite verb and the functor/argument requirements of this verb are satisfied by the constituents associated with the rest of the string.

From this perspective then, ellipsis is syntactically problematic in that it is characterised by an incomplete syntax or to put it in categorial terms: the chain of functors and arguments which normally constitutes a sentence is 'broken' by the absence of a functor (the verb) in the case of gapping and of an argument (the VP) in the case of VP ellipsis\(^1\). As noted in chapter 1, this raises two problems. First, the overt constituents contained in the e-clause must be licensed. In categorial grammar, constituents are usually licensed by their relation to some other constituent(s). For instance, a verb is licensed iff its subcategorisation requirements are fulfilled and vice-versa, an NP is licensed iff it satisfies the subcategorisation requirements of some functor. But as we have seen, ellipsis is characterised by the lack of either a functor or an argument. Thus some device other than the rule of functional application must ensure that the

\(^1\)Here I rely on the generally accepted assumption that auxiliaries are functors of the category VP/VP.
overt constituents in the e-clause are syntactically licensed. Second, the elliptical construct itself must be licensed: if the functor/argument chain which leads to the construction of a sentence category is broken, how then is the e-clause to be assigned a category? And further, how is this category to be licensed? The object of this chapter is to answer these two questions. The outline is as follows. Gapping is discussed first, then VPE. For both phenomena, the syntactic properties of the elliptical clause are presented first. An analysis is then developed within UCG(2) which is tested against the data. Finally, the proposed analysis is compared with existing approaches.

First, I introduce some terminology. Consider the example in (3.1).

(3.1) Jeremy fed a crocodile and Sarah a monkey.

Following [Chao 1987], I use the term a-clause to denote the antecedent clause (here Jeremy fed a crocodile) and e-clause to refer to the elliptical clause (Sarah did too). The term e-constituents will designate the main overt constituents in the e-clause and a-correspondents the constituents in the a-clause which are parallel to the e-constituents. Finally, the term a-verb will be used to refer to the root verb in the a-clause. Thus in the gapped clause above, Sarah and a monkey are the e-constituents, Jeremy and a crocodile are the a-correspondents and fed is the a-verb.

3.1 Gapping

3.1.1 Syntactic properties of Gapping.

Gapping has been extensively studied in the linguistic literature. This section summarizes the set of constraints which have been observed to operate on the syntax of gapped clauses.

The missing head restriction

A primary characteristic of gapped clauses is that the verbal head is missing. Moreover whenever gapping affects a verbal or a sentential argument, any verbal element higher in the clause

\[^2\text{See, for example, [Ross 1970; Koster 1978; Kuno 1976; Hankamer 1973; Sag 1980; Neijt 1980; Chao 1987; Russell 1987 and Steedman 1988].}\]

\[^3\text{I take the verbal head to be the root verb.}\]
must also be missing (see [Fiengo 1974 and van Riemsdijk 1978]). Hence a gapped clause always lacks a root verb and possibly more. This is illustrated by the following examples.

(3.2)  

a. Jon will look at Mary and Bill kiss her.  
b. *Jon will look at Mary and Bill won't at her.  
c. Jon knows how to eat spaghetti and Peter macaroni  
d. Jon knows how to eat spaghetti and Peter how to cook macaroni  
e. *Jon knows how to eat spaghetti and Peter wonders macaroni  
f. This doctor said that I should eat salmon and that doctor tuna.  
g. *This doctor said that I should eat salmon and that doctor said tuna.

In (3.2a) the e-clause is missing a verbal head and is thus a well-formed gapped clause. By contrast, (3.2b) is ungrammatical because the e-clause does contain a verbal head i.e., won't. Example (3.2c) shows a case in which both the lower and the higher VP are lacking. Example (3.2d) shows that only the higher verb may be gapped whereas (3.2e) illustrates that if only the lower verb is gapped then the resulting sentence is ungrammatical. More generally gapping may affect any number of auxiliaries and complement verbs beginning with the tensed element and going rightward. So for instance, the examples in (3.3) are all grammatical cases of gapping but the ones in (3.4) are not.

(3.3)  
a. I want to try to begin to write a novel and Mary to try to begin to write a play.  
b. I want to try to begin to write a novel and Mary to begin to write a play.  
c. I want to try to begin to write a novel and Mary to write a play.  
d. I want to try to begin to write a novel and Mary a play.

(3.4)  
a. *I want to try to begin to write a novel and Mary wants a play.  
b. *I want to try to begin to write a novel and Mary wants to write.

Note that complement adjectives may also be gapped as illustrated in (3.5).

(3.5)  
a. Jon is proud of his car and Bill of this house.
b. Jon is easy to please and Bill to annoy.

The major constituent restriction

As noted in [Hankamer 1973], the remnants of the gapped clause must be major constituents. Roughly\(^4\), major constituents are constituents which could function as arguments or adjuncts of the a-verb. Consider the following sentences:

\[
(3.6) \quad \begin{align*}
\text{a. } & \text{*This bar has sold much wine to boring customers and that will real ale to eager.} \\
\text{b. } & \text{This bar has sold much wine to boring customers and that restaurant much ale to eager clients.}
\end{align*}
\]

\(3.6a\) is ungrammatical because the remnants are not major constituents (for instance, that could not function as a subject of the a-verb sold). By contrast \(3.6b\) is grammatical, because that restaurant, much ale and to eager clients could function as subject, object and indirect object of the a-verb and are thus major constituents. Note that while a major constituent is necessarily a maximal projection\(^5\), the converse is not true. This is illustrated by the examples in \(3.7\).

\[
(3.7) \quad \begin{align*}
\text{a. } & \text{*Jon spoke to Fred and Mark Peter.} \\
\text{b. } & \text{Jon spoke to Fred and Mark to Peter.} \\
\text{c. } & \text{*Kim likes Sue and Lee to Lucy.} \\
\text{d. } & \text{*Pat wanted to go to Bern and Carlo going to Rome.}
\end{align*}
\]

In \(3.7a\), the remnant Peter is a maximal projection (NP) but it could not be an argument of the a-verb spoke as it does not obey its subcategorisation requirements. Thus Peter is not a major constituent and \(3.7a\) is ungrammatical. By contrast, \(3.7b\) is well-formed because the PP to Peter could be the argument of the a-verb spoke. Examples \(3.7c-d\) further illustrate the fact that maximal projections need not be major constituents.

\(^4\)For a more precise definition, see section 3.1.2.

\(^5\)The term maximal projection was first introduced in the X-bar theory to designate the maximal constituent which had to be built to satisfy the subcategorisation requirements of the head of this constituent. It is more generally used to denote constituents such as S, NP, VP, PP, AP etc.
Note also that remnants in the gapped clause need not have the same category as their a-correspondents. Thus consider the following examples (taken from [Sag et al 1984]).

(3.8)  
a. Kim is rather foolish and Lou a complete idiot.  
b. Leslie seems to be just surviving and Terry in dire need of help.  
c. We consider Leslie rather foolish and Lou a complete idiot

In (3.8a) and (3.8c), a remnant with category NP has an a-correspondent of category AP. In (3.8b), the correspondence is between a PP and a participial complement. More generally, [Sag et al 1984] note that a- and e-correspondents are constituents which may always be conjoined to form the argument of a particular verb. For instance in (3.8a), *rather foolish and a complete idiot* is a coordinate structure which can satisfy the subcategorisation requirements of be. That is, the category divergence between remnants and a-correspondents is allowed in exactly the same range of cases as occurs in coordinate structures. This suggests that a prerequisite for a theory of gapping is that it be able to handle constituent coordination.

The examples below further illustrate the usefulness of the major constituent restriction.

(3.9)  
a. *Jon spoke to the visitor from France and Mark, from Belgium.  
b. First Kim’s picture of Sandy astonished Fido and then Lee’s, Felix.  
c. Kim and Sandy astonished Fido and Lee, Felix.

(3.9a) is similar to (3.7a), except that this time the remnant and its a-correspondent are of the same category: PP. However, the a-verb subcategorises for a very specific type of PP that is, one which is introduced by the preposition to. Hence the PP from Belgium does not satisfy the subcategorisation requirements of the a-verb and consequently cannot be a major constituent so that the gapped clause is syntactically ill formed and the resulting discourse ungrammatical. (3.9b) illustrates a case where the two remnants are NPs while the a-clause contains 3 NPs. Without the major constituent restriction, the two NPs of the e-clause could be in correspondence with any of the three NPs in the a-clause. However, with respect to the a-verb astonished there can only be two major constituents: the subject NP and the object NP. Thus there is no ambiguity as to the way in which e-clause and a-clause correspond: Lee’s corresponds to the subject NP and Felix to the object. A similar reasoning holds regarding (3.9c) in which Lee can only be the subject of the missing verb and Felix the object.
Gapping within arguments and within adjuncts

Complement verbs can only be gapped if they occur within an argument (cf. Pesetsky 1982 and Chao 1987]). When they occur within an adjunct, gapping renders the sentence agrammatical. This is illustrated in (3.10) below.

(3.10) a. *Jon left without telling his boss and Bill his colleagues.
   
   b. This doctor said that I should eat salmon and that doctor tuna.

Discontinuous remnants

The missing material in the e-clause can consist of several discontinuous constituents. Consider the examples in (3.11) below (here and in in some other places in the thesis, I use $\emptyset$ to indicate the position of the missing constituent and $[\emptyset = X]$ to indicate the resolvent).

(3.11) a. The man gave chocolate to Mary and the boy $\emptyset_1$ flowers $\emptyset_2$.
   
   $[\emptyset_1 = \text{gave}; \emptyset_2 = \text{to Mary}]$

   b. Max wanted Ted to persuade Alex to get lost and Walt $\emptyset_1$ Ira $\emptyset_2$.
   
   $[\emptyset_1 = \text{wanted}; \emptyset_2 = \text{to persuade Alex to get lost}]$

   c. Max wanted Ted to persuade Alex to see Mary and Walt $\emptyset_1$ Ira $\emptyset_2$.
   
   $[\emptyset_1 = \text{wanted}; \emptyset_2 = \text{to persuade Alex to see Mary}]$

   d. Massachusetts elected John Harris congressman, and Pennsylvania $\emptyset_1$ Bill Thomas $\emptyset_2$.
   
   $[\emptyset_1 = \text{elected}; \emptyset_2 = \text{congressman}]$

   e. Harry believes Jon to be good and Peter $\emptyset_1$ Paul $\emptyset_2$.
   
   $[\emptyset_1 = \text{believes}; \emptyset_2 = \text{to be good}]$

   f. Jon took Harry to the movies and Bill $\emptyset_1$ Mike $\emptyset_2$.
   
   $[\emptyset_1 = \text{took}; \emptyset_2 = \text{to the movies}]$

   g. Jon eats a cake with a fork and Bill $\emptyset_1$ a pie $\emptyset_2$.
   
   $[\emptyset_1 = \text{eats}; \emptyset_2 = \text{with a fork}]$

   h. Betsy talked to Alan on Sunday and Peter $\emptyset_1$ to Sandy $\emptyset_2$.
   
   $[\emptyset_1 = \text{talked}; \emptyset_2 = \text{on Sunday}]$
Examples (3.11a-f) illustrate how several non-adjacent constituents may be missing from the e-clause where the first missing constituent is the verb and the second is an argument. Examples (3.11g-h) are similar except that the second gap is an adjunct rather than an argument. In such cases, the discourse is ambiguous between a reading where the adjunct in the a-clause also modifies the e-clause and one where it does not. For instance, (3.11h) means either Betsy talked to Alan on Sunday and Peter talked to Sandy on Sunday or Betsy talked to Alan on Sunday and Peter talked to Sandy. Note that there are also several possible readings for example (3.11c) which have not been indicated here (for instance, (3.11c) may also be represented as Max wanted Ted to persuade Alex to see Mary and \( \theta_1 \) Walt \( \theta_2 \) Ira with resolvents \([\theta_1 = \text{Max wanted}; \theta_2 = \text{to persuade Alex to see}]\). These are discussed further in section 3.1.3.

Syntactic relationship between a- and e-clause

Although gapping is usually restricted to coordinate structures as in (3.12a), it may also occur across speakers in discourse as in (3.12b).

(3.12) a. Jon likes Mary and Bill Sarah.
    b. A: I shall bring the salad.
       B: And I the wine.

Backward gapping does not occur in English (cf. 3.13).

(3.13) *Bill, Sarah and Jon likes Mary.

The e-clause must be root (cf 3.14).

(3.14) a. *Jon persuaded Fred that Bill Sam.
    b. *Jon liked movies long before Bill concerts.
    c. *Jon likes movies and everybody knows that Bill concerts.

This section concludes the discussion of the properties of gapped clauses. The next section shows how an analysis of gapping can be developed which accounts for these properties.
3.1.2 Syntactic analysis of gapping

Any account of the syntax of gapping should answer the following questions. First, do the remnants of a gapped clause form a constituent and if so, what is the category of this constituent? Second, how is the incomplete syntax of a gapped clause to be licensed? Third, how does the resulting analysis extend to related phenomena such as stripping or conjunction reduction? The object of this section is to answer each of these questions.

How does gapping relate to other elliptical constructs

Cases of elliptical constructs which bear a strong similarity to gapping include stripping and conjunction reduction (as we shall see in chapter 8, sluicing, null-complement anaphora and VPE have very different linguistic properties and need not be considered here). The analysis I propose generalises to stripping but not to conjunction reduction. The difference between gapping and stripping is illustrated by the following examples:

(3.15)

a. Jon likes Mary and Peter Sarah. (*Gapping*)

b. Jon likes Mary and Peter too. (*Stripping*)

Briefly, stripping is characterised by the fact that the elliptical clause contains only one NP and possibly some adverbials and a discourse particle. There is a clear sense in which the elliptical constructions illustrated by the examples in (3.15) are related: they both lack a verb and possibly other constituents and the missing material is contained in a clause that is coordinated to the e-clause. Thus the question immediately comes to mind of whether stripping and gapping ought to be handled by some unifying ellipsis principle. In the linguistic literature, they have often been treated as two separate phenomena. More recently, however, analyses have been developed which convincingly argue that stripping and gapping share a cluster of common properties ([Chao 1987; Russell 1987 and Sag et al 1984]). In particular stripping like gapping obey the missing head and the major constituent constraint as illustrated by the following examples.

(3.16)

a. Jon likes Sarah and Peter too.

b. *Jon likes Sarah and Peter likes too.
c. This bar has sold much wine to boring customers and that restaurant also.
d. *This bar has sold much wine to boring customers and that also.

On the basis of these properties, I opt here for an analysis of gapping which naturally extends to stripping. Now consider the sentence in (3.17) which illustrates a case of conjunction reduction.

(3.17) Jon gave a rose to Sarah and a tulip to Sylvette.

At first sight, the second conjunct in this sentence seems to belong to the same paradigm as stripping and gapping: it can also be seen as missing a verb which occurs in the preceding conjunct. The question of whether this is in fact the case is a moot point. Roughly, three main positions are possible. According to transformational grammar, (3.17) is derived from Jon gave a rose to Sarah and Jon gave a tulip to Sylvette. But this analysis is unsatisfactory in that it fails to generalise to very similar sentences such as:

(3.18) A man gave a rose to Sarah and a tulip to Sylvette.

The argument is a semantic one: if transformations are to be meaning preserving, then it cannot be the case that A man gave a rose to Sarah and a tulip to Sylvette is derived from A man gave a rose to Sarah and a man gave a tulip to Sylvette because the two sentences do not have the same meaning. An alternative is to view sentences such as (3.17) as a case of gapping involving VPs rather than sentences. Under such a view, (3.17) is then thought of as meaning Jon gave a rose to Sarah and gave a tulip to Sylvette, which is semantically correct. Such an approach is developed in [Sag et al 1984 and Russell 1987]. The problem with such an analysis is that it implies either that a rose to Sarah and a tulip to Sylvette cannot coordinate (because otherwise the coordination rule would apply rather than whatever device is used to deal with gapping) or that the sentence in (3.17) is structurally ambiguous. The second implication is simply odd, whilst the first one is denied by sentences as in (3.19) where similar constituent sequences do conjoin (i.e, Sarah a rose and Sue a tulip in (3.19a) and Mary on Tuesday and Paul on Wednesday in (3.19b)).

(3.19) a. Jon gave either Sarah a rose or Sue a tulip.
b. Jon talked to Mary on Tuesday and Paul on Wednesday.

To summarise, the most plausible analysis of sentences such as (3.17) is one which takes sequences such as Sarah a rose to be admissible conjuncts. Of course this implies departing from most traditional phrase-structure based account of syntax, but theories nonetheless exist which allow for such an analysis. In particular, the Lambek calculus ([Moortgat 1988]) provides a framework in which such a departure is possible. I will not investigate the matter further here and simply assume that sentences such as (3.17) are best seen as cases of constituent coordination and consequently need not be taken into account by a treatment of gapping.

What is the category of a gapped clause?

Some evidence that gapped clauses are constituents is that they can conjoin. For instance, in (3.20) the gapped clause the yellow cable to terminal B conjoins with a second gapped clause the black cable to terminal C to yield a coordinate construct.

(3.20) The blue cable must be connected to terminal A and either the yellow cable to terminal B or the black cable to terminal C.

But if gapped clauses are constituents in their own right, what is their category? Their surface constituent structure gives no indication as to what this category should be because typically, gapped clauses lack a head and traditionally, the head of a constituent is understood as determining in some way the category of this constituent. In categorial terms, this means that there is no main functor from which the category of the gapped clause could be derived.

7For instance in GB, the head of a constituent has the same category as the constituent itself but a different bar level (e.g., N is the head of N'). Similarly in GPSG, the head features of the head daughter are shared by the mother category. Since head features include categorial information, this means that the categorial information carried by the daughter is the same as that carried by the mother. Note though that the notion of head is theory dependent. For instance, in GB the verb is not the head of a clause (whereas it is in OG). So in this case, the lack of verb in a gapped clause means that an important constituent is missing in the clause which is part of the complement of the head (rather than the head itself). A more general way to state that the phrasal structure of gapped clauses provides no information about their category would then be to say that gapped clauses are syntactically incomplete in that they do not satisfy the standard functor/argument requirements of a fully fledged constituent so that the combination of the parts does not result in the forming of any given constituent. As this is somewhat long-winded, and as OG does take the verb to be the principal functor of a sentence, I will continue to refer to the lack of verb in gapped clauses as lack of head.
But consider now the following syntactic properties of gapped clauses. First, note that gapped clauses may conjoin with sentences. Second, gapped clauses can be modified by sentential adverbials as in *Jon will probably go to see Mary and necessarily Paul, Sarah*. Third, gapped clauses can be subject to right node raising as in *Jon likes to be known to and Peter to be unknown to, the officers on site or in First Jack gave Peter, then Jon and later Paul a farewell present*. These three properties strongly suggest that gapped clauses are constituents of category S. However note that the distribution of gapped clauses is a lot more restricted than that of normal sentences. As noted in section 3.1.1, although gapped clauses can be coordinated, they cannot be subordinated. Furthermore as already noticed, they markedly differ from normal sentences in that they contain no verbal head. So perhaps the right conclusion is that gapped clauses form a particular class of sentences. We then need to define what this class is. There are three possibilities here. A first possibility is to allow empty categories into the framework, and to let gapped clauses be sentences containing empty verbs. Such an approach is adopted in [Russell 1987]. A second possibility is to have a grammar such that the category of gapped clauses expresses the fact that gapped clauses lack one or more constituent(s). This approach is pursued within a categorial framework in [Steedman 1989 and Wood 1988] and within GB in [Chao 1987]. In the categorial frameworks, the category of a gapped clause is $S\backslash V$ where $V$ may be any type of verb (transitive, intransitive etc.) whereas in the GB approach, it is $S^-$ that is, a special type of sentence which may lack any number of constituents\(^8\). There is yet a third alternative, which is that the category of gapped clauses is simply $X^2\ast$, where $X^2$ is a variable over categories of bar level two and $\ast$ is the kleene star indicating the presence of one or more objects of the type it suffixes. That is, the category of a gapped clause is a variable category ranging over any sequence of one or more categories of bar level two. Such an approach is defended in [Sag et al 1984].

Which of these categories (a sentence containing an empty verb, a sentence missing a verb or a variable ranging over sequences of bar-level two categories) is the most appropriate for gapped clauses essentially depends on the linguistic framework being used and on its general assumptions, as different categories will require the introduction in the grammar of different mechanisms which will be more or less in tune with the general philosophy embodied in this particular grammar. As the more general question of how gapped clauses are to be licensed by the grammar is discussed in the next section, I defer to then the task of deciding on the precise category of gapped clauses. For the moment, suffice it to say that I will take gapped clauses to be special sorts of sentences.

---

\(^8\)The details of each of these approaches are discussed in section 3.1.4.
How are gapped clauses licensed?

The syntactic licensing of gapped clauses involves two main sub-tasks. First, the category of the gapped clause itself must be licensed. Second, the various constraints on gapping described in section 3.1.1 must be accounted for.

Consider the first task first. The problem is to determine how gapped clauses are to be assigned a category by the grammar. As already mentioned in the preceding section, three possibilities are open as to the resulting category: it can be a sentence containing an empty verb, a sentence missing a verb or a variable ranging over sequences of bar-level two categories. In what follows, I review how different existing approaches allow for those various possibilities and discuss their pros and cons with respect to their compatibility with UCG(2). Having opted for an approach where a gapped clause has category $X^2$ (that is, an approach of the third type) I then go on to show how this category may be added to the set of UCG(2) categories.

In [Steedman 1989], gapped clauses are categorised as $S\backslash V$. To allow for such a category assignment the grammar needs to be extended to ensure that the category $S\backslash V$ can be derived from any sequence of remnant categories (for instance, the sequence (NP, NP)). In [Steedman 1989], this is achieved by augmenting the grammar with additional combinatory rules. There are several problems with this type of approach. From the linguistic point of view, the proliferation of rules makes it very hard to predict which combinations will be allowed and which will be ruled out. In short, it often leads to overgeneration. From the computational point of view, the uncontrolled application of the many rules induces the computation of spurious parses that is, parses which are derivationally distinct but semantically equivalent. The proliferation of rules in a categorial grammar can lead to such an increase in the search space that the parsing problem becomes computationally intractable. Although some solutions to this problem have been developed for categorial systems such as Steedman’s Combinatory Categorial Grammar ([Hepple/Morrill 1989]) and the Lambek calculus ([Koenig 1989]), the problem of how these results apply to feature-based categorial grammar remains unsolved. So it seems best to avoid the proliferation of categorial rules in UCG(2). Finally, from the empirical point of view, the problem is that the category $S\backslash V$ inadequately characterises gapped clauses because in some cases a gapped clause will be missing not only a verbal head, but also an NP or a PP (e.g, Jon put the wine on the table and Peter, the bread).

We have seen that adding rules to the grammar is not a viable option. [Wood 1988] proposes an alternative: she introduces in the grammar an operation which she calls reconstitution. Briefly,
given two adjacent constituents bearing the category $X/Y$ and $Z/V$ (where $Y \neq Z$), this operation permits the reconstitution of a constituent of category $Y/Z$. Apart from the fact that the status and functioning of this operation within the grammar remains rather unclear, this approach suffers from the same empirical shortcoming as Steedman’s in that it does not account for discontinuous missing material in the gapped clause.

Yet another alternative is described in [Chao 1988] where the syntax of gapped clauses is licensed by a *defective* $X^-$ schema, and a global constraint on surface structures dubbed $H^-$ projection. Although as we shall see in section 3.1.4, the spirit of this analysis is not so dissimilar from mine (in particular, both approaches take gapped clauses to be special sorts of sentences), the particulars of Chao’s analysis are not fit to be integrated into a categorial framework because they crucially relies on concepts such as the X-bar schema which are absent from UCG(2).

A fourth possibility is to adopt an approach of the type developed in [Sag et al 1984]. This approach assigns to the gapped clause a category which simply describes what is visible and makes no hypothesis about what this category should or ought to be. As it manages to avoid the problems mentioned before and thus succeeds in adequately characterising the syntax of gapping by making fewer assumptions, I will adopt it. That is, I take the category of a gapped clause to be a variable over one or more categories of bar level two. This category, I represent with $X^2$. This of course, raises the question of how such categories come to be derived by the grammar. Here I will follow [Lambek 1958] and, more recently [Wood 1989], and assume that a categorial calculus should also allow for the formation of *product categories*. Intuitively, a product is just a category of type $X^*$ i.e., a sequence of categories which functions as a single (complex) category: it can be the argument of a functor category, a conjunct in coordinate constructions etc. To allow for the formation of product categories within UCG(2), I augment the set of rules described in chapter 2, section 2.2.2 with the rule in (3.21), which I will call the *product rule*.

---

9Such an analysis may seem to ignore the suggestion made above that gapped clauses are a special sort of sentences (unlike e.g., $S\text{\text{\text{\_\text{\text{\_inf}}}}}$ or $S\text{\text{\text{\_\text{\text{\_comp}}}}}$, the category $X^2$ bears no obvious relation to the category $S$). However, we will see in the next section that the overall analysis of gapping which I propose is in effect based on the idea that gapped clauses are related to constituents of category $S$.

10By introducing this new rule into the grammar, we run into problems similar to those raised when discussing Steedman’s approach: this additional rule might lead to overgeneration and computational inadequacy. However, as we shall see later on, the approach I propose does avoid the empirical shortcomings present in Steedman’s approach.
But not every sequence of maximal projections is a well-formed gapped clause. To capture the restrictions on gapping observed in section 3.1.1, the grammar must include some mechanism by which a sequence of maximal projections is or is not licensed as a gapped clause. There are basically two dimensions along which an analysis of how gapped clauses are to be licensed may vary. First, there is the origin of the licensing: is the gapped clause licensed by syntax, by semantics etc. or perhaps by a combination of several information sources? In general\(^{11}\), licensing always include syntax and semantics because on the one hand, some restrictions on gapping are clearly syntactic (cf. the major constituent restriction) and on the other hand, gapping is in essence an anaphoric phenomena which requires a semantic antecedent. But as illustrated in [Steedman 1989], it may also involve pragmatics (cf. section 3.1.4). In the analysis I propose, only syntax and semantics are taken into account. A sketch of how additional sources of information could be make to bear on the analysis of gapping is given in chapter 8.

A second dimension of variation for treatments of gapping concerns the way in which the licensing material (be it a syntactic, a semantic or a multi-dimensional object) is made available. Here there are two main possibilities: the licensing material can either be recovered (cf. [Chao 1987]) or it can be predicted (cf. [Steedman 1989 and Russell 1987]). The difference between the two types of approach can be explained as follows. When dealing with elliptical or anaphoric constructs, we have to decide which information needs to be passed on to the next sentence or conjunct. We can either try to make a guess as to what will be needed (this is prediction) or extract from the a-clause the information needed to license the e-clause (this is recovery). Here I chose to adopt a recovery type approach. The main reason for this is that it guarantees the appropriate degree of parallelism\(^{12}\) between a- and e-clause. This point is best illustrated by some examples. Consider the sentences in (3.22).

\[(3.22)\] a. Peter talked to his boss on Tuesday and Jon on Wednesday.

b. Jon believes Paul to be good and Peter David.

\(^{11}\)An exception is [Russell 1987] where gapping is licensed at the level of semantics only. As we shall see in section 3.1.4, this raises some problems from the syntactic point of view.

\(^{12}\)I use the term parallelism here intuitively. Its precise meaning should become clear shortly.
CHAPTER 3. SYNTAX

Here, the intended interpretation for the elliptical clauses is Jon talked to his boss on Wednesday and Peter believes David to be good respectively. Suppose we use a prediction approach. Typically, predictive approaches predict continuous constituents. For instance, in (3.22a) the following constituents might be predicted: \([v \: \text{talked}]\), \([vp \: \text{talked to his boss}]\) and \([vp \: \text{talked to his boss on Tuesday}]\) from which the following interpretations will be generated: Jon talked on Wednesday, Jon talked to his boss on Wednesday and Jon talked to his boss on Tuesday on Wednesday. Thus, the grammar might overgenerate. Conversely, it might also undergenerate. For, if only continuous constituents are predicted, then the e-clause in (3.22b) might either fail to be interpreted, or be wrongly interpreted as meaning Peter believes David. By appropriately comparing the constituents present in the e-clause with the global structure associated with the a-clause, these problems can be avoided. In fact, the recovery approach I propose is based on a substitution operation which in effect ensures that these problems do not arise.

Given these observations about the general characteristics of the present analysis, let us now consider in more detail how gapping is licensed by the grammar. The general idea is to augment the grammar with a licensing principle which must hold of any gapped clause for this clause to be recognised as grammatical. This principle can be summarised as follows.

**Licensing principle:** a gapped clause is licensed by the grammar iff the sequence of categories forming its product category can be substituted into the derivation tree of the corresponding a-clause.

This raises the question of how substitution is defined. One way to think of gapping is to view the linguistic information associated with a gapped clause as providing a partial specification of some fully-fledged sentence. The intuition behind using substitution then is that by substituting the information associated with the gapping remnants into the derivation tree of the a-clause, a full specification of this sentence can be recovered. More specifically, it must be the case that (i) each of the categories associated with the gapping remnants can be substituted for some major constituent in the a-tree and (ii) the new tree resulting from this substitution is derivable under the present grammar. As we shall see in section 3.1.3, the substitution approach successfully captures the restrictions on gapping listed in section 3.1.1. Furthermore, it yields a natural account of gapped clauses as special sorts of sentences: from the partial specification provided by the gapping remnants and the background information stemming from the a-clause, substitution allows for the specification of a fully-fledged sentence to be derived. An example
should illustrate this. Consider the sentence in (3.23).

\[(3.23) \quad \text{Jon likes Mary and Paul Sarah.}\]

The substitution of the categories forming the product category $NP(Paul) \ast NP(Sarah)$\(^{13}\) into the derivation tree $(r_a)$ of the a-clause can be represented as in Figure 3.1 where the substitution operation of the list of categories $List$ formed by a product category into a tree $r$ is represented by substitute($r$, $List$) and $NP(Paul) \ast NP(Sarah)$ is represented as $(NP(Paul), NP(Sarah))$.

\[\text{substitute} \quad \begin{array}{c}
S, [NP(Paul), NP(Sarah)] \\
\end{array} = \quad \begin{array}{c}
S \\
\end{array} \]

\[\begin{array}{c}
\begin{array}{c}
NP \\
Jon \\
\end{array} \\
\end{array} \quad \begin{array}{c}
S/\text{NP} \\
\end{array} \\
\begin{array}{c}
(S/\text{NP})/\text{NP} \\
\text{likes} \\
\end{array} \\
\begin{array}{c}
\text{NP} \\
\end{array} \\
\begin{array}{c}
\text{NP} \\
\end{array} \\
\begin{array}{c}
\text{NP} \\
\end{array} \\
\begin{array}{c}
\text{NP} \\
\end{array} \\
\begin{array}{c}
P_a \\
Paul \\
\end{array} \\
\begin{array}{c}
S/\text{NP} \\
\end{array} \\
\begin{array}{c}
(S/\text{NP})/\text{NP} \\
\text{likes} \\
\end{array} \\
\begin{array}{c}
\text{NP} \\
\end{array} \\
\begin{array}{c}
\text{NP} \\
\end{array} \\
\begin{array}{c}
\text{NP} \\
\end{array} \\
\begin{array}{c}
\text{NP} \\
\end{array} \\
\end{array} \]

\[\begin{array}{c}
\begin{array}{c}
\text{NP} \\
\end{array} \\
\begin{array}{c}
\text{NP} \\
\end{array} \\
\begin{array}{c}
\text{NP} \\
\end{array} \\
\begin{array}{c}
\text{NP} \\
\end{array} \\
\end{array} \]

Figure 3.1: Example of substitution.

I now proceed to give a formal definition of substitution. This involves first defining the notion of a derivation tree within $UCG(2)$, second, defining the notion of major constituency and third, defining substitution itself. I defer to the next sections the discussion of the merits and theoretical status of the proposed analysis.

A derivation tree is a recursive specification of the derivation of a given constituent. Special provision needs to be made for gapped clauses: the tree of a coordinated gapped clause results from substituting the gapping remnants into the derivation tree of the a-clause where the a-clause is identified as being the first sentential constituent f-commanding the e-clause\(^{14}\). Formally, the definition runs as follows.

\(^{13}\)For simplicity, I use the abbreviation Cat(String) to designate a sign with syntax Cat and phonology String.

\(^{14}\)Given two nodes $X$ and $Y$ in a derivation tree, $X$ f-commands $Y$ if $X$ is a functor and either $X$ is a sister of $Y$ or $X$ is the sister of $Z$, $Z$ is the mother of $V$ and $V$ f-commands $Y$. 
Definition 3.1 (UCG(2) definition of a derivation tree)

Let $\sigma$ denote the substitution operation defined further below. Let $X, Y, Z$ be UCG(2) signs. For any sign $X$, let $\tau_X$ denote the derivation tree associated with $X$. Then for any sign derived under UCG(2), a tree $\tau$ is built which is defined as follows.

Unary rule. If a unary rule of the form $X \Rightarrow Y$ applies then $\tau = Y(\tau_X)$.

Binary rule (1). If a binary rule of the form $X, Y \Rightarrow Z$ applies and $Y$ is not a product category, then $\tau = Z(\tau_X, \tau_Y)$.

Binary rule (2). If a binary rule of the form $X, Y \Rightarrow Z$ applies where $X$ is the category associated with non-constituent coordination and $Y$ is a product category, then if $\tau_a$ is the derivation tree of the a-clause, and $\sigma(\{\tau_a, Y\}) = \{\tau_a, \{\}\}$, $\tau = Z(\tau_X, \tau_a)$.

If no tree can be built, the corresponding sentence is not grammatical under UCG(2).

Major constituency is then defined as follows. As we have seen, major constituents are potential arguments or adjuncts of the a-verb involved in the ellipsis. So for instance, in Jon loves Mary and Peter Sarah, the constituents associated with Peter and Sarah are major constituents because they could function as subject and object of the a-verb loves. However there are cases where a gapping remnant, rather than being a potential argument of the a-verb, is a potential argument of a verb contained in a sentential argument of the a-verb. So for instance in Jon knows how to eat spaghetti and Peter macaroni., the NP associated with macaroni could be the object NP of the verb eat contained within the sentential argument of the a-verb knows. Given these observations, I will take major constituents to fall into three cases. A major constituent can be the argument of a verb, the adjunct of a verb or the argument of a verb contained within the sentential argument of a verb. The formal definition of a major constituent is as follows:

Definition 3.2 (Major constituent)

$\alpha$ is a major constituent of a tree $\tau$ if $V$ is the verbal head of $\tau$ and

1. $\alpha$ is an argument of $V$ or
2. $\alpha$ is an adjunct of $V$ or
3. $\beta$ is a sentential argument of $V$ with tree $\tau_\beta$ and $\alpha$ is a major constituent of $\tau_\beta$. 
Note that this definition defines major constituents as actual arguments (or adjuncts) of a verbal head. This is in contrast with the notion of major constituency proposed so far, where major constituents were taken to be potential (rather than actual) arguments or adjuncts of the a-verb. The twist behind Definition 3.2 is that it defines major constituents to be constituents in the a-clause. Gapping remnants are then said to be well-formed major constituents iff they can be substituted for some major constituent in the a-tree. In this way, a notion of major constituency for gapping remnants is indirectly derived, which is in fact consistent with the informal explanations given above.

In the grammar, information about major constituency is modelled by adding to the set of attributes an attribute called *Major Constituency Domain* or MCD for short. For any given tree, the MCD value of the root sign will consist of the sequence of major constituents which occur in this tree. For instance, the MCD value of the mother sign of the derivation tree for *Jon likes Mary* will be the sequence (NP(Jon), NP(Mary)). The way in which the value of the MCD attribute is obtained straightforwardly reflects Definition 3.2. First, we stipulate that the MCD value of a verb unifies with the value of its subcategorisation list. That is, verbs in the lexicon will be assigned signs of the form:

\[
\begin{align*}
\text{pho} : & \text{Pho} \\
\text{synt} : & \begin{array}{c}
\text{cat} : s / [1] x \\
\text{mcd} : [1]
\end{array} \\
\text{sem} : & \text{Sem} \\
\text{order} : & \text{Order}
\end{align*}
\]

(3.24)

where, as explained in chapter 2, section 2.1, square boxes indicate reentrancy so that the value of the subcategorisation list is also that of the MC Domain. This caters for the first clause of Definition 3.2.

Second, we need to account for the fact that adjuncts are major constituents. In UCG(2), adjuncts are catered for by introducing a unary rule which makes intransitive verbs subcategorise for an adjunct. This unary rule can be schematically represented as follows (if and Adjunct abbreviate signs with category np and Adjunct respectively, where Adjunct can be any adjunct category such as pp or ap).
To account for the fact that adjuncts can be major constituents, this rule is modified as follows:

That is, if the input verb has MCD value \([1]\), then the same verb subcategorising for an adjunct \([2]\) has for MCD value \([1] + [2]\) where + simply indicates list concatenation.

Finally, we need to account for major constituents which are arguments contained within the sentential argument of a verb. This can be encoded in the lexicon by defining the MCD value of verbs which take a sentential argument as follows:
That is, the MCD value of any verb taking a sentential argument consists of the list concatenation of its subcategorisation list and of the MCD value of its sentential argument. The information stored in the lexicon is then percolated by the \textsc{ucg}(2) rules from the root verb to the root sign. The reason for this is that in \textsc{ucg}(2) the flow of information between signs within the derivation tree is regulated by an implicit head principle. For each rule application, the current functor is the head and all the features of this head are shared by the mother. For instance, consider the forward application rule:
Whenever this rule applies, all the feature values of the functor sign (including the MCD value) are also values of the result sign. Since the value of MC domains is specified on verbs and since verbs are functors, it follows that the MCD value of any root verb is percolated up the tree and combine with that of other verbs as specified in the lexical entries for verbs (and a few other constituents such as for instance, complementsers). This is best illustrated by an example. Consider the sentence *Jon claimed that he likes Sarah*, the percolation of the MCD values within the derivation tree can be represented as illustrated in Figure 3.2.

That is, the MC domain of the root sign consists of the sequence (NP(Jon), S(he likes Sarah), NP(he), NP(Sarah)).

The role of the MC domain is to constrain substitution: gapping remnants may only substitute for major constituents; that is, gapping remnants must be potential arguments of the a-verb.
Figure 3.2: Major Constituency Domain for Jon claimed that he likes Sarah.
CHAPTER 3. SYNTAX

Some examples will illustrate this. Consider the sentence in (3.29a) with major constituents the man and at the girl.

(3.29) a. The man looked at the girl
     b. and the dog at the bone.
     c. *and the dog the bone.

The gapping clause in (3.29b) is a possible continuation for (3.29a) because the dog and at the bone can substitute for the man and at the girl respectively and yield a sentence. In contrast, (3.29c) is not a permissible continuation because the girl is not a major constituent and so the bone may not be substituted for it.

Thus the grammar now allows for information about major constituency to be encoded in the derivation tree of any grammatical sentence. To reconstruct the derivation tree associated with a gapped clause and check that the constraints on remnants are obeyed, the substitution operation is now defined. The general idea underlying this definition is that substitution should recursively apply to the different sub-parts of the tree and for each sub-tree being considered, a different action should be taken depending on whether the mother node of this sub-tree is a major constituent.

Intuitively, substitution is similar to priority union on strings of feature structures. Consider for instance the sentence Jon likes Mary and Peter Sarah. We can think of the gapped clause as forming the string:

\[ x + NP(Peter) + y + NP(Sarah) + z \]

where \( x, y, z \) are variables ranging over signs, + denotes string concatenation and \( NP(Peter), NP(Sarah) \) abbreviates the feature structures associated with the NPs Peter and Sarah respectively.

To obtain a full specification of the e-clause, we then compare this string with the string of constituents formed by the a-clause by trying to unify the two strings: the unification of two strings succeeds if a substitution of values for variables can be found which makes the two strings equal where a variable can match 0 or more elements in a string.\(^\text{16}\) The examples in (3.30) show

\(^{15}\)The definition of substitution given below requires that any substituted constituent combine with its sister (if any) under the UCG rules of derivation. So here, the bone may not be substituted for at the girl because it fails to combine with its sister looked which subcategorises for a PP[at] rather than for an NP.

\(^{16}\)For a more formal definition of string unification, see [Siekman 1978].
string equations and the substitutions of values to variables which satisfy these equations (Ø denotes the empty string.).

<table>
<thead>
<tr>
<th>String equations</th>
<th>Substitutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a + b + c = x + c)</td>
<td>(a + b/x)</td>
</tr>
<tr>
<td>(a + b + c = x + y + z)</td>
<td>(a/x, b/y, c/z)</td>
</tr>
<tr>
<td>(\emptyset + a + z = u + v + z)</td>
<td>(\emptyset, a, u, v, z, z)</td>
</tr>
<tr>
<td>(a + b + a = x + y + z)</td>
<td>(a/x, b/y, a/z)</td>
</tr>
</tbody>
</table>

(3.30)

But consider the sentence in (3.31).

(3.31) Jon likes Mary and Peter Sarah.

In this case, the strings of constituents associated with a- and e-clause are:

\[
\emptyset + NP(Jon) + V(likes) + NP(Mary) + \emptyset \\
\emptyset + NP(Peter) + y + NP(Sarah) + z
\]

Clearly these two strings will not unify. Thus unification is inadequate for recovering a full specification of gapped clauses. Intuitively, what is required is an operation which will help recovering the missing material while keeping the information contained in the e-clause. Priority union is such an operation: it allows for the aggregation of incompatible information provided that one of the two incompatible structures is given precedence. If we use priority union on strings and give the gapped clause string precedence, then we get the intended result: the missing material is recovered and the syntactic parallelism between e- and a-clause is verified.

As an example, consider the sentence in (3.31). In this case, the priority union, written \(\Rightarrow\), of the string of constituents associated with the a-clause with the string of gapping remnants is:

\[
\emptyset + NP(Jon) + V(likes) + NP(Mary) + \emptyset \Rightarrow z + NP(Peter) + y + NP(Sarah) + z \\
= \emptyset + NP(Peter) + V(likes) + NP(Sarah) + \emptyset
\]
Just as required. So the general idea underlying the analysis of gapping proposed here is that the partial information associated with gapped clauses is enriched by a comparison with the linguistic information associated with the a-clause. This comparison can be roughly understood as involving the priority union of the strings of constituents associated with a- and e-clause.

However this abstract characterisation of gapping is inadequate in several respects. First, it says nothing about the nature of the elements contained in the strings to be compared (should they be maximal projections, lexical items, constituents etc.?) As we shall see, there are some important restrictions on what constitute a correct element for comparison (in particular, priority union should only affect major constituents). Second, it does not enforce another restriction on gapped clauses which is that the linguistic specification resulting from the comparison process must describe a sentence constituent - or to put it another way, the recovered structure must be such that a sentence can be derived from it by the grammar. In both cases, it is unclear how the very general operation of priority union can be tailored to obtain the correct results. In what follows, I thus develop an analysis based on a more specific operation called substitution and show how it can be made to interact with other grammar components to yield a proper account of gapping. The general idea underlying the analysis remains the same however.

Substitution is defined in three steps. First, I define the notion of a mother node for UCG(2) derivation trees. Second, I define an auxiliary function, called $\sigma$, which is used to recurse down the tree. Third, I define substitution proper. The definition for mother nodes runs as follows.

**Definition 3.3 (Mothernode)**

If $\tau$ is a tree then the mother node of $\tau$, notated $\mu(\tau)$, is $M$ if  

either $\tau$ is a binary tree of the form $M(\tau_1, \tau_2)$

or $\tau$ is a unary tree of the form $M(\tau_1)$

or $\tau$ is a leaf node labelled with the category $M$.

The auxiliary function $\sigma$ which is used to recurse down the tree is then defined thus.

**Definition 3.4 ($\sigma$)**

Let $PRODUCT_i$ denote product categories, $\tau_i$ denote UCG(2) derivation trees and $Mi$ denote UCG(2) signs. Then, $\sigma_i((\tau, PRODUCT))$ is defined as follows.
(1) If \( \tau \) is a binary tree of the form \( M(\tau_1, \tau_2) \)
Then \( \sigma_1((\tau, PRODUCT)) = (M_1(\tau_1, \tau_2), PRODUCT_3) \)
If \( \sigma((\tau_1, PRODUCT)) = (\tau_{11}, PRODUCT_2) \) and
\( \sigma((\tau_2, PRODUCT_2)) = (\tau_{22}, PRODUCT_3) \) and
\( \mu(\tau_{11}), \mu(\tau_{22}) \Rightarrow M_1 \)

(2) If \( \tau \) is a unary tree of the form \( M(\tau_1) \)
Then \( \sigma_1((\tau, PRODUCT)) = (M_1(\tau_1), PRODUCT_2) \)
If \( \sigma((\tau_1, PRODUCT_2)) = (\tau_{11}, PRODUCT_2) \) and
\( \mu(\tau_{11}) \Rightarrow M_1 \)

(3) If \( \tau \) is a leaf
Then If \( \mu(\tau) \) is a major constituent
Then Either (31) \( \sigma_1((\tau, PRODUCT)) = \sigma((\tau, PRODUCT)) \)
Or (32) \( \sigma_1((\tau, PRODUCT)) = (\tau, PRODUCT) \)
Else (33) \( \sigma_1((\tau, PRODUCT)) = (\tau, PRODUCT) \)

Finally, substitution is defined as follows.

Definition 3.5 (Substitution)
Let \( R_i \) denote gapping remnant categories and \( PRODUCT, PRODUCT_1 \) denote product categories. Let \( \tau, \tau_i \) denote derivation trees. For any node \( A_i \) contained in a tree \( \tau_i \) let \( A_i \) be a major constituent of \( \tau_i \) iff it appears in the MCD value of the mother node of \( \tau_i \). Let \( PRODUCT \) be a product category whose first element is \( R_1 \) and whose tail is \( PRODUCT_1 \) (i.e., \( PRODUCT = R_1 \ast PRODUCT_1 \)). \( \sigma((\tau, PRODUCT)) \) is defined as follows.

If \( \mu(\tau) \) is a major constituent
Then (1) Either \( \sigma((\tau, PRODUCT)) = (R_1, PRODUCT_1) \)
(2) Or \( \sigma((\tau, PRODUCT)) = \sigma_1((\tau, PRODUCT)) \)
Else (3) \( \sigma((\tau, PRODUCT)) = \sigma_1((\tau, PRODUCT)) \)

That is, there are three possible cases. If the mother node of the tree is not a major constituent, then substitution recursively applies to the different sub-parts of the tree via \( \sigma_1 \). If it is a major constituent, then there are two possibilities: either the first category in the list of remnant categories substitutes for this tree (for instance, in \( A \) man likes Sarah and a boy Mary, the remnant sign NP(a boy) can be substituted for the subtree associated with a man because the mother of this subtree is a major constituent) or substitution continues to apply within this tree.
Note that this means that substitution is non-deterministic and will sometimes yields more than one result (that is, substitution is a relation rather than a function). In this way, sentences such as *Jon introduced Peter to Paul and Jeremy to Sarah* can be accounted for, where the gapped clause can be interpreted either as meaning: *Jeremy introduced Peter to Sarah* or *Jon introduced Jeremy to Sarah*. An example should help illustrate the workings of $\sigma$. Consider the sentence in (3.32) with derivation tree for the a-clause as in Figure 3.3 where NP(Jon) and NP(Mary) are major constituents.

(3.32) Jon likes Mary and Paul Sarah.

To obtain the derivation tree $\tau_\sigma$ of the e-clause, we calculate:

$$\sigma((S(NP(Jon), IV(TV(\text{likes}), NP(Mary))), NP(Paul) \ast NP(Sarah)))$$

The calculation goes as follows. The first two clauses of Definition 3.5 cannot apply because $S$ is not a major constituent. So the third clause applies and since the input tree is a binary one, we have:

$$\sigma((S(NP(Jon), IV(TV(\text{likes}), NP(Mary))), NP(Paul) \ast NP(Sarah))) =$$

$$\sigma_1((S(NP(Jon), IV(TV(\text{likes}), NP(Mary))), NP(Paul) \ast NP(Sarah))) =$$

$$\langle S'(NP', IV'), P3 \rangle$$

if

$$\sigma((NP(Jon), NP(Paul) \ast NP(Sarah))) = \langle NP', P2 \rangle$$

and

$$\sigma((IV(TV(\text{likes}), NP(Sarah)), P2)) = \langle IV', P3 \rangle$$

and

$$\mu(NP'), \mu(IV') \Rightarrow S'$$
We now calculate the first condition. NP(Jon) is a major constituent, so clause (1) of \( \sigma \) can apply thus yielding:

\[
\sigma((NP(Jon), NP(Paul) \ast NP(Sarah))) = (NP(Paul), NP(Sarah))
\]

and NP' = NP(Paul) = \( \mu(NP') \) and P2 = NP(Sarah). By a similar reasoning, the second equation can be resolved as follows:

\[
\sigma((IV(TV(likes), NP(Mary)), NP(Sarah))) =
\]

\[
\sigma_1((IV(TV(likes), NP(Mary)), NP(Sarah))) =
\]

\[
(IV'(TV', NP''), P3') \text{ if}
\]

\[
\sigma((TV(likes), NP(Sarah))) = (TV', P2') \text{ and}
\]

\[
\sigma((NP(Mary), P2')) = (NP'', P3') \text{ and}
\]

\[
\mu(TV'), \mu(NP'') \Rightarrow IV'
\]

TV(likes) is a leaf which is not labelled with a major constituent so clause (3) of \( \sigma \) and (33) of \( \sigma_1 \) apply, yielding:

\[
\sigma((TV(likes), NP(Sarah))) = (TV(likes), NP(Sarah))
\]

so that:

\[
TV' = TV(likes) = \mu(TV') \text{ and } P2' = NP(Sarah)
\]

By contrast, NP(Mary) is a major constituent so clause (31) of \( \sigma_1 \) applies and we get:

\[
\sigma((NP(Mary), NP(Sarah))) = (NP(Sarah), \emptyset)
\]

so that:

\[
NP'' = NP(Sarah) = \mu(NP'') \text{ and } P3' = \emptyset
\]

Given the UCG(2) rules, we have that:

\[
TV(likes), NP(Sarah) \Rightarrow IV(likes \ Sarah)
\]

and further:

\[
NP(Paul), IV(likes \ Sarah) \Rightarrow S(Paul \ likes \ Sarah)
\]

And given the equalities we established, we finally have:

\[
\sigma((S(NP(Jon)), IV(TV(likes), NP(Mary)), NP(Paul) \ast NP(Sarah)))) =
\]

\[
(S(NP(Paul), IV(TV(likes), NP(Sarah)))), \emptyset
\]

So substitution is successful and \( \tau_e = S(NP(Paul), IV(TV(likes), NP(Sarah))) \)
Finally, we need to say something about the distribution of gapped clauses. As noticed in section 3.1.1, gapped clauses only appear in coordinated structures. I now turn to the question of how the present grammar generates coordinate structures involving gapping. Following [Wood 1988], I will assume that conjunctions such as and, or etc. are binary functors over constituents of identical categories (i.e., constituents of category \((X\backslash X)/X\) where \(X\) ranges over \(\text{UCG}(2)\) categories). To account for gapping, I introduce a new category for conjunctions:

\[
\begin{align*}
\text{pho} & : \text{Pho} \\
\text{cat} & : \text{sent} \\
\text{synt} & : \text{sent}/ \text{pho} \\
\text{mcd} & : \text{Mcd} \\
\text{sem} & : \text{Sem} \\
\text{order} & : \text{post} \\
\text{and} & (\text{Sem1}, \text{Sem2}) \\
\end{align*}
\]

(3.33)

where \(\text{Sem2}\) is the semantics of the mother node in the reconstructed tree of the e-clause. In what follows the category in (3.33) will be abbreviated to \((S\backslash S)/X^2\). Thus we get the following derivations\(^{17}\) (A CCG derivation is represented as follows. At the top is the string to be analysed with below each word, the associated category. Further below, each new line indicates a rule application. The scope of the rule is indicated by the length of the line whereas the name of the rule is indicated at the right of the line.> FA abbreviates forward functional application and <FA, backward functional application.):

\(^{17}\)Note that given that the grammar contains two types of conjunctions (i.e., \((X\backslash X)/X\) and \((S\backslash S)/X^2\)), spurious derivations become possible. For instance, consider a sentence of the form \(S & G & G\), where \(S\) abbreviates a sentence, & a conjunction and \(G\) a gapped clause. There are two possible ways of deriving this sentence: \((S & G) & G\) and \(S & (G & G)\). To obtain only one derivation, I use features to allow or block certain combinations. In essence, this feature system will favour derivations of the type \(S & G\) where \(S\) is a simple (rather than a coordinate) sentence and \(G\) is either a simple or a complex gapped clause.
To summarise, the basic idea underlying the analysis just described is that tree reconstruction is an essential ingredient of gapping analysis. There are two main reasons for this. First, reconstruction ensures that the sentence is interpretable. The question of how gapped clauses are assigned an interpretation by the grammar is discussed at length in chapter 4. Second and more relevantly, it serves to check the grammaticality of the gapped clause. In this section, I have given the details of how reconstruction is to proceed. In the next section, I show how reconstruction accounts for the syntactic restrictions on gapping observed in section 3.1.1.
CHAPTER 3. SYNTAX

3.1.3 Analysis and data

As observed in section 3.1.1, a first constraint on gapping is that gapping remnants must be major constituents. Three cases of major constituency violation were given: when some remnant was not a maximal projection, when it did not satisfy the subcategorisation requirement of the a-verb and when it was not the argument or the adjunct of a verbal head. I consider each case in turn.

The first case was illustrated by the ungrammaticality of the sentence *This bar has sold much wine to boring customers and that will real ale to eager where the list of remnants includes a determiner, an auxiliary, a preposition and an adjective none of which have the required bar level. This case is excluded in the present grammar by categorising gapped clauses as $X^2$.

The second case was illustrated by the contrast between (3.34a) and (3.34b).

(3.34) a. *Jon spoke to Fred and Mark Peter.
   b. Jon spoke to Fred and Mark to Peter.

Under the analysis developed above, the ungrammaticality of the first sentence is recognised during substitution. Substitution requires that the tree of the gapped clause be entirely reconstructed: whenever a remnant is substituted within the antecedent tree, it must be able to combine with its sister. In the examples under consideration, the first example is rejected as ungrammatical because the NP Peter does not satisfy the subcategorisation requirements of spoke and thus no tree can be reconstructed for the gapped clause. In the second example, the PP to Peter does satisfy these requirements and the sentence is accepted. Note that because categorial identity is not required to hold between the remnants and the major constituents they substitute for, sentences such as Kim is rather foolish, and Lou a complete idiot are unproblematic.

The third case concerns sentences such as Kim and Sandy astonished Fido and then Lee Felix where the second remnant cannot substitute for Sandy even though it is a major projection and could syntactically combine with Kim to form a coordinate constituent. In the above analysis, this restriction on gapping is due to the fact that Sandy is not a major constituent and thus cannot be substituted for.

A second constraint on the syntax of gapping is that all verbal heads must have been deleted.
So Jon will look at Mary and Bill kiss her is fine and so is Jon will look at Mary and Bill at Sarah but Jon wants to take a bus and Gloria intends a taxi is not. Under the present analysis the ungrammaticality of the last example follows from the fact that intends is not a maximal projection and thus is not a well-formed remnant. The first example is accepted because the MC domain of auxiliaries consists of the merging of their own subcategorisation list (i.e., (VP)) with that of their argument (i.e., the subcategorisation list of the VP it takes as argument).

This means that in Jon will look at Mary both look at Mary and at Mary are major constituents so that at Sarah is a well formed remnant because it can substitute for at Mary.

A third observation was that gapping remnants could be arguments and adjuncts in non nominal arguments of the main verb but not arguments and adjuncts in adjuncts of the main verb. Thus The child insisted that she wanted chips and the mother salad is fine but Jon left without telling his boss and Bill his colleagues is not. These restrictions fall out from the definition of major constituents: salad is a major constituent because it is the argument of a verb occurring within an argument of the main verb. In contrast, his colleagues isn't because it occurs within an adjunct of the main verb.

Another fact about gapping is that it can affect any number of complement verbs. Thus both I want to try to begin to write a novel and Mary to write a play and I want to try to begin to write a novel and Mary a play are grammatical. Again, under definition 3.3, this follows from the fact that both to write a novel and a novel are major constituents and thus can be substituted for by to write a play and a play respectively. The fact that intermediary verbs cannot occur in isolation in the gapped clause follows from the definition of remnants given by the gapping rule (isolated verbs are of bar level one whereas the gapping rule requires that the gapped clause consists of constituents with bar level two).

A fourth and final fact that had to be taken into account is that a gapped clause may contain multiple gaps. This is no particular problem for the present analysis since no restriction is made as to the position of the remnants with respect to the antecedent tree: any number of gaps may separate any number of remnants, the only constraint (besides major constituency) is that the resulting tree be accepted by the grammar.

Overgeneration and acceptability

As the astute reader will doubtless have noticed, the analysis described in 3.1.2 is rather permissive as to the number of readings it associates with gapped sentences. For instance consider
the sentence in (3.35).

(3.35) Jon persuaded Jeremy to examine Peter.

This sentence contains four major constituents: Jon, Jeremy, Peter and Jeremy to examine Peter

Consequently, under the present grammar all of the following continuations are acceptable.

(3.36) a. Jon persuaded Jeremy to examine Peter and Bill Tom.
    b. Bill [persuaded] Tom [to examine Peter].
    e. ?Jon persuaded Jeremy to examine Peter and Bill Bruno to examine Tom.

Intuitions about gapping are notoriously unstable and vary from speaker to speaker. However the general consensus about the data in (3.36) is that (3.36a) is fine, but that (3.36b-e) are hard if not impossible to understand under the reading given in italics below each line.

So the grammar overgenerates. I believe that this is a normal state of affairs and that the overall acceptability of a gapped clause is a complex factor involving several non-syntactical ones such as: lexical variation, parallelism, information structure and processing. In what follows, I consider each of these factors and show that examples can be found which clearly indicate that constraints on gapping interpretation should not be treated at the level of syntax alone but should result from the interaction of syntax with processing strategy, lexical information and information structure.

Let us look at lexical variations first. Consider the following examples.

(3.37) a. Jon promised Jeremy to donate $300 and Bill $400.
    Bill [promised Jeremy to donate] $400.

b. Jon promised Jeremy to donate $300 and Bill $400.
c. The child insisted that she wanted chips and the mother that she should have salad.

*The mother [insisted] that she should have salad.*

All of these examples are grammatical. Interestingly, they exhibit exactly the same patterns as the ungrammatical examples in (3.36). The only difference resides in the type of the a-verb. Exactly what the correlation is between lexical information and gapping is a matter for further research. Note though that this interaction of the lexicon with syntax can be accounted for particularly well in lexicalist frameworks such as UGC(2).

The following examples illustrate the effect of parallelism on gapping.

(3.38) a. Jon persuaded Dr. James to examine the old man and Bill the old woman.

*Bill [persuaded Dr. James to examine] the old man.*

b. Jon persuaded Dr. James to examine Sarah and Dr. Jones Mary.

*[Jon persuaded] Dr. Jones [to examine] Mary.*

Although these examples are hard to process, some speakers find them acceptable. Since they obey the same pattern as examples (3.36b-c) above (which were unacceptable), this suggests that semantic parallelism between correspondents influences the acceptability of gapped clauses.

The effect of information structure and processing on gapping was first brought to attention in [Kuno 1976]. Kuno argues first that the gapping remnants must represent new information and second, that there are independent processing factors which drastically affect gapping acceptability. The effect of information structure on gapping is illustrated by the examples in (3.39) below.

(3.39) a. Who persuaded Jon to see whom?

*Peter persuaded Jon to see Mary and Paul Sarah.*

*Paul [persuaded Jon to see] Sarah.*

b. Jon persuaded who to see whom?

*Jon persuaded Peter to see Mary and Paul Sarah.*

*[Jon persuaded] Paul [to see] Sarah.*

When provided with the appropriate context, the readings which were unavailable in (3.36b)
and (3.36c) become available. Under Kuno's theory, this is due to the fact that restrictions on the interpretation of gapped clauses are informational rather than syntactic: a gapped clause is acceptable under a certain reading if the remnants introduce new information. To account for the tendency of many speakers to prefer some interpretations over others, Kuno posits three perceptual factors: the minimal distance principle (MDC), the tendency for subject-predicate interpretation (TSPI) and the requirement for simplex-sentential relationship (RSSR). The MDC simply states that the remnants of the gapped clause can be most readily coupled with the a-constituents of the same category which were processed last. This accounts for instance for the fact that on hearing the sentence Jon gave Mary a book and Sarah a rose, the preferential interpretation is Jon gave Sarah a rose and not Sarah gave Mary a rose. The TSPI posits that when the remnants are NP and VP, there is a tendency to interpret these two remnants as forming a sentence such that the NP becomes the subject of the VP. Finally the RSSR stipulates that gapping remnants are most readily interpretable as entering into a simplex-sentential relationship. This explains why for instance, Jon persuades Thomas to examine Jane and Bill Martha is uninterpretable if Bill is to be the correspondent of Jon and Martha that of Jane. According to Kuno, the more a sentence violates these principles the less it will be acceptable: if it violates three principles it is probably unacceptable to everyone, if it violates two principles it might become more acceptable to some speakers etc. There is probably more that needs to be said about the precise content of these principles and their interaction and I would not claim that they would suffice to precisely account for the acceptability of gapping. However given the extreme variation from speaker to speaker concerning gapping acceptability, the idea that perceptual factors play an important role in determining the interpretation of gapped clause is an attractive one. I will not attempt to refine Kuno’s analysis further, rather I will assume that something of the order of Kuno’s principles is at work and that it interacts with the grammar described above to limit overgeneration.

3.1.4 Previous approaches to gapping

The literature on gapping is extensive. Transformational accounts include [Jackendoff 1971; Hankamer 1971; Langendoen 1975; Stillings 1975 and Sag 1980]. More recent analyses include [Chao 1987; Russell 1987; Sag et al 1984 and Steedman 1989]. In this section, I briefly describe each of these more recent proposals and show how they compare with the analysis described in section 3.1.2.

[Steedman 1989] — a Combinatory Categorial Grammar (CCG) approach
Steedman’s approach to gapping relies on three main ideas. First, it exploits the fact that in CCG, the notion of a constituent is a highly permissive one: the reason for this is that the grammar rules allow the formation of constituents which in standard grammars do not exist. For instance, in Steedman’s grammar the subject can combine with the verb to form a constituent. This allows for a simple account of the non-standard syntax exhibited by gapped clauses. Second, it uses the parametric neutrality of combinatory rules according to which given any three constituents related by a CCG rule, knowledge of any two of these constituents will suffice to determine the third (for instance, if we know that some unknown constituent X forward combines with a constituent of category A to yield a constituent of category B, then we know that the category of X is B/A). This property of CCG is embodied in what Steedman calls the revealing rule (notated Decomp) whose left version ensures that given any rule application of the form X, Y → Z knowledge of Y and Z will yield X. The revealing rule allows for the recovery of a ‘revealed’ constituent which will eventually contribute to determine the interpretation of the gapped clause. Third, it adopts Kuno’s proposal that the semantics of the gapped constituent in the gapped clause corresponds to the given information (or topic) of the antecedent clause. This contributes both to determining the meaning of the gap and to licensing the syntax of the gapped clause.

Without going into too much details, Steedman’s analysis can be illustrated by the following schema (T abbreviates type raising, B, composition, <Decomp the left-conjunct revealing rule and < &, backward coordination).

\[
\begin{align*}
\text{TOPIC} & \quad \text{TOPIC} & \quad \text{ACLause} & \quad \text{and} & \quad \text{EClause} \\
V & \quad V & \quad S & \quad & \quad \text{S/V} < \text{Decomp} \\
& & \quad \text{S/V} < \text{T, B} & \quad \text{S/V} < \text{&} \\
& & \quad \text{S/S} < \text{FA} & \quad \text{S} \\
\end{align*}
\]

An informal gloss of this derivation is as follows. First the revealing rule (<Decomp) yields a constituent of category S/V by ‘abstracting away’ the topic from the a-clause. Simultaneously, type raising and composition ensure that the gapped clause is assigned a category of type S/V where V ranges over any type of verb. Second, these two constituents of category S/V are conjoined by the constituent coordination rule. Third, the resulting constituent is backward applied to the topic category. Intuitively, the approach boils down to abstracting the topic from
the a-clause, conjoin the result of this abstraction with the e-clause and ‘quantifying’ the topic back into the resulting coordinated construct. Although the intuition is fine, I find that there are several important problems with this approach.

First it is unclear how pragmatics interacts with syntax and semantics. In the version of CCG which Steedman uses to develop the above treatment of gapping, syntax and semantic operate in tandem within a unification based extension of traditional CCG. Thus any category contains both syntactic and semantic information and combination of two categories crucially involves unification in a way similar to UCG(2). What is unclear however is the status of the topic category and of its interaction with syntax. According to Steedman, the topic category is just like any other category (that is, it contains both syntactic and semantic information) except that it is not derived by the grammar; it is given by the pragmatic component. The way in which this occurs is left undefined which is fair enough given the current state of our knowledge about pragmatics. What is more worrying however is the relationship which Steedman’s approach suggests between pragmatics on the one hand and syntax and semantics on the other. According to his treatment, it seems that the grammar should allow for two types of objects: purely linguistic objects such as NPs on the one hand and pragmatic ones such as the topic category on the other. Such an approach runs contra to the idea that linguistic information can be represented by complex feature structures whose mode of combination account for the interaction of the different components. It is also unclear how and why the syntactic rules should apply to pragmatic objects. More specifically, it is unclear how syntax and pragmatics are made to interact in such a way that the derivation in (3.40) obtains.

From the empirical point of view the following shortcomings can be noted.

Because the category of the e-clause must first conjoin with that of the revealed constituent and then apply to the topic category, the relationship between the category of the missing category in the gapped clause and that of the constituent representing the topic of the antecedent clause is in effect one of identity. A problem with this analysis is that such a rigid relationship between the missing head of the gapped clause and that of the topic category is in fact not appropriate. Thus Steedman’s analysis will fail to account for such sentences as Jon is rather foolish and Lou, a complete idiot because analysis of the right conjunct will yield a leftward looking functor over a transitive verb subcategorising for an NP whereas the topic category (which is extracted from the a-clause) will be that of a verb subcategorising for a predicative expression. Note however that this shortcoming could probably be remedied providing that the treatment of coordination proposed by Steedman were amended to take into consideration the proposal made by [Sag et
al 1984] in order to deal with such cases of coordination as Lou is rather foolish and a complete idiot.

Another problem appears with respect to cases where the gapped clause only misses an auxiliary as in Jon will kiss Mary and Peter, talk to Sarah. Under Steedman’s analysis, this sentence will be judged ungrammatical because the remnants in the right conjunct will not be able to compose to yield a unique constituent. To see this, recall that the category of a gapped clause is obtained by first type-raising and then composing the categories of the gapping remnants. In the present example, type raising would yield the sequence $S_{[fin]}/(S_{[fin]}\backslash NP) S_{[inf]}\backslash NP$ (where $\text{Cat}[X]$ indicates a category with syntactic category $\text{Cat}$ and morpho-syntactic feature $X$) i.e, a sequence of the form $X/Y W/Z$. Since this sequence does not conform to the requirements set by the rule of composition (which may only apply to sequences of the form $X/Y Y/Z$) the two constituents cannot combine and derivation fails.

A third problem with Steedman’s approach is that it makes no distinction between gapping within adjuncts and gapping within arguments. So for instance, the grammar will allow both This doctor said that I should eat salmon, and this doctor said I should eat tuna and *Jon left without telling his boss, and Bill his colleagues because in both cases the missing head is a transitive verb and a category can be produced for the topic of the antecedent clause which is also that of a transitive verb\(^{18}\). Since the verbal function looked for by the gapped sentence is identical in category with that associated with the topic in the antecedent clause, derivation proceeds in both cases without failing and the two sentences are judged grammatical.

The analysis also fails to account for cases of multiple gaps. Multiple gaps pose two problems here. First, how is the right conjunct to account for the fact that two gaps are present? Note for instance that in Harry wants Ipswich to win and Barry, Watford the right conjunct will be assigned the same category as in Jon loves Mary and Peter, Sarah i.e, a lefward looking functor over transitive verbs, clearly not what is needed. Second, how is the category for the given information to be constructed? As Steedman’s himself remarks: “the important question of how the corresponding syntactic category is made available from the analysis of the left conjunct remains open”.

On the more positive side, an interesting characteristic of Steedman’s proposal is that, given

\(^{18}\)I am assuming here that the topic category is directly derived from the syntax of the a-clause that is, that it corresponds to some sub-constituent in the a-clause. This might be presuming however since Steedman leaves open the question of how the syntax of the topic category is to be defined.
the general framework provided by CCG, it explains why verb initial and verb medial languages only allow forward gapping while verb final languages only allow backward gapping.

[Chao 1987] - a GB approach

Chao's analysis of gapping can be summarised as follows. First a defective X schema is introduced to license the overt constituents of gapped clauses. This schema is characterised by the fact that it allows for constituents whose structural head is not instantiated and in which head projections are optional. To limit overgeneration, the notion of an H⁻ projection is then defined which in effect implements the "missing head restriction" noted in section 3.1.1 (if an H projection consists of those categories that inherit the descriptive content of a semantic head, then an H⁻ projection is an H projection that only contains X⁻ constituents that is, constituents derived under the defective X schema). Finally to ensure that the resulting structures are well formed at the level of logical form, an e-reconstruction procedure is introduced. E-reconstruction substitutes the major constituents of an H⁺ projection under the corresponding nodes in the H⁻ projection of the corresponding e-clause where correspondence between nodes results from a mapping between the H⁺ projection of the a-clause and the H⁻ projection of the e-clause. For the sentence to be accepted as grammatical, the resulting LF must obey both the theta-criterion and the principle of full interpretation. In effect this ensures that every constituent has a unique semantic and thematic role to play in determining the overall meaning of the associated sentence.

From the empirical point of view, the differences between Chao's approach and mine are small. A first difference concerns Sag et al's data on the subcategorisation identity of a-clause and e-clause: because the notion of correspondence introduced by Chao requires that any two nodes which stand in a correspondence relation to each other be of the same syntactic category, it is necessary that the remnants in the gapped clause bear the same category as the corresponding constituents in the a-clause. Thus Chao's analysis will fail to account for the grammaticality of Jon is rather stupid and Lou, a complete idiot. A second difference is that Chao's analysis will also fail to account for the grammaticality of Jon will go to Paris and Peter, stay at home because the V stay at home is not a headless constituent. Thus the headless constituent formed by the VP stay at home will contain a non X⁻ constituent. Thus the resulting projection will not be an H⁻ projection and will be ruled out at the level of surface structure.

More interestingly, the differences between Chao's approach and mine stem essentially from the theoretical differences between the two grammatical frameworks used. UCG(2) is a monostratal theory, that is one in which all sources of linguistic information are grouped into one single repre-
sentational level. By contrast, GB allows several levels of representation: deep structure, surface structure, a level of phonological form and one of logical form. With respect to the treatment of gapping, this difference is reflected as follows. Whereas in Chao's account, gapped clauses are licensed by the intricate interaction of various principles (principle of full interpretation, theta criterion, defective X schema, correspondence between projections, e-reconstruction etc) operating on different representational levels of linguistic structure, in UCg(2) all constraints are stated on one single representational level: the sign. The main advantage of such an approach is that it is more perspicuous and as such more amenable to computation as the linguist need only consider one level of representation rather than keep in mind the various principles interacting between the different linguistic levels. Note also that from a methodological point of view, because UCg(2) is an implemented theory of grammar, it allows for direct (and impartial) testing of the theory. That is, the theory is more easily falsifiable.

[Sag et al 1984, Russell 1987] - two GPSG approaches

Russell's approach relies on two main ideas. First a metarule is introduced to account for the syntax of gapped clauses: this metarule in effect licenses null verbs in the syntax. Second, an interpretive procedure is defined which imposes conditions on gapping which are usually placed on the syntactic form of gapping rather than on its semantics: for instance it ensures that the antecedent clause precedes the gapped clause and that the two clauses are related by some coordination constituent. An important constraint placed on this interpretative procedure is that the semantic type of the antecedent must be the same as that of the missing head in the gap clause.

The main problem with this approach is that in effect the requirement that the two verbs in the a- and in the e-clause have the same subcategorisation potential is stated at the semantic rather than at the syntactic level (this is essentially the import of the restriction on semantic types mentioned above). As a result sentences such as Jon gave a book to Rose and Peter, Claire a panda will be accepted as grammatical.\(^{19}\)

In [Sag et al 1984] a GPSG-based approach is developed which is very similar in spirit to the analysis proposed in section 3.1.3 above. A coordination rule is proposed which allows a list of maximal projections to coordinate with a sentence or a VP under the condition that the resulting structure is interpretable. Furthermore the interpretation of a gapped clause is determined as

\(^{19}\)Though of course, this depends on the type of semantics used. If the semantics allows mention of theta-roles in the semantic representation, then the problem just mentioned does not arise.
follows. First the local trees associated with each remnant in the gapped clause are substituted for the corresponding constituent in the derivation tree associated by the grammar with the antecedent clause. Second, the semantics of the resulting tree is computed: this yields the semantics of the gapped clause verifying at the same time that the new tree is syntactically well-formed with respect to the grammar.

There is one main difference between this approach and mine. Sag et al's does not incorporate the notion of a major constituent whereas mine does. In effect, under Sag's analysis the local tree of each remnant in the gapped clause may substitute for any local tree in the derivation tree of the antecedent clause. Restrictions on this substitution process result from the requirement that the resulting tree must be derivable by the grammar. Because there is no such notion as major constituency, ungrammatical sentences such as Jon left without telling his colleagues and Peter, his friends will be accepted by the grammar. Another problem due to the lack of a "major constituent constraint" is illustrated by sentences such as First Kim's picture of Sandy astonished Fido, and then Lee's Felix. Under Sag et al's analysis, this sentence will be assigned two readings rather than one as expected: one in which Felix has been substituted for Fido (the correct reading) and another in which Felix will have been substituted for Sandy. Similarly, for Kim and Sandy astonished Fido and Lee, Felix, five readings will be produced which correspond to each of the possible substitutions of Lee and Felix with respect to the three NPs occurring in the antecedent clause: Kim, Sandy and Fido.

3.2 Verb Phrase Ellipsis

3.2.1 Syntactic properties of Verb Phrase Ellipsis

A salient characteristic of VPE is that the VP in the e-clause is an auxiliary. For instance in Jeremy fed his crocodile and Sarah did too, the VP in the e-clause consists of the auxiliary did. The lack of an auxiliary in the e-clause renders the sentence ungrammatical as illustrated by the following examples taken from [Gazdar et al 1982].

\[\text{Jeremy fed his crocodile and Sarah did too, the VP in the e-clause consists of the auxiliary did.}\]

\[\text{The lack of an auxiliary in the e-clause renders the sentence ungrammatical as illustrated by the following examples taken from [Gazdar et al 1982].}\]

\[\text{[Gazdar et al 1982]}\]
(3.41) a. *First Sandy began reading Freud and then Sandy began.
   
   b. *Kim got arrested by the CIA and Sandy got also.
   
   c. *Kim considered joining the navy but Sandy never considered.
   
   d. *Kim wanted to go and Sandy wanted, too.

As observed in [Sag 1980], another restriction on the auxiliary of the VPE clause is that it may be neither in the gerund nor in the progressive form. This is illustrated by the following examples (also taken from [Gazdar et al 1982]).

(3.42) a. *Kim is being noisy and Sandy is being, too.
   
   b. *Kim was being watched by the FBI and Chris was being, too.

In comparison with gapping, the syntactic relation which may hold between an e-clause containing a VPE and the corresponding a-clause is very permissive. Consider the following examples.

(3.43) a. Jeremy smiled and Sylvette did too.
   
   
   c. Jeremy smiled before Sylvette did.
   
   d. Speaker A: Who ate the cake?
      Speaker B: Bruno did.
   
   e. If Jeremy can, he will go running.
   
   f. Jeremy knows all the crocodiles which Sylvette does.

(3.43a) illustrates a case where the relationship between e-clause and a-clause is one of intrasentential coordination. (3.43b) shows a case of intersentential VPE. In (3.43c), the e-clause is subordinated to the a-clause. In (3.43d), VPE occurs across speakers in discourse. (3.43e) illustrates a case of backward anaphora where the VPE precedes and is more deeply embedded than its antecedent. Finally (3.43d) is a case of antecedent contained ellipsis where the e-clause is contained within the a-clause. In short, the syntactic relationship holding between e- and a-clause appears to be unrestricted. However there are some syntactic restrictions on how the syntax of the e-clause relates to that of the a-clause. In particular, the possible values of the
verbal features of the e-clause are restricted by the values of these features in the a-clause. Furthermore the presence or absence of arguments in the e-clause also influences the syntactic status of the antecedent VP. In the rest of this section, I consider each of these phenomena in turn.

As often observed in the linguistic literature, the tense and modality of the antecedent VP are not inherited by the VPE clause. This allows for instance sentences such as *Jon ran yesterday and Peter will tomorrow, where the a-clause is in the simple past whereas the e-clause is in the future tense. By contrast, voice and aspect are or can be inherited. Consider voice first.

\[\text{(3.44)}\]

\begin{align*}
\text{a.} & \quad \text{Jeremy smiled, and Sylvette did too.} \\
\text{b.} & \quad *\text{Jeremy smiled, and Sylvette was too.} \\
\text{c.} & \quad \text{Although Jeremy was often smiled at, Sylvette wasn't.} \\
\text{d.} & \quad *\text{Although Jeremy was often smiled at, Sylvette didn't.}
\end{align*}

While (3.44a,c) are fine, (3.44b,d) are ungrammatical because the voice of the e-clause differs from that of the a-clause. So concordance of voice between a- and e-clause is obligatory with regard to VPE\(^{21}\). Aspect behaves somewhat differently, as the following examples illustrate.

\[\text{(3.45)}\]

\begin{align*}
\text{a.} & \quad \text{Jeremy may be questioning our motives but Peter isn't.} \\
\text{b.} & \quad *\text{Jeremy may be questioning our motives but Peter doesn't.} \\
\text{c.} & \quad \text{Jeremy may have questioned our motives but Peter hasn't.} \\
\text{d.} & \quad *\text{Jeremy may have questioned our motives but Peter isn't.} \\
\text{e.} & \quad \text{In the past, Jeremy may have been questioning our motives but now he isn't.} \\
\text{f.} & \quad \text{Jeremy may have been questioning our motives but Peter hasn't.}
\end{align*}

Examples (3.45a-d) show that the aspect (perfective or progressive or both) of the antecedent VP must be compatible with the subcategorisation requirements of the VPE auxiliary. Thus in (3.45a) and (3.45b), the antecedent VP is in the progressive. In the first case the VPE auxiliary requires a progressive VP and the resulting sentence is grammatical. By contrast in the second case (3.45b) the VPE auxiliary does is not compatible with a progressive VP and the resulting

\(^{21}\)As noted in [Sag 1980] there are cases of VPE which do allow for voice to differ between the two clauses. For instance *That can all be explained. Please do I have nothing to say about such examples.*
sentence is ungrammatical. (3.45c-d) are similar save that the antecedent VP is in the perfective aspect. (3.45e) shows that when the antecedent is both perfective and progressive, the VPE clause may be in the progressive only whereas, (3.45f) shows that the VPE clause may also be both in the progressive and in the perfective aspect.

Then there is the question of how arguments in the a-clause relate to the VPE in the e-clause. In other words: are all arguments in the e-clause always inherited by the a-clause? Usually they are. However there are two exceptions: pseudo-gapping ([Levin 1978]) and antecedent contained deletion ([Bouton 1970]). In each case, the e-clause inherits a VP lacking one argument rather than a saturated VP. Pseudo-gapping is illustrated by the following examples (taken from [Russell 1987]).

(3.46) a. They won't read poetry but they will newspapers.
    b. They won't try to attack us unless we do them.
    c. Cedric likes water more than Gloria does gin.
    d. He didn't bite me. I might him.

Pseudo-gapping is characterised by the fact that the e-clause contains an auxiliary and one argument. There are two reasons for thinking that pseudo-gapping is more properly classified with VP-ellipsis rather than with gapping. First, like VPE, pseudo-gapping sentences contain an auxiliary; if this is omitted the result cannot be differentiated from gapping, and if a non-auxiliary verb is substituted, the result is ungrammatical: *They won't try to attack us unless we try them. Second, as observed in [Levin 1978], pseudo-gapping is permissible in a range of syntactic environments that corresponds very closely to those in which VPE occurs.

Antecedent-contained deletion (ACD) is characterised by the fact that an e-clause containing a VPE is contained within the corresponding a-clause. Example (3.47) below illustrates this.

    b. No one greeted everyone Bill did.
    c. Betsy's father wants her to read everything her boss does.
    d. Sam put everything Mary told him into the case.
    e. Jon visited every girl who asked him to.
(3.47a) gives the basic case of antecedent contained deletion. In [Sag 1980], it is argued that ACD is only allowed if the object NP within which VPE occurs has wide scope over the subject NP of the corresponding verb. (3.47b) shows that this claim is inaccurate as there is no reading under which everyone has wide scope over the subject as one, that is, (3.47b) cannot mean that for any person greeted by Bill, there is no one who greeted this person. (3.47c) shows that the antecedent VP need not be a main verb, that is, it can be embedded within a complement of the main verb. (3.47d) shows that ACD is possible with ditransitive verbs and (3.47e), that the VPE in an ACD can be embedded too.

3.2.2 Analysis and data

Before we consider the issue of how the restrictions on VPE observed in the preceding section are to be accounted for by the grammar, two more fundamental questions about the syntax of VPE need to be answered. First, what is the syntactic category of a clause containing a VP ellipsis: is it a sentence or a sentence missing a VP? Second, should a VPE be represented by an empty VP? That is, which of the two trees in Figure 3.4 should be assigned to a VPE-clause? (I use the convention that VP, denotes a sign with category VP, empty phonology and unknown semantics). I consider each question in turn. Firstly, what is the category of a clause containing a VPE? As we have seen in section 3.2.1, VPE clauses, like sentences, can be coordinated,

\[ \text{Figure 3.4: Possible phrase structure trees for a VPE clause.} \]

22Note that antecedent contained deletions are special cases of VP ellipsis in that their syntactic environment is restricted to relative clauses.
subordinated or root. That is, they have the same distribution as sentences. Furthermore, if we adopt the generally accepted view that only constituents of the same category may coordinate, then the fact that VPE clauses coordinate with sentences further suggests that VPE clauses are sentences. In what follows, I take these syntactic facts about VPE clauses to be sufficient evidence for categorising VPE clauses as constituents of category S.

Next I discuss the following question: should VPE induce the presence of empty VPs in the grammar? Syntactically, VPE clauses deviate from the canonical pattern of sentence constituency in that rather than consisting of a subject NP followed by a VP, they consist of a subject followed by an auxiliary. Assuming that auxiliaries are assigned the category VP/VP, this means that VPE clauses are sentences missing a VP. But as we have seen above, there is compelling evidence that VPE clauses have category S. How are we to accommodate this contradiction? In many linguistic theories (Government and Binding theory, Generalised Phrase Structure Grammar, Head-driven Phrase Structure Grammar), missing constituents are represented by phonologically null constituents. Null constituents come in two types, depending on whether they are syntactically or semantically controlled. Syntactically controlled null constituents are usually referred to as traces or gaps whereas semantically controlled null constituents are generally called empty categories or null anaphors. The basic intuitions behind these two types of categories are the same: (i) empty categories allow for the standard notion of constituent structure to be preserved (for instance, in Jon, Mary likes we can preserve the idea that likes subcategorises for an object NP occurring to its right by saying that the missing NP is in fact not missing and is represented by a trace occurring in lieu of the missing object) and (ii) empty categories must be licensed (either syntactically or semantically) by some other object called the filler (in the case of syntactically controlled gaps) or the antecedent (in the case of null anaphors).

However not all linguistic theories assume the presence of empty categories. UCG(2) in particular, does not. There are several reasons for this. Firstly, UCG(2) is both a feature-based and a categorial grammar and as such, belongs to a paradigm of grammars which are traditionally (though not always\(^2\)) surface-based. Surface-based grammars provide a direct characterisation of the actual surface order of the string elements in a sentence. Typically, surface oriented theories of grammar do not resort to empty categories because this would mean hypothesising the content of the string (i.e., hypothesising the presence, the number and the position of traces in the string) rather than taking it as it is. Secondly, there is no tangible evidence for the existence of empty categories, thus Occam’s razor will favour any theory which does not resort

\(^2\)Thus RMPSG is feature-based but not wholly surface-based in that it allows empty categories. Similarly, the Lambek calculus although categorial is not surface-based for the same reason.
to such additional theoretical constructs. Moreover, [Pollard/Moehier 1989] argue that "there simply is no such thing as the location of the gap in the phonological structure of the phrase that contains it". Although this may sound controversial in the case of languages with a relatively fixed word order such as English, it is undoubtedly true when free-constituent-order languages like Japanese are taken in consideration: for these languages it is simply impossible to locate the null constituent. In a theory which does not posit the existence of empty categories, this issue simply does not arise. Thirdly, ceteris paribus a grammar without empty categories is more computationally tractable than one with empty categories because the problem of hypothesising and locating traces does not arise.

So, there are good reasons for avoiding empty categories in the grammar. To complete the argument, I now need to show how an account of VPE can be developed within UC(2) which does not resort to empty categories. Let us start by considering the UC(2) treatment of auxiliaries. Auxiliaries are associated with signs of the form:

\[
\begin{align*}
\text{pho} &: \text{PhoF} \\
\text{synt} &: \text{VP} \\
\text{sem} &: \text{SemF} \\
\text{order} &: \text{OrderF}
\end{align*}
\]

That is, an auxiliary is a VP subcategorising for another VP. Now consider the case of an auxiliary occurring in a VPE clause, or VPE auxiliary. If a VPE clause has category S and if the grammar does not allow empty categories, then every auxiliary clause must have two categories: VP and VP/VP. So, of the two possible derivations indicated in Figure 3.4, the leftmost one is the one I adopt. Still, there is a sense in which a VPE auxiliary is lacking a verb phrase. Consider the semantics. Intuitively, the semantics of a VP ellipsis is provided by the semantics of the VPE auxiliary applied to the semantics of the antecedent VP. Under the present approach, this antecedent is taken to be identified at the level of discourse. There are two possibilities here: either a threading technique could be used which would pass on a

\[24\] For more details see chapter 6.
list of VP semantics available for anaphoric reference, or some resolution process could search the derivation tree of the a-clause for a VP sign whose semantics would provide the semantic argument required by the semantics of the auxiliary. The first possibility assumes that VPE is a purely semantic process, that is, that non-semantic information is irrelevant when resolving VPE whereas the second doesn’t (because the sign of the antecedent VP contains not only semantical but also syntactic, phonological and discourse information). As we have seen in the preceding section, some non-semantic factors have to be checked upon when resolving VPE. For instance, the voice of the antecedent must be identical with that of the e-clause. I will therefore assume an approach of the second type. Suppose then, that auxiliaries that occur in VPE clauses are assigned a sign of the form:

\[
\begin{align*}
\text{pho} & : \text{Pho} \\
\text{synt} & : \text{VP} \\
\text{formula} & : \text{Sem1(Sem2)} \\
\text{disgap} & : (\text{Pho}2: \text{VP}: \text{Sem}2: 02) \\
\text{order} & : \text{Order}
\end{align*}
\]

where the new attribute disgap denotes a discourse gap i.e., a missing constituent which must find an antecedent at the level of discourse\(^{25}\). The restrictions on the value of this discourse gap state that the filler must be a sign of category VP and semantics Sem2 which provides the semantic argument to the function Sem1 denoting the semantics of the auxiliary\(^{26}\). For instance, if the semantics of the antecedent is \(A_x.\text{run}(x)\) and the auxiliary is \(A P.A_y.\text{will}(P(y))\), the semantics of the resulting VP will be \(A_x.\text{will}(\text{run}(x))\).

Clearly, this succeeds in adequately modelling the approach to VPE sketched above that is, one in which VPE is taken to be an ellipsis which is not associated with an empty category. I now

---

\(^{25}\)Although I use a terminology akin to the one used in connection with the phenomena of unbounded dependencies, it should, I hope, be clear that the notion of discourse gap differs from that of syntactic gap in that the latter is licensed by the identification of a filler at the level of syntax whereas the former requires a filler within the discourse domain. It follows that unbounded dependencies can be subject to syntactic constraint such as subjacency whilst discourse gaps do not because they are regulated by discourse rather than syntactic constraints. The analogical terminology is meant to point to the similarity in the treatment of syntactic and discourse gaps: both must be licensed by some other object, a filler in the case of a syntactic gap and an antecedent in the case of a discourse gap.

\(^{26}\)This is oversimplified. Chapter 4 gives a more appropriate semantics for VPE.
proceed to show how the grammar can be completed to account for the restrictions on VPE observed in section 3.2.1.

A first requirement on the syntax of a VPE clause is that it contains an auxiliary. So for instance, *First Sandy began reading then Paul did* is grammatical but *First Sandy began reading then Paul began* is not. Under the present approach, this requirement follows from the fact that sentences must have category S. Because auxiliaries can be fully fledged VPs, VPE clauses can be fully fledged sentences. In contrast, any clause containing a verb subcategorising for a VP but no corresponding VP (for instance, *Sandy considered or Jon got*) will have category S/VP (if any) and the associated sentence will thus fail to be recognised by the grammar as being a sentence of English, which is as required.

A second requirement on VPE clauses is that the auxiliary be neither a gerund nor a progressive. This can be encoded as follows. Suppose we derive VPE auxiliaries from ordinary auxiliaries by means of the following lexical rule\(^27\):

\[
\begin{align*}
\text{pho} : \text{PhoF} \\
\text{synt} : \text{VP}/ \\
\text{sem} : \text{SemA} \\
\text{order} : \text{post}
\end{align*}
\]

We can then restrict the application of this rule to auxiliaries whose verb form value is different from *prp* (for present participle i.e., gerunds and progressive) simply by stating that the verb form value of the input sign may not be *prp*\(^28\). Thus *Peter won’t have* will be a derivable sentence under the grammar because the VP *have* can be derived by the above lexical rule. By contrast, *Peter is being* is not a derivable sentence because the auxiliary *being* has the verb form value

\(^{27}\)There are cases of VP ellipsis where the antecedent is not a VP. For instance, in *Jon is stupid and Peter is too*, the antecedent is an AP. To deal with this cases, we could modify this lexical rule to ensure that the value of the syntactic attribute of the active part of the sign unifies with the value of the syntactic attribute in the discourse gap.

\(^{28}\)Recall that the input sign is the sign to which the lexical rule applies to yield a new (output) sign. Graphically the input sign appears to the left of the arrow in the lexical rule and the output sign, to the right.
so that the above lexical rule cannot apply to it to produce a VP. Thus is and being cannot form a VP and Peter is being cannot be a sentence.

At first sight, this analysis may seem rather ad hoc. One could argue for instance that such a restriction on VPE acceptability should fall out of the syntactic structure of the auxiliary system. Such a position is adopted in [Akmajian et al. 1979] where VPs are defined by the rule:

$$V^n \rightarrow V[+\text{Aux}], V^{n-1}$$

and VP deletion is defined as deleting any $V^n$ with $n \geq 1$. Under this analysis, sentences such as Kim is being noisy are assigned the tree structure:

$$V^2(V(is), V^1(V(being), AP(noisy)))$$

so that Kim is being noisy and Sandy is too is grammatical because being noisy is a constituent of category $V^1$ and consequently, may be deleted. In contrast, *Kim is being noisy and Sandy is being too cannot be derived because the potential deletion target noisy is only a subpart of $V^1$ and thus may not be deleted. As argued in [Gazdar et al., 1982], there are two problems with the view that restrictions on VPE should follow from the syntax of the auxiliary system. First it fails to generalise to other closely related examples where VPE is blocked:

(3.51) *Kim’s having resigned was surprising, but Lee’s having came as no surprise.29

Second, it fails to account for the fact that as shown in [Huddleston 1978], in many varieties of English (including British English) examples like (3.42) are fully acceptable. Thus a feature-based approach such as the one described above has two advantages. First it adequately embodies Sag’s generalization that VPD is blocked if an -ing form immediately precedes the deletion target. Second, as Gazdar et al remark: “it allows for subtle variation of language to be treated by subtle variations in the grammar” [Gazdar et al. 1982:607].

A third requirement on VPE is that the subcategorisation requirements of the VPE auxiliary are satisfied by the antecedent. As discussed at length in the preceding section, only the aspect and the voice of the antecedent need to be taken into account. Briefly, the voice of the a-clause must be the same as that of the e-clause. Furthermore, the aspect of the two clauses must be identical and the aspectual requirements of the VPE auxiliary must be satisfied. Again this is easily encoded in the lexical rule by requiring that the aspectual value of the sign associated

29 This should be compared with Kim’s having been assigned to the patrol was surprising, but Lee’s having been came as no surprise.
with the antecedent VP be set by the VPE auxiliary. For instance, the sign associated by the lexical rule with the VPE auxiliary is will be such that the aspectual value of the antecedent is as required by is i.e., progressive. Thus Jeremy may be questioning our motives but Peter isn't is accepted by the grammar whereas Jeremy may have questioned our motives but Peter isn't isn't. In this way, the subcategorisation requirements of the VPE auxiliary can be taken into consideration.

Finally, antecedent contained deletion and pseudo-gapping need to be accounted for. Antecedent contained deletion is treated as a case of VPE involving a filler-gap dependency, and where the syntax of the antecedent is TV rather than VP. For instance, in Jon reads every book that Bill does, the value of the disgap attribute will indicate that the syntactic category of the antecedent must be TV. Furthermore, the unary rule mapping a verb phrase missing n arguments into a verb phrase missing n - 1 arguments and a filler will apply to the auxiliary in the e-clause so that the relative clause that Bill does is assigned the derivation in Figure (3.5) where S[NP] and VP[NP] are abbreviations for signs of category S and VP respectively, and such that their threading attribute indicates a missing NP. That is, the UGC(2) treatment of filler-gap dependencies extends to VPE auxiliaries. To deal with pseudo-gapping, we introduce

![Figure 3.5: Antecedent contained deletion.](image)

two lexical rules which take as input a VPE auxiliary and returns a new auxiliary similar to the input sign save that this new auxiliary subcategorises for an object NP or PP. Thus in they won't read poetry but they will newspapers, the e-clause will be derived as indicated in Figure (3.6)
where VP/NP is a sign derived by the first of the two new lexical rules.

\[
\begin{array}{c}
S \\
 NP \\
  \text{They} \\
 S/NP \\
  \text{will} \\
  \text{VP/NP} \\
 NP \\
  \text{Newspapers}
\end{array}
\]

Figure 3.6: Pseudo-gapping.

This concludes the description of the analysis of VPE. The next section shows how this analysis compares with existing approaches.

### 3.2.3 Previous approaches to VPE

[William 1977]: a transformational interpretive approach

[William 1977] distinguishes between two components of grammar: the sentence grammar and the discourse grammar. Furthermore, he posits a principle of "strict utterance" according to which all rules of discourse grammar apply after rules of sentence grammar. With respect to VPE, he argues that VPE is a discourse level phenomenon and should thus be handled by the discourse grammar.

There are several problems with this approach. First, since the rules of the discourse grammar apply only to the logical forms provided by the sentence grammar, this means that no syntactic constraints can be stated on VPE. That is the subcategorisation requirements of the VPE auxiliary cannot be checked. Second, since only the logical forms associated with full VP are used, cases of pseudo-gapping cannot be handled. Third, as argued in [Stainton-Ellis 1988], it is unclear how gapping is to be dealt with. Clearly, gapping is subject to syntactic constraints. Thus under William's approach, it should be dealt with by the sentence grammar. But clearly,
gapping can occur across discourse speakers (cf. *I will miss you. And I, you*). Thus the serial architecture of language processing proposed by Williams seems to run into a problem here. More generally, this model is psychologically not very plausible. In fact research by [Marslen-Wilson/Tyler 1977] strongly suggests that processing is carried out in an incremental fashion whereby words and sentences are parsed and interpreted from left-to-right with grammar modules interacting in parallel.

[Sag 1980]: a transformational deletion approach

[Sag 1980] argues for a mixed theory of VPE in which both deletion and interpretative rules have a role to play. A deletion rule operating at the phonological level allows any VP to be deleted provided that some other VP is available whose logical form (LF) is an alphabetic variant of the VP to be deleted. Under Sag's analysis VPE is a syntactic phenomena and should thus be dealt with at the syntactic level.

The main attraction of Sag's analysis is that it does allow for both syntactic and semantic constraints to be taken into account: the antecedent must be a VP and the semantic representation of the deleted VP must be an alphabetic variant of that of the antecedent. However, several problems arise. First, because the antecedent must be a VP, Sag's analysis will fail to account for cases of pseudo-gapping. Furthermore, it leads to a treatment of antecedent contained deletion which is unsatisfactory for two reasons. First, it requires that the object NP containing the VPE have wide scope. As we have seen in section 3.2.1, this is not necessarily the case. Second, it leaves unexplained the syntax of object relative clauses containing VPE. If such clauses contain a full VP, then the syntax must be extended to account for the fact that object relatives may include a full sentence provided that the corresponding VP is elided. This seems to be a rather odd analysis of relatives. Sag leaves the issue undisussed.

Another problem with Sag's analysis is that it assumes VPE to be resolved at the level of syntax. But as already mentioned, this is clearly not the case and it is not difficult to find cases of VPE which apply across discourse as illustrated in example (3.52) taken from [Chao 1987].

(3.52) Italian authorities apparently figured that the lure of a lot of cash might tempt Liceo

---

30Semantic constraints on VPE are discussed at length in chapter 5. In particular, the import of the notion of alphabetic variance on VPE permissibility is made clear there.
Galli to show his face in Europe, where they could get their hands on him. Last week, in any case, it did, he did and they did. (NYT Sept 19, 1983)

From a computational perspective, the problems raised by Sag's analysis are those raised by any transformational account. Because transformations operate from the deep structure to the surface structure, the grammar is geared toward generation rather than analysis. The use of multiple levels of linguistic structure makes grammar development and testing difficult.

[Chao 1987]: a GB-based approach

Chao's syntactic analysis of VPE\textsuperscript{31} is that VPE is represented in the syntax by an empty category. Further restrictions on VPE come from the fact that in GB any empty category must be licensed and identified at the level of logical form. Under Chao's analysis VPE is subject to the general empty category principle and is thus licensed if it is properly governed by a lexical INFL (that is, an INFL such that information about tense and agreement are instantiated by some lexical item such as a modal, an auxiliary verb or the do of do-support construction). Furthermore, it is identified provided that its descriptive (i.e., semantic) content can be determined. There are basically two ways according to which the descriptive content of a VPE can be recovered: either the missing VP can be syntactically reconstructed from the a-clause or the VPE may be identified with a discourse-supplied functional antecedent (syntactic reconstruction is needed for cases where the antecedent of the VPE contains an extracted NP or trace).

A first problem with Chao's approach is that, as in Sag's analysis, the antecedent of a VPE must be a VP itself, which again leaves pseudo-gapping unexplained. And again, antecedent contained deletion is only allowed if the object NP containing the VPE takes wide scope over the sentence. Note further that in the approach I propose, the distinction between syntactic and discourse antecedent introduced by Chao to account for cases where the antecedent VP includes an extracted NP is superfluous. This is because in the present approach, the antecedent is defined both syntactically and semantically. Thus in the case of pseudo-gapping, the syntax of the e-clause indicates that the antecedent must be a transitive verb, rather than an intransitive one. Similarly, in the case of antecedent-contained deletion the syntax of the auxiliary in the e-clause clearly indicates first, that the antecedent VP must be a TVP an second, that a filler

\textsuperscript{31}Another GB-based analysis of VPE is developed in [Lobeck 1990]. As it is very similar to Chao's it will not be discussed here.
must be provided for the missing NP in the e-clause. If neither of these two conditions is respected, analysis will fail and the corresponding sentence will be rejected as ungrammatical. In this way, the fact that antecedent contained deletion and VPE on an antecedent containing some extracted NP may only occur within one sentence need not be postulated as is done in [Chao 1987].

3.3 Summary and concluding remarks

At the onset of this chapter, I set out to answer two questions about the syntax of ellipsis: first, how are the constituents in the e-clause to be licensed (or in other words, what is the category of the e-clause)? And second, how is the elliptical construct itself to be licensed? The answers given in this chapter can be summarised as follows. First, I make a fundamental distinction between VPE and gapping by showing (cf. section 3.1.1 and 3.2.1) that they have different properties, the most fundamental difference being that the incomplete syntax of gapping is characterised by the absence of a functor, the verb, whereas VP ellipsis is characterised by the absence of an argument, the verb phrase. Having drawn this distinction, I then proceed to develop a ucg(2) analysis of each type of ellipsis. In the case of gapping, I answer the first question (what is the category of the e-clause?) by augmenting the set of rules with the product operation thus allowing gapped clauses to be assigned a product category that is, a category ranging over sequences of maximal projections. The second question (how is the e-clause licensed?) is answered by introducing a licensing principle which in essence requires that the categories associated with gapping remnants be substituted within the derivation tree of the a-clause in such a way that the resulting tree is derivable under ucg(2). VPE is treated in a very different fashion, whereby the problem of assigning a category to the VPE clause is circumvented by introducing in the grammar two categories for auxiliaries: VP/VP which is used for non-elliptical sentences and VP which is used for VPE clauses. Licensing on the other hand, results from the requirement that a VP sign be available from the linguistic context which unifies with the value of a new attribute, dubbed disgap.

It is worth noting that in each case (VPE and gapping) the licensing of the e-clause involves ucg(2) signs that is, multi-level information involving both syntax and semantics. Thus in the case of gapping, licensing involves the derivation tree of the a-clause whose nodes are labelled with ucg(2) signs whereas in the case of VPE, licensing depends on the presence in the linguistic context of a ucg(2) sign of category VP. As we shall see in the next chapter, this characteristic
can be put to good use when dealing with the interpretation of ellipsis. That is, the semantic information associated with the licensing material (also called the antecedent) may be used to determine the semantic interpretation of the ellipsis. Chapter 4 shows how this can be done.
Chapter 4

Semantics I

Just as the precedent chapter investigates the syntax of verbal and VP ellipsis, this chapter is concerned with their semantics. Here, three main questions arise. First, how is the antecedent identified? Second, what is the relationship between the semantics of this antecedent and that of the ellipsis? Third, how is the sloppy/strict ambiguity described in chapter 1 to be accounted for?

In the preceding chapter, we have seen that syntax plays a role in licensing VPE and gapping. Briefly, a VPE is licensed just in case a VP of the sort required by the auxiliary in the e-clause is part of the derivation tree of the a-clause. Gapping, on the other hand, is licensed by a comparison of the gapping remnants with the derivation tree of the a-clause: roughly, the a-clause provides the syntactic items which the e-clause is missing to form a fully fledged sentence. But while syntax plays an important role in licensing elliptical constructs, my claim is that it is also crucial in determining the antecedent of the ellipsis. In this chapter, I defend the view that the missing semantics in the elliptical construct can be reconstructed from the semantics of these items in the a-clause which syntactically license the ellipsis — I will refer to these licensing items as the antecedent of the ellipsis.

The next question to be answered concerns the way in which the semantics of an elliptical construct relates to the semantics of its antecedent. With regard to pronominal anaphors, a traditional classification consists in distinguishing between coreferential and bound-variable anaphors. Whereas a coreferential anaphor denotes the same object as its antecedent, a bound
variable anaphor acts as a logical variable — it denotes an arbitrary rather than a particular individual and is bound by some logical operator, usually a quantifier. The distinction between the two is illustrated by the following examples.

(4.1) a. Jon said that he likes Sarah

b. In Barbapapaland, no man shaves himself

In (4.1a), the pronoun he is coreferential with Jon in that both denote a single individual named Jon. By contrast, the reflexive himself in (4.1b) does not denote any particular individual but some arbitrary individual z of which it is said that in Barbapapaland, no z is a man that shaves z. More generally, reflexives are usually taken to be bound-variable anaphors whereas other pronominal anaphors can be either coreferential or bound.

Given this classification, the question arises of whether elliptical constructs such as VPE and gapping fall in one or the other of the two classes. In this chapter, I argue that neither is the case, but rather that the semantics of elliptical constructs is a function $\mathcal{R}$ of the semantics of the antecedent. I define this function to be a multi-valued function\(^1\) which yields as many values as there are possible readings for the elliptical constructs (this caters in particular for the sloppy/strict ambiguity described in chapter 1); $\mathcal{R}$ also ensures that the semantics obtained for the e-clause is the correct one.

Finally, the issue of how sloppy identity is to be accounted for, is discussed at length in section 4.2: the proposal is that sloppy identity is only possible if parallelism obtains between a- and e-clause.

The outline of the chapter is as follows. In section 4.1, I argue that the meaning of an elliptical construct is a function of the meaning of its antecedent and then go on to describe how this antecedent is to be recovered. These two ideas are then implemented within a proposal which precisely specifies how VPE and gapped clauses are to be interpreted within the present grammar. Section 4.2 builds on this first proposal, and considers the phenomenon of sloppy identity. In this section, an analysis of sloppy identity is given which is then integrated in the framework developed in section 4.1.

\(^1\)A multi-valued function is a function whose values are sets rather than single values.
4.1 Semantics of elliptical constructs

A common characteristic of elliptical clauses involving either VPE or gapping is that (i) their semantics is under-defined and (ii) the missing semantics may be recovered from some other clause in the linguistic context, the a-clause. As a first approximation, the interpretation of elliptical constructs may be defined as follows:

The semantics of an elliptical clause involving either VPE or gapping is a function of the semantics of the e-constituents and of some unknown semantics which needs to be recovered from the a-clause.

or more formally\(^2\),

\[ A1. \ ESem = f(R, e_1, \ldots, e_n) \]

where \( ESem \) denotes the semantics of the e-clause, \( R \) is the unknown semantics (also referred to here as the \textit{missing semantics}), \((e_1, \ldots, e_n)\) represents the semantics of the e-constituents and \( f \) is the function determining the interpretation of the e-clause. Suppose that we take \( f \) to be functional application. Then assumption \( A1 \) is modified to \( A1' \) and the main problem in defining the interpretation of an e-clause resides in defining the missing semantics \( R \).

\[ A1'. \ ESem = R(e_1) \ldots (e_n)^3 \]

Now recall that VPE and gapping are characterised by the fact that both their syntax and their semantics is under-specified\(^4\). This suggests that perhaps the missing semantics can be recovered from the missing syntax. That is, perhaps the missing semantics simply is the semantics of this item in the a-clause which syntactically licenses the ellipsis. By way of example consider the following sentences.

\(^2\)I assume here that meanings of phrases are to be represented by terms of a typed higher-order system with lambda-abstraction. Equality between two terms denotes \( \alpha \beta \eta \) interconvertibility (see [Hindley/Seldin, 1986]).

\(^3\)Contrary to what the notation suggests, note that I am not making any claim about the order in which the missing semantics \( R \) and the semantics of the e-constituents combine. This essentially depends on the details of the semantics and of the grammar and is not our concern here.

\(^4\)In fact, this is what I take to be the essential difference between anaphors and ellipses: whereas sentences including ellipses are under-specified both syntactically and semantically, sentences involving anaphors are only semantically underspecified.
In (4.2a), the VP ellipsis in the second conjunct is licensed by the VP *likes Mary* occurring in the a-clause. Furthermore, the semantics $\lambda z.\text{like}(z, \text{mary})^5$ of this VP can be seen as providing the missing semantics of the e-clause. Similarly, in (4.2b) the constituent providing the missing semantics for the e-clause is the verb *likes* with semantics $\lambda y.\lambda z.\text{like}(z, y)$. Thus in general, the unknown semantics in the e-clause can be seen to be related to the semantics of the item(s) in the a-clause which syntactically license the ellipsis. More precisely, the missing semantics of e-clauses can be defined as follows. In the case of VPE, there is only one missing constituent which can be of category VP (simple case), VP/NP (pseudo-gapping) or VP[NP] (antecedent-contained deletion), and the semantics of the elided VP is provided by the semantics of some appropriate antecedent. In the case of gapping, there can be one or more missing constituent(s) and the licensing antecedent(s) are determined by the tree-building rule for sentences involving gapped clauses: they are those constituents in the reconstructed tree of the e-clause which are not associated with the remnants of this e-clause (that is, they are a-constituents which have no e-correspondents). The semantics of the e-clause is simply the semantics of the reconstructed tree where missing constituents contribute their semantics.

So far, we have made the following assumptions:

$A1'$. $ESem = R(e_1) \ldots (e_n)$

$A2$. The missing semantics $R$ is the semantics of those syntactic items in the a-clause which license the ellipsis in the e-clause. In the case of VPE, this antecedent is either a VP, a VP/NP or a VP[NP]. In the case of gapping, the antecedent(s) is (are) determined by the tree-reconstruction operation.

But while this first sketch gives an idea of how the semantics of elliptical constructs can be recovered, it is unacceptable for two reasons. First, it does not account for the interaction of ellipsis with pronominal anaphora: that is, the sloppy/strict ambiguity of pronouns cannot be accounted for. Second, it yields an inappropriate semantics for the e-clause. First, let us examine the sloppy/strict ambiguity problem. Consider the sentence:

$^5$For expository purposes, the semantics has been simplified here. I defer to the end of this section the definition of the exact semantics assigned to the e-clause.
Jeremy fed his crocodile and Sarah did too.

Clearly the first clause is unambiguous: it means that Jeremy fed Jeremy's crocodile. However the second clause, as already observed, is ambiguous between Sarah fed Sarah's crocodile and Sarah fed Jeremy's crocodile. Since I claim that the first clause is unambiguous, the ambiguity in the second clause is best reflected in the semantics of the e-clause. Recall that I start from the assumption that the meaning of sentences involving elliptical constructs is a function of the meaning of its constituents and of the semantics of the antecedent. Now if the interpretation procedure is to account for the sloppy/strict distinction, it is necessary that this function be multi-valued. So rather than assuming A1', I make the following assumption:

A1''. \( E\text{Sem} = (f(R)) (e_1) \ldots (e_n) \) where \( f \) is multi-valued and yields as many values as there are ambiguities in the interpretation of the e-clause⁶.

The second problem in identifying the missing semantics of the e-clause with the semantics of part of the a-clause is that the resulting semantics for the e-clause is simply incorrect. Consider the following sentence for example.

(4.4) Jeremy fed a crocodile and Sarah did too.

Two interpretations are possible: either Jeremy and Sarah feed the same crocodile or they feed a different crocodile each. On the first reading, \( a \text{ crocodile} \) has wide scope and the meaning of the antecedent VP is \( \lambda x. \text{feed}(x, y) \) where \( y \) is the variable introduced by the quantifier \( a \text{ crocodile} \). This case is unproblematic. On the second reading however, \( a \text{ crocodile} \) will have narrow scope and the meaning of the antecedent VP will be \( \lambda x. \exists y (y/d_2)[\text{crocodile}(y) \land \text{feed}(x, y)] \). Now a problem arises. This problem has to do with the semantics of discourse markers. Recall (cf. chapter 2, section 2.3) that a discourse marker denotes a function from states into individuals. Furthermore transitions from states to states are triggered by state-switchers which themselves are introduced by NPs. More precisely, a state-switcher updates the domain of the assignment function with a new discourse marker. Consider now the interpretation of the whole discourse in (4.4) with translation:

\[
{(j/d_1) \exists y (y/d_2)[\text{crocodile}(y) \land \text{feed}(j, y) \land (s/d_3) \exists y (y/d_3)[\text{crocodile}(y) \land \text{feed}(s, y)]}
\]

⁶The details of how this function is to be defined are discussed in the next section.
In this formula, four state-switchers occur. Thus the interpretation of the different parts of the formula will depend on five different states. Let us call these states $s_0, s_1, s_2, s_3, s_4$ where $s_0$ is the initial state. The first state-switcher in the formula induces a transition to $s_1$ at which the interpretation of $d_1$ is defined. Similarly, the second state-switcher induces a transition to $s_2$ on which both $d_1$ and $d_2$ are defined while the third state-switcher induces a transition to $s_3$ on which $d_1, d_2$ and $d_3$ are defined. The problem occurs when the fourth state-switcher is evaluated.

As mentioned above, the semantics of state-switchers is to update the function which assigns a meaning to discourse markers with a new binding. But what happens if a state-switcher introduces a binding which is already in this function? There are three possibilities. The first is to assume that the new binding is in fact not a new binding and to consider that the semantics of the state-switcher in such a case is void. The second is to assume that the old binding already present in the function which assigns individuals to discourse markers is replaced by the new one. Both approaches are problematic in that they entail the loss of some discourse information, namely that the first occurrence of a crocodile introduces a discourse marker in the discourse model. This is shown to be inadequate by the following example:

(4.5) Jeremy fed a crocodile and Sarah did too. Both have long green tails, white long teeth and a very hungry stomach.

In the second sentence, both refers to the two crocodiles — but if the information about the first crocodile has disappeared it is unclear how such cases can be dealt with. Finally, the third solution is to somehow forbid formulae in which the same discourse marker is introduced by more than one state-switcher. With respect to elliptical constructs this means that we cannot just use the translation of their antecedents to compute their interpretation. Rather, elliptical constructs will be assigned as meaning representation an alphabetic variant of the translation associated with the antecedent. An alphabetic variant $\Phi'$ of a formula $\Phi$ is identical to $\Phi$ up to renaming of bound variables and discourse markers — since all discourse markers\(^7\) of the antecedent are renamed in the semantic representation of the elliptical clause, this guarantees that no two state-switchers mention the same discourse marker, which is just as required.

The analysis I propose relies on the idea that the meaning of an elliptical construct is a function of the meaning of its antecedent. Call this function $R$ (for Resolution). $R$ has two roles: first it creates an alphabetic variant of the antecedent translation and second, it ensures that any pronouns contained in the antecedent may be resolved both as sloppy and as strict if both cases

\(^7\)Strictly speaking, this is inaccurate. As we shall see in section 4.2, not all discourse markers should be renamed — only those that do not represent pronouns should.
are possible (if no sloppy reading is available, only the strict interpretation is produced). The precise definition of \( R \) is given in section 4.2 where sloppy identity is dealt with.

To summarise, the analysis I propose for determining the semantics of elliptical constructs relies on the following assumptions.

A1. \( E_\text{Sem} = (R(R))(e_1) \ldots (e_n) \) where \( R \) (i) creates an alphabetic variant of \( R \) and (ii) yields as many values as there are ambiguities in the interpretation of the elliptical construct denoting \( R \).

A2. As above.

I now turn to the question of how the grammar enforces an interpretation of elliptical constructs along the lines just described.

Consider VPE first. How is the semantics of a verb phrase ellipsis obtained in our grammar? Here several factors need to be taken into account. In chapter 5, I argue that VPE resolution occurs at discourse level: the antecedent clause of a clause containing a VPE must be connected by some discourse relation to the e-clause. In the sequel, I assume that the a-clause has been identified by the discourse grammar and that its derivation tree is available when resolution occurs. In chapter 3, I require that some of the attribute values of the sign associated with the VPE auxiliary unify with the corresponding attribute values in the sign associated with the antecedent VP. We now make the following amendments to this requirement. Instead of requiring that the semantics of the antecedent unifies with that of the ellipsis, we now state that if \( \Phi \) is the semantics of the antecedent VP then \( R(\Phi) \) is the semantics of the ellipsis. As a result, the semantics of the e-clause now is \( \Psi(R(\Phi)) \) where \( \Psi \) is the semantics of the subject NP\(^8\) and \( R \) is as defined in section 4.2. For instance, suppose we have the sentence:

\[(4.6) \quad \text{Jon fed a crocodile and Peter did too.}\]

Then the semantics of the antecedent VP \( \text{fed a crocodile} \) will be \( \lambda x. \exists y [y/d_2] [\text{crocodile}(y) \wedge \text{feed}(x,y)] \). The semantics \( \Phi \) of the e-clause will then be determined as follows. First assume that the semantics \( R(\Phi) \) of the VPE in the e-clause is:\(^9\)

\(^8\)Or \( \Psi(\Delta(R(\Phi))) \) in the case of pseudo-gapping, with \( \Delta \) the semantics of the remaining object or indirect object.

\(^9\)The definition of \( R \) is given in section 4.2.
\[ \mathcal{R}(\Phi) = \lambda x'. \exists y' (y'/d_2') [\text{crocodile}(y') \land \text{feed}(x', y')] \]

To obtain the semantics of the e-clause, the semantics \( \Psi \) of the subject NP is then applied to \( \mathcal{R}(\Phi) \). Let us assume that the semantics of the subject NP, \( \text{Peter} \) is \( \lambda P_1 \cdot \{p/d_3\} P_1(p) \), then by \( \beta \)-conversion, we have that the semantics \( \Psi(\mathcal{R}(\Psi)) \) of the e-clause is:

\[
\lambda P_1 \cdot \{p/d_3\} P_1(p) \ (\lambda x'. \exists y' (y'/d_2') [\text{crocodile}(y') \land \text{feed}(x', y')])
= \{p/d_3\} \exists y' (y'/d_2') [\text{crocodile}(y') \land \text{feed}(p, y')] \]

Let us now turn to gapping. Reconstruction of the derivation tree for the e-clause yields a mother node whose semantics can be thought of as the resolved semantics of the e-clause. However, for the correct results to be obtained, \( \mathcal{R} \) must first be applied to the semantics in the a-clause which contributes to resolve the semantics of the e-clause. In order to do so, the semantics of those elements in the reconstructed tree which are contributed by the a-clause must be a function of \( \mathcal{R} \). This can be achieved if the definition of substitution given in chapter 3, section 3.1.2 is appropriately modified that is, if whenever a constituent in the a-tree is not substituted for (i.e., it is a missing constituent in the e-clause), a modified copy of this constituent is returned by substitution such that if the semantics of this constituent is \( \Phi \) in the a-clause, then it becomes \( \mathcal{R}(\Phi) \) in the e-clause. So clauses (32) and (33) of Definition 3.4:

\[ \sigma_1((r, \text{Product})) = (r, \text{Product}) \]

becomes:

\[ \sigma((r, \text{Product})) = (r', \text{Product}) \]

where \( r' \) is a sign just like \( r \) except for the semantics: if \( \phi \) is the semantics of \( r \), then \( \mathcal{R}(\phi) \) is the semantics of \( r' \). For instance, consider the sentence in (4.7).

(4.7) Jon likes Sarah and Peter Rose.

Assume that the semantics of \( \text{likes} \) in the a-clause is \( \Phi = \lambda \mathcal{R}_2 \lambda z (\mathcal{R}_2(\lambda y. \text{like}(z, y))) \). Then the missing semantics \( \mathcal{R}(\Phi) \) in the e-clause is:

\[ \mathcal{R}(\Phi) = \lambda \mathcal{R}_2' \lambda z' (\mathcal{R}_2'(\lambda y'. \text{like}(z', y'))) \]
Assume further that the semantics of the two gapping remnants Peter and Rose is \( \Delta = \lambda P_1.\{p/d_3\} P_1(p) \) and \( \Psi = \lambda P_3.\{r/d_4\} P_3(r) \) respectively. Then the resolved semantics of the e-clause will be \( \Delta(\mathcal{R}(\Phi)(\Psi)) \) which by \( \beta \)-conversion is equivalent to:

\[
\lambda P_1.\{p/d_3\} P_1(p)(\mathcal{R} (\lambda x^{'}. \lambda y^{'}. \text{like}(x^{'}, y^{'})))(\lambda P_3.\{r/d_4\} P_3(r))
\]

\[
= \{p/d_3\} \{ r / d_4\} \text{like}(p, r)
\]

4.2 Sloppy identity

As already mentioned, pronouns which occur in the antecedent of the elliptical construct often give rise to a sloppy/strict ambiguity in the interpretation of the e-clause. In this section, I investigate the interaction of VP and verbal ellipsis with pronouns. First, I summarise some well-known properties of sloppy pronouns (cf. section 4.2.1). Given these properties, some obvious questions arise about the interpretation of sloppy pronouns: why (and how) is the antecedent of the sloppy pronouns uniquely determined? Why are sloppy pronouns less sensitive to the morpho-syntactic features of their antecedent than 'ordinary pronouns'? Where does the sloppy/strict ambiguity stem from? Or put another way, which of the antecedent and the elliptical construct is ambiguous? In section 4.2.2, I propose an analysis of sloppy pronouns which answers these questions. This analysis is based on the idea that parallelism is a key notion in accounting for sloppy identity and shows how the constraints regulating the interpretation of sloppy pronouns interact with the function \( \mathcal{R} \) to determine the interpretation of elliptical constructs. Section 4.2.3 compares the present approach with existing analyses and section 4.2.4 discusses some remaining puzzles.

4.2.1 Properties of sloppy pronouns

Here a caveat is in order concerning the terminology being used. Throughout this thesis, the term 'sloppy pronoun' is employed. Now strictly speaking and given my commitment to concrete syntax, this is inaccurate simply because there is no such thing as a sloppy pronoun. That is, there is no pronoun in the e-clause which could take a sloppy interpretation. Rather, there is an e-clause whose interpretation may involve the interpretive reconstruction of a pronoun which in turn may result in the interpretation of this clause being either strict or sloppy. For simplicity however, I will continue to use the term 'sloppy pronoun' to refer to the interpretive
reconstruction of a pronoun which leads to the e-clause being assigned a sloppy interpretation. Furthermore, in referring to the various parts of discourse involved in the creation of a sloppy/strict ambiguity, I will generally adopt a transformational terminology. Thus a pronoun in the a-clause which triggers a sloppy/strict ambiguity in the interpretation of the e-clause is called a trigger pronoun. Moreover, the antecedent of a trigger pronoun will be called trigger antecedent whilst the antecedent of a target pronoun will be referred to as target antecedent. For instance, in *Harold scratched his arm and Jon did too*, *Harold* is the trigger antecedent, *his* the trigger pronoun and *Jon* the target antecedent.

As already mentioned in chapter 1, the existence of sloppy pronouns was first observed in [Ross, 1967] and is illustrated by the following example:

(4.8) Harold scratched his arm and Jon did too.

Ross observed that such a sentence has two possible readings. A first reading is one where *Harold scratched Harold's arm and Jon scratched Harold's arm*; it is called the *strict* reading and corresponds to the case where the target pronoun in the e-clause remains anaphorically related to the trigger antecedent in the a-clause. The second reading is: *Harold scratched Harold's arm and Jon scratched Jon's arm*; this is the *sloppy* reading and corresponds to having the target pronoun being anaphorically related to some NP in the e-clause which I will call the target antecedent. Although the example given above involves a case of VP ellipsis, sloppy pronouns appear in many other types of environment as illustrated by the examples in (4.9), taken from [Reinhart 1983]. As we shall see in section 4.2.2, one of the characteristics of the analysis I propose is that it extends more easily to such cases of sloppy identity than most other existing analyses.

(4.9) a. I told you that you would be famous and Jack told Betty. (Null complement anaphora)

b. Zelda bought Siegfried a present on his wedding day and Felix too. (Gapping)

c. My father has never shot himself with a gun but my grandfather did it in 1984. (*Do it* anaphora)

d. Jack believes that he is losing his hair and Peter believes it too. (*It* anaphora)

e. Jon likes more of his children than Bill hates. (Comparative deletion)

f. Harry found a place to park his car before Mary found one. (*One* anaphora)
g. Jon always gives his profits to overseas aid but Sam uses them to expand his business. (Pronoun of laziness)

A second property of sloppy pronouns is that they are less sensitive to the gender, number and person of their antecedent than ordinary pronouns. Consider gender first.

\[(4.10) \quad \text{Jon washes his car and Sarah does } \theta \text{ too.} \]
\[\theta = \text{washes her car}\]

Although the trigger pronoun has masculine gender, the sloppy interpretation of the e-clause allows the target pronoun to be anaphorically linked with an NP which has feminine gender. This sloppiness extends to reflexives and generalises to all cases of sloppiness, in particular to those involving gapping, as shown in (4.11) below.  

\[(4.11) \quad \text{a. Jon told Sarah about himself and Mary did } \theta \text{ too.} \]
\[\theta = \text{Mary told Sarah about herself}\]
\[\text{b. Jon showed Mary his portrait and Sarah } \theta \text{ Paul.} \]
\[\theta = \text{Sarah showed Paul her portrait}\]

Sloppy pronouns can also be insensitive to number, as exemplified by the following sentences.  

\[(4.12) \quad \text{a. Jon and Mary washed their car and Paul did } \theta \text{ too.} \]
\[\theta = \text{Paul washed his car}\]
\[\text{b. Jon and Mary showed Frank pictures of themselves and Peter did } \theta \text{ too.} \]
\[\theta = \text{Peter showed Frank pictures of himself}\]

\[10\text{But note that Jon hit himself, and Mary did too is odd. That is, sloppy pronouns are not always insensitive to the gender of the their antecedent. In the case of reflexives, it seems that some syntactic factor influences the acceptability of sloppy pronouns in that gender clash is more acceptable when the reflexive is not a direct argument but occurs as part of a prepositional phrase. Here I will assume that both the strict and the sloppy reading should be produced at the level of semantics and that impossible readings should be excluded by the interaction of semantics with other linguistic components such as syntax and pragmatics.}\]

\[11\text{Again, note that insensitivity to number varies. For instance, The boys washed themselves but Sue didn't is odd. That is, reflexive sloppy pronouns seem to be more sensitive to the gender of their antecedent than non-reflexives one.}\]
c. Jon and Mary showed Frank pictures of their kids and Paul showed Sarah pictures of his kids.

Person may also be ignored in the resolution of sloppy pronouns. The following examples illustrate.

\[(4.13)\]
\[a. \quad I \text{ showed Mary pictures of myself but Peter didn't.} \]
\[\varnothing = \text{show Mary picture of himself} \]
\[b. \quad I \text{ showed Mary pictures of myself and Peter Sarah.} \]
\[\varnothing = \text{Peter showed Sarah pictures of himself} \]

Finally note that sloppy pronouns do not necessarily retain the anaphoric status (bound or coreferential) of the pronoun in the a-clause: the latter may be bound and the former referential or vice-versa as shown in (4.14) below.

\[(4.14)\]
\[a. \quad \text{Every boy said that Mary kissed him. Peter said that she did, Jon said that she did and Paul said that she did. ([Asher 1990])} \]
\[b. \quad \text{Although Jon washed his car, no one else did.} \]

As already mentioned in the introduction several questions naturally arise from these observations about sloppy pronouns. First, why is the e-clause anaphorically ambiguous? That is, why does the pronoun contained in the a-clause become ambiguous when interpreted in the e-clause. Second, how is the sloppy reading of this pronoun determined? That is, why is there no problem in identifying its new antecedent? Third, why are sloppy pronouns insensitive to the morphological properties of the NP to which they are anaphorically related? The next section sketches an analysis of sloppy pronouns which answers these questions.

\[12\]Here again, reflexives seem to fare differently, as shown by the difficulty in getting a sloppy reading for There is something you know about yourself that I don't.
4.2.2 Analysis

General assumptions

The analysis I propose makes three main assumptions about the sloppy interpretation of elliptical clauses. The first and most basic assumption is that sloppy interpretations are only possible when parallelism\textsuperscript{13} obtains between a- and e-clause\textsuperscript{14}. For instance, in the sentence *Jeremy fed a crocodile and Sarah, a monkey*, the e-clause will be said to be parallel to the a-clause because the NP associated with *Sarah* is in some sense parallel to the NP associated with *Jeremy* while *a monkey* is parallel with *a crocodile* (in what follows, I will use the term e- and a-correspondent to designate constituents which are parallel to each other and belong to the e- and a-clause respectively). Drawing on this parallelism idea, I then make a second, more specific assumption which is that the antecedent of a sloppy pronoun is that NP in the e-clause which is parallel to the antecedent of the trigger pronoun in the a-clause. Finally, a third assumption I make (and which was already made explicit in section 4.2) is that the interpretation of sloppy pronouns is determined by the function $\mathcal{R}$ which defines the interpretation of elliptical constructs. These three basic assumptions can be summarised as follows:

A1. Sloppy interpretation only occurs when parallelism obtains between a- and e-clause. When parallelism obtains, a mapping $\mathcal{P} : NP_e \rightarrow NP_a$ can be defined from the NPs ($NP_e$) of the e-clause into those ($NP_a$) of the a-clause. Given a pair of the form $(x, \mathcal{P}(x))$, $x$ is said to be an e-correspondent and $\mathcal{P}(x)$ an a-correspondent.

A2. The antecedent of a sloppy pronoun is the e-correspondent of the antecedent of the trigger pronoun. That is, if $(x, y) \in \mathcal{P} : NP_e \rightarrow NP_a$ and $y$ is the antecedent of the trigger pronoun in the a-clause, then $x$ is the antecedent of the target pronoun.

A3. The interpretation of a sloppy pronoun depends on the function $\mathcal{R}$ which determines the interpretation of elliptical constructs.

Now two questions come to mind. First, do these assumptions make any predictions as to when sloppy pronouns occur? Second, how do these assumptions help us answer the questions raised

\textsuperscript{13}A precise definition of parallelism is given further on in this section.

\textsuperscript{14}Since by definition, e-clauses involve elliptical constructs, this proposal rules out pronouns of laziness and coreferents unless these are taken to be elliptical constructs too, which is not very plausible since they do not lack any syntactic material. How the ideas presented here might be extended to deal with such cases of sloppy identity would take us too far afield and I will leave the question open here.
at the end of the previous section?

Let us examine the first question first: does the proposed analysis make any prediction as to the possibility or impossibility of sloppy readings? There are two obvious predictions made here. The first is that if the antecedent of the trigger pronoun has no e-correspondent, then no sloppy reading is possible. The second is that crossed readings are impossible, where crossed reading are readings in which the antecedent of the sloppy pronoun is not parallel to the antecedent of the trigger pronoun (that is, the antecedent of the sloppy pronoun is not the e-correspondent of the antecedent of the trigger pronoun). Although these two facts about sloppy pronouns may appear obvious, I believe they are important characteristics which in fact point to the basic nature of sloppy pronouns: sloppy pronouns are pronouns which are subject to a strong parallelism constraint. The following examples illustrate the effect of the first prediction: if there is no e-correspondent, there is no sloppy reading.

(4.15)  

a. The person who introduced Mary to Jon refused to give him her phone number and so did the person who introduced Sue to Bill. (Strict/sloppy)  
b. The person who introduced Mary to Jon refused to give him her phone number and so did Peter. (Strict)  
c. The policeman who arrested Bill refused to read him his rights and so did the policeman who arrested Jon. (Strict/sloppy)  
d. The policeman who arrested Bill refused to read him his rights and so did the judge. (Strict)

In (4.15a) the antecedents Mary and Jon have as e-correspondents Sue and Bill respectively, so a sloppy reading of the trigger pronouns him and her is possible. Similarly in (4.15c), Bill has as e-correspondent Jon so a sloppy reading of him and his is possible. By contrast in (4.15b) and (4.15d), the antecedent NPs in the a-clause have no e-correspondents and no sloppy interpretation of the pronouns is available.

The second prediction is that there can be no crossed reading. The following examples illustrate.

(4.16)  

a. I told you₁ that you₁ would be famous and Jack told Betty 0.  

$0 = \text{that } (Betty/*Jack) \text{ would be famous}$
b. Norma told Beth₁'s boyfriend to give her₁ a dime and Judy told Lois' boyfriend to θ, also.

\[ \theta = \text{give (Lois/*Judy)} \]

c. Bill showed Frank₁ his₁ portrait and Carl Peter θ.

Carl [showed] Peter [(Peter's /*Carl's) portrait]

(4.16a) illustrates a case of Null Complement Anaphora in which the trigger pronoun is anaphorically linked to the object NP you. Under the present analysis, the sloppy pronoun in the e-clause must be anaphorically related to that NP in the e-clause which is parallel to the antecedent of the trigger pronoun in the a-clause, that is in this particular cases, to the object NP Betty. This prediction is confirmed by the data since (4.16a) can be interpreted as meaning either (4.17a) or (4.17b) but not (4.17c).

(4.17) a. I told you₁ that you₁ would be famous and Jack told Betty that you₁ would be famous.

b. I told you₁ that you₁ would be famous and Jack told Betty that Betty would be famous.

c. I told you₁ that you₁ would be famous and Jack told Betty that Jack would be famous.

Similarly under the reading indicated by the subscripts, (4.16b) can be interpreted as (4.18a), or as (4.18b), but not as (4.18c).

(4.18) a. Norma told Beth₁'s boyfriend to give her₁ a dime and Judy told Lois's boyfriend to give Beth a dime.

b. Norma told Beth₁'s boyfriend to give her₁ a dime and Judy told Lois' boyfriend to give Lois a dime.

c. Norma told Beth₁'s boyfriend to give her₁ a dime and Judy₂ told Lois' boyfriend to give Judy₂ a dime.

The same law holds for gapping: (4.16c) means either (4.19a), or (4.19b), but not (4.19c).

(4.19) a. Bill showed Frank₁ his₁ portrait and Carl showed Peter Peter's portrait.
b. Bill showed Frank₁ his₁ portrait and Carl showed Peter Frank’s portrait.

c. Bill showed Frank₁ his₁ portrait and Carl showed Peter Carl’s portrait.

Let us now come back to the second question raised above: How do these assumptions help us answer the questions raised at the end of the previous section? The questions to be answered are:

1. Why is the e-clause ambiguous?

2. How is the antecedent of the sloppy pronoun uniquely identified (and why are crossed readings ruled out)?

3. Why are sloppy pronouns less sensitive to the morphological characteristics of their antecedent NP than normal pronouns?

Now consider the assumptions we made. Do they help us giving an answer to those questions? Question (1) is at least partially answered by assumption A3. The reason for this is that as we have seen in section 4.1, the function \( R \) which determines the interpretation of elliptical constructs is taken to be a multi-valued function where different values correspond to different interpretations and in particular to the strict/sloppy distinction. This accounts for the fact that the e-clause may be ambiguous even though the a-clause may not be. Question (2) finds an answer in assumption A2 under which parallelism is taken to determine the antecedent of a sloppy pronoun: it is the e-correspondent of the antecedent of the trigger pronoun. This explains why the antecedent of a sloppy pronoun is uniquely identified and why there can be no crossed readings. Finally, assumption A3 gives some answer to question (3) in that if the interpretation of a sloppy pronoun depends on the function \( R \) determining the meaning of elliptical constructs, then clearly the resolution mechanism determining the interpretation of sloppy pronouns is distinct from that used for standard pronouns (whereas ‘standard pronouns’ are directly resolved to some antecedent NP, the interpretation of sloppy pronouns depends on the function \( R \)). In what follows, we shall see that unlike the resolution procedure for standard pronouns, the resolution mechanism for sloppy pronouns is in fact insensitive to the morpho-syntactic properties of the pronoun antecedent. This accounts for the fact that the morpho-syntactic properties of both pronoun and antecedent play no role in the resolution of sloppy pronouns.
Defining parallelism

The notion of parallelism on which the proposed account of sloppy pronouns is based must now be made precise. Although it has been shown that sloppy pronouns may occur in all sorts of environments, the present analysis will concentrate on only two of these possible environments: gapping and VP ellipsis. However the devices used are very general and it is not unreasonable to think that the described approach may extend to all cases of sloppy identity involving elliptical constructs. Whether this is in fact the case and what if any, are the modifications to be introduced in order to do so, will here remain a topic for further research.

So how is parallelism to be defined? At what level? Is the parallelism we need syntactic or semantic or perhaps both? Although it might be possible to define parallelism at the semantic level, it seems that this is asking for unnecessary complications: what we need when dealing with sloppy identity, is to establish the parallelism which must hold between the NPs of the a-clause on the one hand, and the NPs of the e-clause on the other. That is, we need to base parallelism on syntactic information. So, here I will assume that the notion of parallelism required to explain sloppy identity is syntactic rather than semantic — that is, parallelism will be defined on syntactic trees. Moreover, I will assume that only NPs need to be considered. This assumption is based on purely empirical considerations: sloppy pronouns get their interpretation from an NP which is parallel to the antecedent of the trigger pronoun in the a-clause — any other type of constituent is irrelevant in determining the interpretation of sloppy pronouns. Now we need to determine what makes two sentences be parallel in the sense that if the first clause contains a pronoun then the second might be assigned a sloppy interpretation (if it involves an ellipsis for instance). Here an example will be helpful. Consider the sentence in (4.15a) repeated here for convenience.

(4.20) \[ NP_1 \text{ The person who introduced } [NP_{11} \text{ Mary} ] \text{ to } [NP_{12} \text{ Jon} ] \text{ refused to give him}_12\text{ her}_11\text{ phone number and so did } [NP_3 \text{ the person who introduced } [NP_{31} \text{ Sue} ] \text{ to } [NP_{32} \text{ Bill} ]\].

Here the intuition is that \( NP_1 \) is parallel with \( NP_2 \), \( NP_{11} \) with \( NP_{21} \) and \( NP_{12} \) with \( NP_{22} \). This can be represented as in Figure (4.1).

In other words, parallelism can be thought of as a one-to-one mapping from the set of NPs in a given derivation tree to the set of NPs in another derivation tree such that this mapping preserves both dominance and precedence. That is, if \( NP_1 \) precedes/dominates \( NP_2 \) in the a-
Figure 4.1: Example of NP mapping between derivation trees.

tree then the e-correspondent of $NP_1$ must precede/dominates the e-correspondent of the $NP_2$.

The formal definition of parallelism runs as follows. First, I define precedence and dominance (these definitions are strongly inspired from [Wall 1972]).

**Definition 4.1 Dominance relation ($C_r$)**

Given two nodes $X$ and $Y$ in a tree $r$, then $X$ dominates $Y$ relative to $r$ written $X C_r Y$, if there is a connected sequence of branches in the tree extending from $X$ to $Y$ and all the branches in the sequence have the same orientation away from $X$ and towards $Y$\(^{15}\). The set of all pairs $(X,Y)$ such that $X C_r Y$ is said to constitute the dominance relation for $r$. The dominance relation defines a weak partial ordering on the nodes of the tree (that is, it is transitive, reflexive and antisymmetric).

**Definition 4.2 Precedence relation ($<_r$)**

Given two nodes $X$ and $Y$ in a tree $r$, then $X$ precedes $Y$ relative to $r$, written $X <_r Y$, iff $X$ does not dominate $Y$ relative to $r$ and $X$ is to the left of $Y$. The set of all pairs $(X,Y)$ such that $X <_r Y$ is said to constitute the precedence relation for $r$. This precedence relation defines a strict partial ordering on the nodes of the tree (that is, it is transitive, irreflexive and asymmetric).

Given these two relations $C_r$ and $<_r$, parallelism can now be defined as follows.

\(^{15}\)For a precise definition of the terms nodes, tree and branches, see [Wall 1972:144].
Definition 4.3 Parallelism between derivation trees (P)
Let \( \tau_1 \) and \( \tau_2 \) be two derivation trees and \( NP_{\tau_1} \), \( NP_{\tau_2} \) be the sets of NPs contained in \( \tau_1 \) and \( \tau_2 \) respectively. Then \( \tau_1 \) is parallel to \( \tau_2 \) iff there is a relation preserving mapping \( P : NP_{\tau_1} \rightarrow NP_{\tau_2} \). That is,
\[
\forall x, y \in NP_{\tau_1} [x \subset_1 \tau_1, y \Rightarrow P(x) \subset_1 \tau_2, P(y)]
\]
and
\[
\forall x, y \in NP_{\tau_1} [x \subset_2 \tau_1, y \Rightarrow P(x) \subset_2 \tau_2, P(y)]
\]
\( P \) is an NP mapping. The correspondent of any NP in \( \tau_1 \) is \( P(NP) \) in \( \tau_2 \). A correspondent in an a-clause is said to be an a-correspondent whereas a correspondent in an e-clause is an e-correspondent.

The mapping \( P \) which holds between two parallel derivation trees induces a mapping \( D \) between discourse markers. So if \( P(NP_i) = NP_j \) and \( NP_i \) introduces the discourse marker \( d_i \) while \( NP_j \) introduces\(^{16} \) the discourse marker \( d_j \) then \( D \) maps \( d_i \) into \( d_j \). More generally:

Definition 4.4 Parallelism between discourse markers (D)
Let \( \tau_1 \) and \( \tau_2 \) be two parallel derivation trees such that \( P : NP_{\tau_1} \rightarrow NP_{\tau_2} \) is an NP mapping from \( \tau_1 \) into \( \tau_2 \). Let \( NP_{\tau_1} \), \( NP_{\tau_2} \), \( DM_{\tau_1} \), \( DM_{\tau_2} \) be the sets of NPs and DMs contained in \( \tau_1 \) and \( \tau_2 \) respectively. For all \( NP_k \), let \( d_k \) be the discourse marker introduced by \( NP_k \). Then \( D : DM_{\tau_1} \rightarrow DM_{\tau_2} \) is a DM mapping relative to \( \tau_1 \) into \( \tau_2 \) iff \( D = \{(d_i, d_j) | (NP_i, NP_j) \in P\} \).

Resolving elliptical constructs

At the beginning of this chapter, I claim that sloppy readings are constrained by parallelism. Having defined parallelism, I now proceed to show how sloppy readings obtain. I start with an example. Consider the sentence in (4.21) with derivation trees as in Figure 4.2 (where for each NP, the phonology and the discourse marker introduced by this NP is indicated. The arrows represent the NP and the DM mappings which can be defined from the e-tree into the a-tree).

\(^{16}\)Given a set \( A \) and a relation \( R_A \) in \( A \) and a set \( B \) and a relation \( R_B \) in \( B \), a function \( f : A \rightarrow B \) is a relation preserving mapping iff \( f \) is a one-to-one correspondence such that for all \( x, y, (x, y) \in R_A \Rightarrow (f(x), f(y)) \in R_B \).

\(^{17}\)I leave here the precise definition of an NP introducing a discourse marker undefined. The intuition is that an NP introduces a discourse marker \( d_i \) iff it is a non-pronominal NP which modify the current context of evaluation with a binding indicating the value of \( d_i \) (cf. section 2.3).
(4.21) Jon$_1$ [$_2$ washes his$_1$ panda] and Peters$_3$ does 0$_2$ too.

\[ \text{NP} \quad \text{VP} \quad \text{NP} \quad \text{VP} \]

S

\[ \text{NP} \quad \text{P} \quad \text{does too} \quad \text{NP} \quad \text{Jon} \quad \text{d1} \]

S

\[ \text{VP} \quad \text{washes} \quad \text{TVP} \quad \text{NP} \quad \text{his panda} \quad \text{d2} \]

Figure 4.2: Parallelism between derivation trees.

As indicated in section 4.1, the interpretation of a VP ellipsis is $\mathcal{R}(\Phi)$ where $\Phi$ is the semantic representation associated with the antecedent VP (here, *washes his panda*). Given the sloppy/strict ambiguity of the e-clause, $\mathcal{R}(\Phi)$ must return two values: one for the sloppy reading and one for the strict reading. To achieve this, the parallelism holding between e- and a-clause must first be established. This is carried out by defining first, $\mathcal{P} : NP_e \rightarrow NP_a$ and second, the derived mapping $\mathcal{D} : DM_e \rightarrow DM_a$. $\mathcal{D}^{-1} : DM_a \rightarrow DM_e$ is also defined where $DM_a$ is the range$^{18}$ of $\mathcal{D}$ and $\mathcal{D}^{-1}$ is the inverse function of $\mathcal{D} : DM_e \rightarrow DM_a$. Intuitively, $\mathcal{D}^{-1}$ is a mapping which returns for any discourse marker occurring in the a-clause the discourse marker associated with the e-correspondent (for instance, in (4.21), $\mathcal{D}^{-1}(d_1) = d_3$, that is, the discourse marker associated with Jon is mapped into the discourse marker associated with its e-correspondent Peter). So in the present case, the following holds.

\[ \mathcal{P} : \{NP(Peter)\} \rightarrow \{NP(Jon), NP(hispanda)\} \text{ and } \mathcal{P} = \{(NP(Peter), NP(Jon))\} \]

\[ \mathcal{D} : \{d_3\} \rightarrow \{d_1, d_2\} \text{ and } \mathcal{D} = \{(d_3, d_1)\} \]

\[ \mathcal{D}^{-1} : \{d_1\} \rightarrow \{d_3\} \text{ and } \mathcal{D}^{-1} = \{(d_1, d_3)\} \]

$\mathcal{R}$ makes then use of $\mathcal{D}^{-1}$ to determine the value of discourse marker. Here the intuition is this. Given a trigger pronoun represented by the discourse marker $d_1$, then on the strict reading, $\mathcal{R}$ returns the same discourse marker i.e., $d_1$ whereas on the sloppy reading, $\mathcal{R}$ returns as value the discourse marker introduced by the e-correspondent of the antecedent of the trigger pronoun i.e., $d_3$.

$^{18}$The range of a function $f$, written $\text{ran } f$, is: $\text{ran } f = \{y \mid (\exists x)(x, y) \in f\}$.
So in the example (4.21) above, this means that $R$ will return as value for the semantic representation $d_1$ of the trigger pronoun two values i.e., $d_1$ and $d_3$. Thus, the application of $R$ to the semantic representation of the antecedent VP *washes his panda* will yield two, rather than one, representations: one in which the pronoun *his* is represented by the discourse marker $d_1$ introduced by *Jon* and the other, where *his* is represented by the discourse marker $d_3$ introduced by *Peter*. 

More generally, $R$ should be defined in such a way that on the strict reading, a pronoun will be represented by the discourse marker introduced by the antecedent of the trigger pronoun whereas on the sloppy reading, it will be represented by the discourse marker introduced by the e-correspondent of this antecedent. Recall that $R$ also creates an alphabetic variant of the semantic representation associated with the a-clause. Given these observations, the resolution function $R$ is defined as follows$^{19}$:

Definition 4.5 $R$

$R$ is a type-preserving function which maps DIL formulae into DIL formulae. It is recursively defined on the structure of DIL formulae. Let $\tau_e$ be the derivation tree of an elliptical clause and $\tau_a$ be the tree of the corresponding a-clause. Let $D$ be a DM mapping between $\tau_e$ and $\tau_a$. Then $R$ is defined as follows.

---

$^{19}$Note that the fourth clause of the definition is not recursive. The reason for this is that there are essentially three ways of treating a discourse marker: either it represents an NP and then it should be renamed, or it stands for a sloppy pronoun and should be replaced by the discourse marker introduced by the target antecedent or it stands for a strict pronoun and should be copied untouched. Within the present version of DIL there is no elegant way of accounting for these differences other than the one I propose. Perhaps a more elegant route would be to distinguish in the semantic representation language between pronouns and non-pronominal NPs. Such an approach is actually defended in [Van der Does 1990] on the ground that there is still much debate as to whether all pronouns should be treated as bound variables. The argument is that as long as this controversy has not been solved, it is best to adopt an approach which does differentiate NPs from pronouns.
\( \mathcal{R}(d_i) = d_i \) or \( d_j \) if \( D^{-1}(d_i) = d_j \)
\( = d_i \) if \( D^{-1} \) is undefined on \( d_i \).
\( \mathcal{R}(c) = c \) if \( c \in \text{CON} \)
\( \mathcal{R}(x) = x' \) if \( x \in \text{VAR} \) and \( x \neq x' \)
\( \mathcal{R}(\{t/d\}) = \{ \mathcal{R}(t)/d' \} \)
\( \mathcal{R}(\phi \land \psi) = \mathcal{R}(\phi) \land \mathcal{R}(\psi) \)
\( \mathcal{R}(\phi \lor \psi) = \mathcal{R}(\phi) \lor \mathcal{R}(\psi) \)
\( \mathcal{R}(\phi \rightarrow \psi) = \mathcal{R}(\phi) \rightarrow \mathcal{R}(\psi) \)
\( \mathcal{R}(\neg \phi) = \neg \mathcal{R}(\phi) \)
\( \mathcal{R}(\exists x \phi) = \exists \mathcal{R}(x)\mathcal{R}(\phi) \)
\( \mathcal{R}(\forall x \phi) = \forall \mathcal{R}(x)\mathcal{R}(\phi) \)
\( \mathcal{R}(\lambda x \phi) = \lambda \mathcal{R}(x)\mathcal{R}(\phi) \)
\( \mathcal{R}(R_n(t_1 \ldots t_n)) = \mathcal{R}(R_n)\mathcal{R}(t_1) \ldots \mathcal{R}(t_n) \)
\( \mathcal{R}(\{t/d\}) = \mathcal{R}(\{t/d\})\mathcal{R}(\phi) \)
\( \mathcal{R}(\neg \phi) = \neg \mathcal{R}(\phi) \)
\( \mathcal{R}(\forall x \phi) = \forall \mathcal{R}(x)\mathcal{R}(\phi) \)

The first clause caters for the strict/sloppy distinction, the second clause ensures that constants are copied over and the third one guarantees that variables are primed so as to produce an alphabetic variant of the translation of the antecedent. The remaining clauses ensure that \( \mathcal{R} \) recursively applies to all subformulae contained within the input formula. To clarify the workings of the present analysis, I will now run through the example in (4.21). Given Definition 4.5, we can now determine the semantics of \( \Theta_2 \) as follows. We know that the semantics of \( \Theta_2 \) is \( \mathcal{R}(\phi) \) where \( \phi \) is the semantics of the antecedent VP i.e.,

\[ \lambda x.3y[y/d_2][\text{panda}(y) \land of(y, d_1) \land u(y) \land \text{wash}(y)(x)] \]

where \( u(y) \) denotes a uniqueness condition on \( y \) designed to reflect the definiteness of the object denoted by the expression his panda. And by the definition of \( \mathcal{R} \) we have that:

\[ \mathcal{R}(\lambda x.3y[y/d_2][\text{panda}(y) \land of(y, d_1) \land u(y) \land \text{wash}(y)(x)]) \]
\[ = \lambda x'.3y'[y/d_2][\text{panda}(y) \land of(y, d_1) \land u(y) \land \text{wash}(y)(x)][x' \mathcal{R}(\text{panda}(y) \land \text{of}(y, d_1) \land u(y) \land \text{wash}(y)(x))] \]
\[ = \lambda x'.3y'\mathcal{R}([y/d_2])\mathcal{R}([\text{panda}(y) \land \text{of}(y, d_1) \land u(y) \land \text{wash}(y)(x)]) \]
\[ = \lambda x'.3y'\mathcal{R}(\text{panda}(y)/d_2)\mathcal{R}([\text{panda}(y) \land \text{of}(y, d_1) \land u(y) \land \text{wash}(y)(x)]) \]
\[ = \lambda x'.3y'\mathcal{R}([y'/d_2])\mathcal{R}([\text{panda}(y) \land \text{of}(y, d_1) \land u(y) \land \text{wash}(y)(x)]) \]
\[
= \lambda x'.3y'\{y'/d'_2\}[R(panda(y)) \land R[of(y, d_1)] \land R(u(y)) \land R(wash(y)(x))]
= \lambda x'.3y'\{y'/d'_2\}[R(panda)R((y)) \land R[of)(R(y), R(d_1))]
\land R(u)(R(y)) \land R(wash)R(y)R(x)]
= \{ \lambda x'.3y'\{y'/d'_3\}[panda(y') \land of(y', d_1)] \land u(y') \land wash(y')(x')],
\lambda x'.3y'\{y'/d'_2\}[panda(y') \land of(y', d_3)] \land u(y') \land wash(y')(x')]]
\]

In the last step, two translations are produced because \(R(d_1)\) has two values: \(d_1\) and \(D^{-1}(d_1) = d_3\); the first yields the strict interpretation and the second, the sloppy one. On the strict reading (where Jon and Peter wash the same panda), the uniqueness condition on \(y\) ensures that Jon and Peter do wash the same panda.

### 4.2.3 Comparison with other approaches

Thus the present approach succeeds in accounting for the sloppy/strict distinction. There are however other existing approaches which achieve just the same result but with different means. This begs the following question: What are, if any, the advantages of this particular approach? To answer this question, I will now briefly present these other approaches and compare them with the one I propose.

[Sag 1980]

One of the earliest treatment of the sloppy/strict ambiguity is developed in [Sag, 1980]\(^{20}\). This treatment is characterised by two claims. First, it contends that pronouns becomes sloppy on application of a special rule, called the \(PRO \Rightarrow BVRule\) and second, it posits that sloppy pronouns are bound variables.

Consider the first claim: pronouns become sloppy on application of the \(PRO \Rightarrow BVRule\). To see what the implications of this claim are, we need to examine the content of this \(PRO \Rightarrow BVRule\). The general background of Sag’s analysis is provided by a transformational grammar framework on the one hand and two rules of semantic interpretation on the other. These two rules contrive to assign the sentence in (4.22) two logical forms and consequently two readings: one strict and one sloppy.

\(^{20}\)[Williams, 1977] provides an account of sloppy identity which is very similar.
(4.22) Jeremy fed his possum and Sarah did too.

The first (obligatory) rule of semantic interpretation is part of the theory of verb phrase ellipsis (VPE) and is thus independently motivated. First proposed in [Partee 1975], it was dubbed Derived VP rule (DVPR). The DVPR operates on surface structures to convert VPs into property-type translations written in λ-notation. For instance, on application of DVPR (4.23a) rewrites as (4.23b).

(4.23) a. \([jon]NP[runs]VP\]
    b. \([jon]NP\lambda z[runs(z)]VP\]

The second rule of semantic interpretation caters for sloppy readings. In Sag's approach, this second interpretive rule is called the PRO ⇒ BV rule and is stated as follows.

\[
PRO \Rightarrow BV \text{Rule} \\
NP_i, \lambda z(...PRO_i...) \Rightarrow NP_i, \lambda z(...x...) 
\]

In essence, the PRO ⇒ BV rule predicts that any pronoun which is anaphorically related to the subject NP and occurs within the VP may be replaced by a λ-bound variable. Since there is nothing else in the grammar to account for sloppy pronouns than the PRO ⇒ BV rule, it follows that only pronouns whose antecedent is a subject NP may be sloppy. Note also that since the PRO ⇒ BV rule applies to lambda abstracts and further, lambda abstracts always correspond to VPs, only pronouns occurring in a VP may become sloppy.

There are two problems with this account of sloppy pronouns. The first is that it presupposes that sloppy readings only occur in connection with VPE. As indicated in section 4.2.1, this is simply not true. The second problem is that it predicts that only pronouns with subject NP antecedents may be sloppy. As illustrated by the examples below, this prediction is not supported by the data.

(4.24) a. You can keep Rosa in her room but not Zelda. (Antecedent = OBJ)
    b. I gave a record to Rosa for her 70th birthday and to Zelda too. (Antecedent = IOBJ)
    c. Everyone’s dog is a worry to them but very few people’s cats are. (Antecedent = POSS)
d. A letter from Bill tells you about his plans but a letter from Jon doesn't. (Antecedent = ADJUNCT)

So clearly the specifics of Sag's analysis fail to give a general account of the sloppy identity phenomenon. But while the PRO ⇒ BV rule may be inadequate, the idea of equating sloppy pronouns with bound variables might still be a good one. One could envisage a generalization of Sag's approach which by adopting the general strategy of using lambda abstraction might succeed in giving a correct account of sloppy identity. For instance, the following approach could be adopted:

1. Create a semantic representation for the missing semantics in the e-clause which abstracts over any trigger pronoun present in the a-clause (for instance, in (4.22), the semantics of the VP ellipsis might be represented as $\lambda x. feed(x, x's possum)$, where $x$ is a $\lambda$-bound variable abstracting over the semantics of the trigger pronoun *his*).

2. When the semantics of the antecedent is combined with the semantics of the e-constituents to establish the semantics of the e-clause, ensure that the $\lambda$-variable introduced by the trigger pronoun is bound by the appropriate argument (for instance, in (4.22), when the recovered semantics is combined with the semantics of the e-clause subject — e.g., $\lambda P. P(sarah)$ —, the $\lambda$-variable $x$ introduced by the trigger pronoun *his* is bound in such a way that *sarah* is substituted for $x$ and the resulting sloppy reading is $feed(sarah, sarah's possum)$, which is as required.).

This works if the antecedent of the trigger pronoun in the a-clause is the a-correspondent of some semantic functor over the missing semantics in the e-clause: in this case, both the antecedent and the pronoun will be represented in the missing semantics by the same variable which will then become bound when the semantics of the target antecedent is quantified in. Unfortunately, this is not always true. Consider the following sentence for instance.

(4.25) Every boy said that Mary kissed him. Peter said that she did, Jon said that she did and Paul said that she did.

Here the missing material in the e-clause is the VP *kissed him* and the antecedent of the pronoun *him* is *every boy* which is not a semantic functor over that part of the a-clause semantics which provides the semantics of the missing material in the e-clause. As a result, the correct semantics
for the e-clause cannot be obtained because even if we abstract over the pronoun to transform it into a bound variable (thus getting a semantics for the elided VP such as $\lambda x.kiss(x, y)$), there is no way in which this bound variable can be bound by the appropriate NP in the e-clause (i.e., Peter, Jon and Paul respectively).

This suggests that the bound variable approach is not sufficient to explain sloppy pronouns. My claim here is that what is missing is the recognition that sloppy pronouns are subject to the parallelism constraint. That is, the antecedent of a sloppy pronoun is determined by the parallelism which holds between two clauses. Furthermore, this parallelism need not hold between e- and a-clause, it might extend to the embedding clauses too (as is the case in example (4.25) above).

[Dalrymple et al. 1990]

A more general approach to sloppy identity is developed in [Dalrymple et al. 1990]. This approach uses a restricted form of second order matching to retrieve the possible interpretations of elliptical constructs from the semantics of their a-clause. It is best illustrated by an example. Consider sentence (4.22) above, repeated here:

(4.26) Jeremy fed his possum and Sarah did too.

Let us assume that the representation of the first conjunct is:

$$\text{feed(}\text{jeremy, jeremy's possum})$$

Now, the way in which Dalrymple et al. propose to recover the missing semantics in the e-clause is by setting up the following equation:

$$\text{feed(}\text{jeremy, jeremy's possum}) = P(\text{jeremy})$$

How this equation is established is a question that is left unresolved. The solutions to this equation are provided by the definition of second order unification and will be:

$$P = \lambda x. \text{feed}(x, \text{jeremy's possum})$$
The first solution yields the strict reading and the second the sloppy one. The last two solutions however are no appropriate semantics for the VP ellipsis in the e-clause. To rule them out, an additional restriction is introduced which relies on distinguishing between ‘primary’ and ‘secondary’ occurrences (a primary occurrence is a first occurrence whereas a secondary occurrence is any occurrence that is not primary). The restriction is that all primary occurrences must be abstracted over. In the above example, the first occurrence of jeremy is a primary occurrence and must thus be abstracted over. Hence under this restriction the last two solutions are ruled out.

The semantics of the elliptic clause is then obtained by applying this newly obtained function to the overt constituents of the e-clause (i.e., the e-constituents). So in the above example we will have:

\[
P(sarah) = \lambda x. feed(x, jeremy's possum)(sarah) \\
= feed(sarah, jeremy's possum) \\
P(sarah) = \lambda x. feed(x, x's possum)(sarah) \\
= feed(sarah, sarah's possum)
\]

There is a sense in which this approach is very similar to mine: parallelism between semantic formulae is used to determine which elements can be abstracted over and thus be interpreted sloppily. However there is an important difference which is that I define how this parallelism is to be defined but Dalrymple et al do not. Therefore it seems that their analysis gives us a recipe to produce sloppy readings when needed but fails to give any explanation as to the way in which sloppy interpretation obtains. Note also that although a general operation (second order matching) is used, it must in fact be tailored to adequately account for natural language. Therefore the difference between this tailored version of second order unification and the \( R \) function is not in fact as great as might first be thought. Finally, note that this approach is only suitable for classical logical languages and would have to be redefined were some dynamic logic (such as DRT or DMG) to be used. The main change here would involve redefining the notion of convertibility in such a way that discourse markers could be renamed without affecting the notion of equality used to established the missing semantics. Such an extension would allow for a proper semantics to be construed for the e-clause and make the unification approach one step
closer to the one I propose.

[Asher 1990]

In [Asher 1990], a DRT based approach to sloppy identity is developed whose basic claim is that sloppy pronouns are induced by discourse relations. Again this is best explained by an example. Consider the following examples:

(4.27) a. Mary has been going around saying that Al proposed to her and Sarah has been saying this too. *(strict/sloppy)*

   b. Mary has been going around saying that Al proposed to her but Sarah has been saying this too. *(sloppy)*

(4.27a) allows both for a strict and a sloppy reading whereas in (4.27b) the sloppy reading is strongly favored. This is explained in Asher’s theory as follows.

In the first case, there is a relation of parallelism between the first and the second conjunct. In [Asher 1990], the relation of discourse parallelism induces a mapping from the DRS of the e-clause onto the DRS of the a-clause. Also it requires that anaphora be resolved in a way that maximises semantic parallelism. So when the *this*-anaphora is resolved, it must be done so as to maximise semantic congruence between e- and a-clause. But there are two ways in which this can be done. First, by having Mary and Sarah assert the same property (this yields the sloppy reading) and second, by having them assert the same proposition (this yields the strict reading).

In the second case the discourse relation holding between the two conjuncts is one of contrast so that again, a mapping may be defined from the DRS of the e-clause into that of the a-clause; semantic parallelism must again be maximised but this time anaphora resolution must obtain in such a way that it ensures some semantic contrast between the two clauses. How is this contrast to be obtained in the case of (4.27b)? First note that our world knowledge is such that we expect a man to propose to one woman only. So if the sentential anaphora is resolved to a proposition, the resulting semantics for the whole discourse will be one according to which Al has proposed to Mary and both Mary and Sarah have been talking about it. There is no contrast, no denial of expectation and it becomes hard to understand why a contrastive particle has been used — in short the discourse becomes incoherent. By contrast, if the sloppy reading is chosen then there is a contrast in that contrary to expectation, Al has been proposing to two
women. This explains why the contrastive particle *but* has been used. This also explains why only the sloppy reading is possible.

The main merit of Asher's analysis is that it explains why some readings are more plausible than others. However it does not seem that it can be taken to be a good explanation of sloppy identity in general. The main problem I have with this analysis is that sloppy identity is claimed to result from the requirements set by discourse relations on the semantic content of the clauses they connect. But if that is so, then it must be the case that all discourse relations must have requirements on the semantics of the clauses they connect which are such that they affect the resolution of pronouns. But take the following sentence for instance:

(4.28) Jon washed his car before Peter did.

It is hard to see how a temporal relation such as the one expressed by the connective *before* can in anyway affect the resolution of pronouns.

### 4.2.4 Remaining puzzles

Consider the following example, discussed at length in [Gawron/Peters, 1988].

(4.29) Jon1 [4 revised his1 paper] before the teacher2 did θ5, and Bill3 did θ4 too.

The intended reading is the one where *Jon* and *his* corefer and the source of the elliptical clause is the entire first conjunct. The main problem raised by this kind of data is to get the analysis to yield the correct number of readings. These are generally assumed to amount to four and can be listed as follows.

(4.30) a. before(revise(jon, jon's paper),
    revise(teacher, teacher's paper)) and
    before(revise(bill, bill's paper),
    revise(teacher, teacher's paper))

b. before(revise(jon, jon's paper),
    revise(teacher, teacher's paper)) and
before(revise(bill, jon’s paper),
  revise(teacher, teacher’s paper))
c. before(revise(jon, jon’s paper),
  revise(teacher, jon’s paper)) and
  before(revise(bill, bill’s paper),
  revise(teacher, bill’s paper))
d. before(revise(jon, jon’s paper),
  revise(teacher, jon’s paper)) and
  before(revise(bill, jon’s paper),
  revise(teacher, jon’s paper))

However, under the present analysis, two additional readings are produced:

(4.31) e. before(revise(jon, jon’s paper),
  revise(teacher, jon’s paper)) and
  before(revise(bill, bill’s paper),
  revise(teacher, jon’s paper))
f. before(revise(jon, jon’s paper),
  revise(teacher, jon’s paper)) and
  before(revise(bill, jon’s paper),
  revise(teacher, bill’s paper))

Thus the question arises of whether the present analysis is actually incorrect. It has repeatedly
been suggested ([Asher 1990, Dalrymple et al, 1990]) that the impossibility of these last two
readings is due to the lack of parallelism they induce. This is illustrated by the following table
(here, the leftmost row refer to the example number and the four next ones indicate the referent
of the pronoun his in the four parts of the semantic representation of (4.29) as represented in
(4.30)).

a. J T B T 
 b. J T J T 
 c. J J B B 
 d. J J J J 
 e. J J J B 
 f. J J B J
The first four cases exhibit a symmetry which is absent from the last two. Therefore it might seem like a good idea to restrict $R$ in such a way that this 'parallelism constraint' comes into play. This would actually be rather easy: we could simply require that $R$ uniformly replaces all occurrences of a given discourse marker by the same discourse marker. In the case of (4.29) above, $R$ would then only yield the first four readings. The derivation would go as follows. First, a parallelism is established between the derivation trees of the first two clauses $\textit{Joni} \textit{revised} \textit{his} \textit{paper}$ and $\textit{the teacher} \textit{did} \textit{too}$ and the discourse marker $d_{\text{teacher}}$ is mapped into $d_{\text{Joni}}$. The semantics of the VP in the first conjunct will then be (in abbreviated form):

\[(4.32) \begin{align*}
\text{a. } & \lambda x. \text{before}(\text{revise}(x, d_{\text{Joni}} \textit{'s paper}), \text{revise}(x, d_{\text{Joni}} \textit{'s paper})) \\
\text{b. } & \lambda x. \text{before}(\text{revise}(x, d_{\text{Joni}} \textit{'s paper}), \text{revise}(x, d_{\text{teacher}} \textit{'s paper}))
\end{align*}\]

The semantics of the second VP ellipsis is then obtained as follows. First note that the parallelism between the two conjuncts means that the discourse marker $d_{\text{Bill}}$ is mapped onto $d_{\text{Joni}}$. The semantics of $\textit{Bill did too}$ is then $R(\phi)$ where $\phi$ is either (4.32a) or (4.32b). Under the constraint that all occurrences of the same discourse marker are replaced by the same value, we then have that the two occurrences of $d_{\text{Joni}}$ in (4.32a) must be replaced either by $d_{\text{Joni}}$ itself or by its e-correspondent $d_{\text{Bill}}$. Thus the two readings in (4.31) are excluded and we only get a total of four readings for the sentence in (4.29).

So it is possible to tailor the present analysis to integrate a 'parallelism constraint' on the resolution of multiple pronoun occurrences — but is this really desirable? I think not and this for several reasons. First, consider the following examples taken from [Sag 1980].

\[(4.33) \begin{align*}
\text{a. } & \textit{Edith} \textit{1 said that finding her husband nude had upset her and Sarah did too.} \\
\text{b. } & \textit{Joni} \textit{1 says that he loves his wife and Harry does too.}
\end{align*}\]

In both cases, two pronouns refer to the same individual and there are consequently two occurrences of the same discourse marker ($d_1$). If all discourse marker occurrences are to be replaced by a unique value in the elliptical construct, there are then only two possible interpretations for each of the elliptical constructs: one in which both pronouns (her/ her and he/his) are interpreted as sloppy and the other where both pronouns are interpreted as strict. But in fact, both sentences allow for a third, non-parallel, reading. Thus a possible reading of (4.33a) is one in which Edith said that finding Edith's husband nude has upset Edith and Sarah
said that finding Edith’s husband nude has upset Sarah. Similarly, a possible reading of (4.33b) is one where Jon says that Jon loves Jon’s wife and Harry says that Harry loves Jon’s wife. So it seems that parallel resolution of pronouns is a preferential strategy rather than an absolute constraint. In [Dalrymple et al, 1990] a further example is given which supports this claim.

(4.34) Dewey announce his victory after the newspaper did, but so did Truman.

Americans and politically conscious people favor a reading of this sentence under which Dewey announce Dewey’s victory after the newspaper announce Dewey’s victory, but Truman announce Truman’s victory after the newspaper announce Dewey’s victory i.e., a non-parallel reading.

A further problem is introduced by the examples in (4.33) because it turns out that these sentences also have an impossible reading. These are:

(4.35) a. Edith said that finding Edith’s husband nude has upset Edith and Sarah said that finding Sarah’s husband nude has upset Edith.

b. Jon says that Jon loves Jon’s wife and Harry says that Jon loves Harry’s wife.

As observed in [Sag, 1980] the correct generalization here seems to be that if two pronouns are anaphorically related and one is more deeply embedded than the other, then if the less deeply embedded pronoun is sloppy so must the more deeply embedded one. This suggests that sloppiness is sensitive to semantic links between pronouns. This could be captured by indexing pronouns with information about depth of embedding and modifying \( R \) in such a way that if one pronoun is more embedded than another and is resolved to an e-correspondent then coreferring pronouns that are less embedded are also resolved to this e-correspondent. However such a solution is rather arbitrary and provides no good explanation of the linguistic phenomena involved. Furthermore it leaves unexplained the fact that in (4.29) the following reading is missing:

(4.36) b. before(revise(jonjon’8 paper),
        revise(teacher, jon’s paper)) and
before(revise(bill, jon's paper), revise(teacher, bill's paper))

In general it seems that semantic dependencies between pronouns and between e- and a-terms play some role in determining the number of possible readings in sentences involving either several ellipses or several pronouns. What exactly these dependencies are remains unclear and existing analyses of sloppy identity all fail to account for them. The present analysis fares no better on this particular point and as I have no good explanation for the above facts, I will not try to accommodate them but leave the matter open for further research.

4.3 Summary

Dynamic semantics views the meaning of natural language expressions as residing in their information state update potential rather than in their truth-conditions. With respect to anaphora, this means that an information state will contain some information about potential antecedents and that this information will be used to determine the denotation of anaphors. In this chapter, I have provided a semantics for VPE and gapping which is set within a dynamic framework. I first show that the semantics of these two elliptical constructs cannot be determined at a purely semantic level — syntax plays an important role in determining the antecedent of the ellipsis. I then go on to define the nature of the relationship holding between the semantics of the antecedent and that of the ellipsis. Here dynamic semantics raises a special problem in that if a discourse marker is introduced by some NP occurring within the antecedent, it may not be reused as such in the semantics of the e-clause for if this were so some information would be lost about the preceding discourse, namely that it introduced this specific marker whereas the e-clause introduces another new discourse entity into the information state. I solve this problem by stating that the semantics of the elliptical construct is a function (R) of the semantics of the antecedent where R ensures that discourse markers introduced in the antecedent are renamed in the e-clause.

A further problem which appears in determining the interpretation of elliptical constructs concerns the sloppy identity phenomenon. In this chapter I develop an account of this phenomenon based on the idea that sloppy pronouns only occurs if the syntactic structure of the a-clause is ‘parallel’ to that of the e-clause. The resulting accounts yields a fairly general treatment of sloppy identity and is shown to fare well in comparison with other recent work on sloppy identity.
such as [Dalrymple et al 1990] and [Asher 1990].
Chapter 5

Semantics II

Just as gapping acceptability is constrained mainly by syntax (cf. chapter 3), VP ellipsis has been shown in [Sag 1980] to be constrained mainly by semantics. Set within a transformational framework and adopting a deletion based approach to VPE, Sag [1980] presents what remains to date the most exhaustive analysis of VP ellipsis and of its interaction with other linguistic phenomena such as extraction, quantification and pronominalisation. Central to this analysis is a semantic constraint, dubbed the alphabetic variance constraint. As we shall shortly see, this constraint is crucial in adequately characterising VPE acceptability. In Sag’s account the alphabetic variance constraint is stated at the level of logical form (LF) and requires checking the LF of both a- and e-clause. Furthermore, the logical form is a formula of a semantic representation language akin to predicate logic augmented with lambda terms, i.e. a classical logic (as opposed to a more dynamic framework where meaning rather than being stated in terms of truth conditions, is stated in terms of information update potential). This raises the following questions: Is this syntactic checking of LFs necessary? Can Sag’s analysis be recast within a dynamic framework such as DIL? Will a dynamic framework help give a more natural account of the alphabetic variance constraint?

Interestingly the main import of the alphabetic variance constraint is to rule out free variables (cf. section 5.1). Modern dynamic theories of semantics for natural language are deeply concerned with the nature and the interpretation of variables. Indeed it is by taking as a starting point the idea that indefinite NPs as well as pronouns translate into free variables that Discourse Representation Theory (DRT) succeeded in solving the problem raised by donkey sentences. As
we shall see, following work of Klein, it is possible to give a DRT based account of VP ellipsis in which alphabetic variance (or α-variance) simply falls out of the semantics (and thus does not need to be stated as an additional constraint on LF). Furthermore, it can also be shown that Klein's DRT-based account can be recast within DIL. Some changes have to be introduced. Firstly, DIL must be made partial (this is crucial in allowing for the alphabetic variance constraint to fall out of the semantics). Secondly, the DIL notion of context must be modified (this is to allow for a DRT like treatment of proper nouns). In DIL, contextual information is encoded in states which function as assignment of values to discourse markers. There is only one notion of context, that of 'local' (or current) context. We add to it the notion of a 'global' context and thus obtain a notion of context which is reminiscent of the distinction made in the semantics for programming languages between (global) store and (local) environment.

This chapter is structured as follows. Section 5.1 outlines Sag's account of VPE and defines the notion of alphabetic variance. Section 5.2 sketches how DRT allows for a semantic definition of alphabetic variance. Section 5.3, the heart of the section, considers in detail how DIL can be extended to yield a model theoretic account of alphabetic variance thus making Sag's syntactic checking on LFs redundant. To further illustrate the workings of the proposed treatment, section 5.4 goes through the semantic interpretation of a discourse violating the alphabetic variance constraint.

5.1 Sag's account of VPE

As already mentioned, Sag's account is set within a transformational framework and assumes that VP ellipsis involves deletion. Deletion theories postulate that two identical constituents are generated in the base, and that one is subsequently deleted providing that some identity condition is met. The main characteristic of Sag's approach is that this identity condition refers neither to the string nor to the syntax of the deleted VP but to its semantic representation, which in transformational grammar is referred to as logical form. More specifically, Sag argues that VPE is characterised by the fact that any VP whose logical form (LF) is an alphabetic variant of that of the antecedent can be deleted. Alphabetic variance is defined on a formula representing the discourse under consideration and holds between two subformulae of this formula which represents the antecedent and the elided VP respectively.

\[1\]Note that Sag's definition of alphabetic variance differs from the definition of α-equivalence familiar from the λ-calculus in that it takes into account the discourse context in cases where the formulae being compared
Definition 5.1 Alphabetic variance (or $\alpha$-variance)

Suppose that $\phi$ and $\psi$ are the LFs of two VPs and that $\theta$ represent the discourse these VPs are part of. Then for $\phi$ and $\psi$ to be alphabetic variants of each other we must have firstly that $\phi$ and $\psi$ are identical up to the renaming of bound variables, and secondly that any variable $x$ which is free in both $\phi$ and $\psi$, is bound by the same operator in $\theta$.

Given this definition of $\alpha$-variance, Sag defines the $\alpha$-variance constraint as follows:

Definition 5.2 Alphabetic variance constraint (or $\alpha$-variance constraint)

With respect to a sentence $S$, VP Deletion can delete any VP in $S$ whose representation at the level of LF is a $\lambda$-expression that is an $\alpha$-variant of another $\lambda$-expression present in the LF of $S$ or in the LF of some other sentence $S'$ which precedes $S$ in discourse. [Sag, 1980:105]

For example in (5.1) below, the LF of the antecedent $\lambda z.love(z,j)$ is an alphabetic variant of that of the elided VP: the two $\lambda$-terms are identical up to renaming of bound variables ($z$ is renamed $y$) and there is no free variable.\(^2\)

(5.1) a. John believes that Mary loves him. But she doesn't.
    b. $\text{believe}(j, \lambda x.(x,j)(m)) \text{ but } \lambda y.\text{love}(y,j)(m)$

By way of contrast, consider the discourse in (5.2):

(5.2) a. No man believes that Mary loves him. *But she does.
    b. $\neg \exists z[\text{man}(z) \land \text{believe}(z, \lambda z.\text{love}(x,z)(m))] \text{ but } \lambda y.\text{love}(y,z)(m)$

This time the LF, $\lambda z.\text{love}(x,z)$, of the antecedent VP contains the free variable $z$. We must therefore verify that $z$ is bound by the same operator in the antecedent and in the anaphoric contain some free variable. In contrast, in the lambda-calculus, two terms are alphabetic variant of each other if they are identical up to renaming of variables.

\(^2\)Throughout this section, I will adopt Sag's notation for representing the meaning of sentences: the semantics of a VP is represented by a $\lambda$-abstract and NPs are represented either by a constant (proper nouns) or by a quantifier (other cases). However to keep things simple I will not adopt Sag's linearisation which places the semantics of the subject before that of the VP. Rather I will represent a sentence meaning by the functional application of the VP semantics to the subject semantics. So for instance the semantics of Jon runs will be represented by the logical form $\lambda x.\text{run}(x)(\text{jon})$ rather than $\text{jon}, \lambda x.\text{run}(x)$.
clause. In fact examination of (5.2b) reveals that \( z \) is bound by the existential in the antecedent clause but remains free in the anaphoric clause. Hence the LF of the antecedent is not an alphabetic variant of that of the deleted VP. Hence VPE is forbidden.

But what is this alphabetic variance constraint useful for? The above example illustrates how it explains some facts about the interaction between pronominalisation and VPE: if in a discourse containing a VPE, a pronoun bound in the a-clause becomes unbound in the e-clause, VPE is unacceptable and the discourse is ill-formed. There are however, many other facts about VPE and its interaction with other linguistic phenomena which become understandable once Sag’s constraint of alphabetic variance is adopted. These include: quantification, extraction, pseudo-clefts, ready constructions and equi-sentences\(^3\). The following examples illustrate this.

First, consider the interaction of VPE with quantification.

(5.3)  
\( a. \) Jon greeted everyone when Peter did.  
\( b. \) Jon greeted everyone and Peter did too.

In (5.3a), the quantifier everyone can have VP scope or sentence scope. In the first case, the sentence is assigned a collective reading: Jon greeted a group of people when Peter greeted the same group of people whereas in the second, it is assigned a distributive reading: for every person \( x \) present, Jon greeted \( x \) when Peter greeted \( x \). Now consider (5.3b).

Only one reading is possible, the collective one. Under Sag’s theory this is explained by the fact that in (5.3a), alphabetic variance is not violated under the collective and the distributive residing. Thus the two readings are possible. By way of contrast, the distributive reading of (5.3b) induces a violation of the \( \alpha \)-constraint, as is illustrated by the following formula.

\[ \forall x [\lambda y.\text{greet}(y, x)(\text{jon})] \land \lambda y.\text{greet}(y, x)(\text{peter}) \]

\(^3\)Note however that Sag’s approach crucially relies on the assumption that the semantics of such linguistics objects as extracted NPs, cleft-sentences and equi-constructions all involve a variable which must be bound by some logical operator. This is clearly controversial. For instance, consider example (5.4): one might want extracted NPs to be represented by some special variable which will be bound by any \( \text{wh} \)-word; in which case Sag’s account fails. Nonetheless Sag’s approach has the merit of yielding a uniform account of a wide range of data concerning the interaction of VPE with other linguistic phenomena. Moreover, an investigation of the correct semantics for these various linguistic phenomena is outside the scope of the thesis. In what follows, I will therefore assume that Sag’s assumptions are defendable and consider the \( \alpha \)-variance constraint to be an adequate restriction on VPE.
CHAPTER 5. SEMANTICS II

Assuming that a universal cannot quantify over a conjoined sentence, we have the result that the free variable $x$ occurring in the semantic representations of the two VPs is bound by the universal in the antecedent clause but not in the e-clause. Thus $\alpha$-variance is violated and VPE is ruled out.

A similar reasoning holds for cases of extraction. Consider the following discourses.

(5.4)  

<table>
<thead>
<tr>
<th>a. We finally got in touch with Al, who my brother Jon tried to visit, but couldn't.</th>
</tr>
</thead>
<tbody>
<tr>
<td>b. *We finally got in touch with Al, who my brother Jon tried to visit, but who he couldn't.</td>
</tr>
</tbody>
</table>

Under Sag's analysis, wh-words are treated as variable binding operators so that the semantic representation of the coordinated construct in (5.4b) is:

$$(\text{who x}) [\lambda y.\text{try}(y, \lambda z.\text{visit}(z, x))(y))(\text{jon})] \text{ but } (\text{who v}) [\neg \text{could}(\lambda z.\text{visit}(z, x))(\text{jon})]$$

Again a free variable ($x$) in the antecedent VP is bound in the a-clause and free in the e-clause. Thus alphabetic variance is violated and VPE is unacceptable.

The interaction of VPE with cleft-sentences can also be shown to be subject to the alphabetic variance constraint. Consider the following examples:

(5.5)  

<table>
<thead>
<tr>
<th>a. What Sandy wanted to buy, but couldn't, was the baseball-bat.</th>
</tr>
</thead>
<tbody>
<tr>
<td>b. *What Sandy carried was the catcher's mitt, and what Betsy did was the baseball-bat.</td>
</tr>
</tbody>
</table>

Under Sag's analysis, pseudo-cleft sentences are represented in terms of set abstraction and the semantic representation for (5.5b) is:

$$(\text{the catcher's mitt} = \exists y.\text{carry}(y, x)(\text{sandy})) \wedge$$
$$(\text{the baseball bat} = \exists y.\text{carry}(y, x)(\text{betsy}))$$
Under the assumption that $\hat{x}$ is a binding operator for the variable $x$, we have that again the formula for the starred example contains a free variable ($x$) and again, the corresponding discourse is ill-formed. Finally consider the interaction of VPE with ready constructions:

\[(5.6)\]

a. Betsy is ready to give up, and Peter is ready to also.

b. *The steak is ready to eat and the chicken is ready to also.

with formula for (5.6b)\footnote{Sag's assumption here is that ready constructions involve an empty subject. $\Delta$ is the semantic representation associated with this empty subject.}:

$$
\lambda z.\text{ready}(x, \lambda y.\text{eat}(y, z)(\Delta))(\text{the steak}) \land \\
\lambda z.\text{ready}(z, \lambda y.\text{eat}(y, z)(\Delta))(\text{the chicken})
$$

For all of the above cases, discourses where VPE is ruled out are discourses in which alphabetic variance is violated because some free variable in the antecedent VP is bound in the antecedent but free in the elided clause. Thus $\alpha$-variance appears to adequately characterise the relation that should hold between an elided VP and its antecedent in that whenever alphabetic variance is violated, VPE is ruled out. This leads us to the question of how $\alpha$-variance is to be determined. In Sag's account, the implicit assumption is that $\alpha$-variance checking requires examining the LF of the whole discourse under consideration to determine whether or not a free variable occurring within the antecedent VP is bound by the same operator in both the antecedent and the deletion site. The next section will show that a DRT-based account of VPE allows for a more natural treatment of $\alpha$-variance i.e., a purely model-theoretic one.

## 5.2 VPE and Discourse Representation Theory

As illustrated in the preceding section, the main problem when dealing with $\alpha$-variance concerns the semantics of free variables. An interesting result that relates to this issue is given in [Chierchia and Rooth 1984]. In essence, the paper shows that DRT forbids the equivalent of free variables, that is, “free discourse markers” (DM).

In DRT, a discourse representation structure (DRS) is a structure consisting of a universe i.e., a set of discourse markers and some \textit{conditions} i.e., a set of formulae restricting the possible
denotation of these discourse markers. Let $U_K$ be the universe of any DRS $K$ and $CON_K$ be the set of conditions listed in $K$. Chierchia and Rooth's proof that any DRS containing a 'free discourse marker' is uninterpretable can be summarised as follows. First note that according to [Kamp 1981], DRSs are interpreted with respect to proper embeddings where proper embeddings are partial functions from discourse markers to individuals. Typically in DRT, interpretation will start with respect to an initial embedding, say $f$, which is undefined for all discourse markers. A DRS $K$ will then be said to be interpretable wrt a proper embedding $f$ iff $\exists g$ such that $f \subseteq U_K$ and $g \models CON_K$ or $g \not\models CON_K$. More relevantly, a DRS $K$ will be uninterpretable wrt a proper embedding $f$ if $\exists d \in CON_K$ such that $g(d)$ is undefined (where $f \subseteq U_K g$).

What Chierchia and Rooth show is that saying that a DM $d$ has no accessible antecedent in $K$ (or equivalently, $d$ is free in $K$) interpreted wrt some proper embedding $f$ is equivalent to saying that $d \in CON_K$ but $d \not\in dom(f) \cup U_K$ ($d$ appears in the conditions listed in $K$ but is not in the set of DMs that are accessible from $K$). Therefore for any function $g$ such that $f \subseteq U_K g$, $g$ will be undefined on $d$. Hence $K$ is uninterpretable wrt $f$. Note that this results crucially depends on embeddings being partial.

By way of example, suppose that we try to interpret the DRS $K_1$ below which represents the sentence $A$ man$_1$ loves him$_2$ (and there is no deictic referent available for him).

$$
\begin{array}{c}
\text{d}_1 \\
\text{man(}d_1\text{)} \\
\text{love(}d_1, d_2\text{)}
\end{array}
$$

To interpret $K_1$ with respect to a model $M$, we must find a proper embedding $f$ of $K_1$ into $M$. $f$ is a proper embedding of $K_1$ into $M$ iff $\exists g$ such that $f \subseteq U_K$ and $g \models CON_K$. Furthermore, $f \subseteq X g$ iff $dom(g) = dom(f) \cup X$ and $f \subseteq g$. $U_K = \{d_1\}$. Since the initial embedding is undefined on all discourse markers, $dom(f) = \emptyset$. Therefore $dom(g) = \{d_1\}$. So we must find an embedding $g$ with domain $\{d_1\}$ that verifies the set of conditions $\Phi$ listed in $K_1$. But $d_2$ occurs in $CON_K$, and $g$ is not defined on $d_2$. Hence $g$ cannot verify $\Phi$ and interpretation fails.

Thus in DRT, if a pronoun appears outwith the scope of its antecedent, interpretation fails. Clearly this gets us one step further in trying to account for $\alpha$-variance semantically, for what
we need to deal with sentences like (5.2) is some mechanism ensuring that a pronoun bound in the antecedent is also bound in the semantic representation associated with the elliptical clause. Traditional DRT allows us to "detect" free pronouns. Does the result transfer to DRT accounts of VPE? In [Klein 1987], a DRT treatment of VPE is given which introduces predicatives DRSs and predicate markers (PMs). It is easy to prove that Klein’s treatment of VPE allows for a semantic definition of α-variance: we know from the Chierchia and Rooth’s paper that if a discourse marker occurs free in a DRS, then interpretation fails. The only difference between Klein’s version of DRT and standard DRT resides in the introduction of predicate markers and predicative DRSs. Therefore all we need to show is that if a discourse marker occurs free in a predicative DRS, interpretation fails (just as when a discourse marker occurs free in a ‘normal’ DRS). This is obvious from the fact that like normal DRSs, predicative DRSs are interpreted with respect to a given embedding and that their interpretation requires the interpretation of all the discourse markers they may contain. Since violation of α-variance automatically results in failure of interpretation, the need for a syntactic check on the global LF of the discourse disappears. The following example illustrates this. Consider the DRS $K_2$ associated with the discourse given in (5.2) and repeated here for convenience.

(5.7) No man$_1$ believes that Mary loves him$_1$. *But she does.
Here a few remarks are in order about the treatment of negation and of proper nouns. In DrT, negation is seen as 'closing off' a DRS thus making the discourse markers in its universe inaccessible to subsequent anaphors. In particular, note that the discourse marker $d_1$ associated with $him_1$ will not be accessible when interpreting the condition $p_1(d_2)$ representing the e-clause. Furthermore, note that proper nouns are treated differently from indefinite NPs: whereas indefinites update the universe of the current, most 'local' DRS with a new discourse marker, proper nouns update the universe of the top-level, or 'global' DRS (for instance, note that the proper noun Mary updates the universe of $K_2$ rather than the universe of the DRS in which the corresponding condition occurs). As we shall see, these two assumptions cause problems when trying to recast Klein's analysis into Dil. Section 5.3 shows how Dil can be modified to resolve these problems.

Now consider how the condition $p_1(d_2)$ introduced by the e-clause is interpreted. First note that the notation $[z]$ indicates a distinguished discourse marker $z$ "which plays the role of the individual to which the predicate is applied" [Kamp 1983:52]. Furthermore, the notation $p_1 : DRS$ indicates that the predicate marker $p_1$ denotes the object denoted by $DRS$. So using the lambda-calculus notation, $p_1$ is equivalent in its intended interpretation to:

$$\lambda d.\text{love}(d, d_1)$$

And similarly, according to the notion of application used by Klein, the application $p_1(d_2)$ of the predicate marker $p_1$ to the individual discourse marker $d_2$ is equivalent to:

$$\text{love}(d_2, d_1)$$

In terms of DRSs, this means that interpreting $p_1(d_2)$ in $K2$ is equivalent to interpreting:

$$\text{love}(d_2, d_1)$$

in $K_2$. But in $K_2$, $d_1$ has no accessible antecedent. Hence $K_2$ is uninterpretable.

### 5.3 VPE and Dynamic Intensional Logic

This section shows how Klein's DrT-based account of VPE can be recast within Dil.
DIL revisited

To give a DIL account of VPE along the lines of Klein (1987), we need to modify DIL in two ways. As already mentioned, partiality is a crucial element in gaining a semantic account of alphabetic variance. But DIL is in no sense partial. Hence we need to introduce into DIL the same type of partiality as is present in DRT. This will be the first modification I propose. Another problem with DIL has to do with the notion of context it provides. As already mentioned in the introduction, the DIL notion of context is in fact too limited to correctly account for pronominal anaphora and in particular for anaphoric reference to proper nouns. In order to obtain a notion of context that is adequate for our purpose, I will thus propose a second modification to DIL: the meaning of an expression rather than being a function of a state will be a relation between contexts where a context will consist (as in some formal semantics for programming languages) of a local context (corresponding to DIL notion of state) on the one hand and a global context (needed to give a proper semantics for proper nouns) on the other.

Adding partiality

The first modification we need to make concerns partiality. It is very simple. To obtain a semantic account of alphabetic variance, we need to have in DIL a notion of partiality similar to that present in DRT. In DRT, partiality is a property of proper embeddings: proper embeddings are partial functions from discourse markers into individuals. This contrasts with the DIL assignments of values to discourse markers which are total functions from states into individuals. By making them partial, we get a notion of partiality which is similar to that present in DRT. This is what we do. When interpreting a discourse, we start with the initial state, a state at which all discourse markers are undefined. As discourse proceeds, NPs introduce state-switchers which allow for interpretation to be made relative to some new state(s) each of which may be richer in information than the precedent. At any stage of discourse interpretation, if a discourse marker has to be interpreted relative to a state which is undefined for this particular discourse marker, interpretation simply fails. In this way any discourse which violates the α-variance constraint will fail to be interpreted because its semantic representation will contain a free discourse marker and any DIL formula containing a free discourse marker induces an interpretation failure.

Modifying the notion of context

Note that the term 'partiality' here only refers to the partiality of the semantics of discourse markers. That is, relations, properties and propositions remain total.
As already mentioned, DRT discourse representations consist of a universe (a set of discourse markers) and a list of conditions. A condition may itself contain a DRS. A condition may also be a complex DRS i.e, one consisting of several DRSs. So in general, a DRS may be a complex structure containing other DRSs. Looking at this structure one can talk of the different levels of the DRS. Suppose for instance that a DRS $K_1$ is embedded in some other DRS $K$. We can refer to $K$ as being the top level of the DRS and think of the universe of $K$ as giving some global notion of context. In contrast, the universe of the current DRS $K_1$ can be thought of as forming the local context. The intuition behind this notion of global and local context is that discourse markers that occur in the global universe are always accessible whereas discourse markers which occur in the local universe are not. In DRT this distinction between local and global context is exploited both for proper nouns and for VPE. The DRS $K_2$ drawn in section 5.2 and repeated here for the sake of convenience illustrates this point.

In this DRS, both the proper noun Mary and the VP loves him introduce their markers in the global universe rather than in the local one. Note that without this notion of global context, it is unclear how anaphoric reference to names can be handled. Thus suppose that DRT does not make use of the top level of the DRS. Then we have the result in this case that the discourse marker ($d_2$) associated with the proper noun Mary must be introduced in the local context that is, in the sub-DRS appearing within the scope of the negation operator. But then $d_2$ will not be
accessible from the main DRS where the condition $p_1(d_2)$ representing the anaphoric clause is listed. Hence interpretation will fail. In this case, of course we want the discourse to be rejected as ill-formed (because the discourse violates the $\alpha$-variance constraint) but not because of the discourse marker for Mary. For if this were so, then the perfectly well-formed discourse Every woman likes Pedro. He2 is a sweetie would also be uninterpretable since the DM introduced by Pedro2 would be local to the sub-DRS triggered by every and would thus be inaccessible from the main DRS where the condition corresponding to the pronoun $he$ is listed.

The problem we encounter is that the notion of context DIL offers is strictly local. Why is this? An intuitive explanation can be given along the following lines. First, note that state-switchers are responsible for updating a state i.e., for creating a new state with respect to which a given (part of the) formula will be interpreted. Note further that the effect of a state-switcher only holds with respect to the part of the formula over which it has scope. Therefore whenever the scope of a state-switcher is closed the binding introduced by this state-switcher is undone and the information introduced by this binding is lost. In this sense the DIL notion of context is local: only bindings that take scope over the formula can be taken into consideration. The following example illustrates.

\[(5.8)\]

\begin{enumerate}
  \item Every man likes Pedro. He2 is a sweetie.
  \item \[\forall y[y/d_1][\text{man}(y) \rightarrow \{p/d_2\}\text{like}(p)(y)] \land \text{sweetie}(d_2)]^{M,s,s}\]
  \item \[\forall y[y/d_1][\text{man}(y) \rightarrow \{p/d_2\}\text{like}(p)(y)]\text{M, s,s}\]
  \item \[\text{sweetie}(d_2)]^{M,s,s}\]
\end{enumerate}

The semantics for (5.8a) is as given in (5.8b) which in turn will depend on the denotations given in (5.8c). Since each conjunct is interpreted with respect to state $s$, any state update occurring within the first conjunct will have no effect on the interpretation of the second conjunct. In particular, the binding \(\{p/d_2\}\) introduced by Pedro remains local to the first conjunct. Hence, even though the discourse in (5.8a) is perfectly well-formed, interpretation of (5.8b) will fail to assign the correct interpretation to the discourse marker $d_2$ representing the pronoun $he$ in the second sentence of the discourse. That is, the notion of anaphoric binding provided by DIL fails to give a proper account of the anaphoric potential of proper nouns.

At this point, formal semantics for block structured programming languages suggests a way out. In this branch of semantics, contextual information is partitioned into two components: the environment which records identifier bindings and the store which records the effect of
assignments. While identifier bindings are local to a block and are undone on leaving the block, assignments have an irreversible effect on the content of the store. In connection with discourse semantics, this idea can be exploited as follows. Paralleling the partition of the computational state, contextual discourse information can be thought of as being recorded in a store consisting of both a local state and a global context. Furthermore, while indefinite NPs introduce state-switchers, names will now be assigned such a DIL translation that they introduce in the formula a context-switcher. That is, they update the global part of the state namely the context. Finally, to model the fact that context updates are passed on throughout discourse, I posit that meaning is not only a function from the model M and a variable assignment g but also a relation between stores f and f' (rather than a function from states). Accordingly, the meaning of any expression a is now denoted by ifgM'[j] rather than [a]Mg where f and f' are of the form (l, gl) with l ranging over (local) states and gl over (global) contexts. In what follows, this is often abbreviated to j[l].

I have now described the two changes that are necessary for a DIL treatment of VPE. These can be summarised as follows: contextual information is represented by a set of stores consisting of a (local) state and a (global) context. States and contexts act as partial rather than total assignments of values to discourse markers. Meaning is a relation between stores (and a function of a model M and an assignment of values to variables g).

The revised version DIL2 of DIL is as follows.

Syntax

Definition 1 (Types)
The set T of types is defined as follows. e is a type. t is a type. If a and b are any types, then a —► b is a type. If a is a type, then s —► a is a type.

Definition 2 (Syntax)

1. If \( \alpha \in CON_a \cup VAR_a \), then \( \alpha \in ME_a \)
2. If \( \alpha \in DM \), then \( \alpha \in ME_e \)
3. If \( \alpha \in ME_{a->b} \) and \( \beta \in ME_a \), then \( \alpha(\beta) \in ME_b \)
4. If \( \phi, \psi \in ME_1 \), then \( \neg \phi, [\phi \land \psi], [\phi \lor \psi], [\phi \rightarrow \psi] \in ME_1 \)
5. If $\phi \in ME_t$ and $v \in VAR_a$, then $\exists \nu \phi, \forall v \phi \in ME_t$

6. If $\alpha \in ME_a, v \in VAR_a$ then $\lambda v \alpha \in ME_{a \rightarrow a}$

7. If $d \in DM, \alpha \in ME_t, \beta \in ME_a$, then $(\alpha/d)\beta \in ME_a$

8. If $d \in DM, \alpha \in ME_t, \beta \in ME_a$, then $(\alpha/d)\beta \in ME_a$

9. If $\alpha \in ME_a$, then $\neg \alpha \in ME_{t \rightarrow a}$

10. If $\alpha \in ME_{a \rightarrow a}$, then $\neg \alpha \in ME_a$

11. Nothing else is in $ME_a$ for any type $a \in \mathcal{T}$

The difference between the syntax of DIL and that of DIL$^2$ emerges in clause 8 which allows for context switchers: whereas state-switchers are of the form $(\alpha/d)$, context-switchers are written $(\alpha/d)$.

Semantics

A model for DIL$^2$ is a 4-tuple $(D, S, F)$ where $D$ is a non-empty set of individuals, $F$ is the interpretation function and $S$ is a set of stores formed out of the cartesian product $LOC \times GLOB$ of the sets of local states and global contexts respectively. Postulates 1 and 2 constrain the behaviour of local states as indicated in section 2.3.2, while global states are subject to similar postulates: postulates 3 and 4. The set of possible denotations of type $a$ is defined as follows.

Definition 3 (Domains of interpretation)

$D_t = D, D_{a \rightarrow a} = D_a^D, D_{a \rightarrow t} = D_a^S$ and $D_t = \{0, 1\}$.

As in DIL, the interpretation function $F$ does not always assign to expressions of $ME_a$ a denotation in $D_a$. In particular, the denotation of a discourse marker is an intensional entity. More specifically, if $\alpha \in DM$ then $F(\alpha) \in D^S$. For all other expressions, if $\alpha \in ME_a$ then $F(\alpha) \in D_a$.

The semantics of DIL$^2$ recursively defines, for any expression $\alpha$ of DIL$^2$ the interpretation of $\alpha$ denoted $f[\alpha]_f(M$ and $g$ are ignored for simplicity) as follows.

Definition 4 (Semantics)

Footnote: Postulates 3 and 4 have the same content as postulates 1 and 2 respectively except that all references to states are replaced by references to global contexts.
1. \( f(c) = F(c) \), for every constant \( c \)
   \( f(v) = g(v) \), for every variable \( v \)

2. \( (l,s)[d]_{(l,s')} = F(d)g(l) \) if \( F(d)(l) \) is undefined
   = \( F(d)(l) \) otherwise, for every discourse marker \( d \)

3. \( (l,s)[\alpha(\beta)]_{(l,s')} = (l,s)[\alpha]_{(l,s')} (l,s)[\beta]_{(l,s')} \)

4. \( (l,s)[\neg \psi]_{(l,s')} = 1 \) iff \( (l,s)[\psi]_{(l,s')} = 0 \)
   = 0 otherwise

   \( (l,s)[\psi \wedge \psi']_{(l,s')} = 1 \) iff \( (l,s)[\psi]_{(l,s')} = (l,s)[\psi']_{(l,s')} = 1 \)
   = 0 otherwise

   \( (l,s)[\psi \lor \psi']_{(l,s')} = 1 \) iff \( (l,s)[\psi]_{(l,s')} = 1 \) or \( (l,s)[\psi']_{(l,s')} = 1 \)
   = 0 otherwise

   \( (l,s)[\neg \psi]_{(l,s')} = 0 \) iff \( (l,s)[\psi]_{(l,s')} = 1 \) and \( (l,s)[\psi']_{(l,s')} = 0 \)
   = 1 otherwise

5. \( (l,s)[\exists \psi]_{(l,s')} = 1 \) iff there is a \( d \in D_a \) such that \( (l,s)[\psi]_{(l,s')} = 1 \), where \( a \) is the type of \( v \)
   = 0 otherwise

   \( (l,s)[\forall \psi]_{(l,s')} = 1 \) iff for all \( d \in D_a \) it holds that \( (l,s)[\psi]_{(l,s')} = 1 \), where \( a \) is the type of \( v \)
   = 0 otherwise

6. \( (l,s)[\lambda \alpha \psi]_{(l,s')} = \text{the function } h \in D_b^2 \) such that \( h(d) = (l,s)[\psi]_{(l,s')} \)
   for all \( d \in D_b \) where \( a \) is the type of \( \alpha \), and \( b \) the type of \( v \)

7. \( (l,s)[\alpha/d]_{(l,s')} = (l,s)[\beta]_{(l,s')} \) where \( \beta = (d \leftarrow (l,s)[\alpha])l' \)

8. \( (l,s)[\alpha/d\beta]_{(l,s')} = (l,s)[\beta]_{(l,s')} \) where \( \beta_1 = (d \leftarrow (l,s)[\alpha])g_1 \)
9. \( (l,v)[\alpha][l,v'] = \) that function \( h \in D_\alpha \)
 such that \( h(i) = (l, s_{i_1}, \alpha_{i_1}, l_1) \) for all \( i = (l, g, l) \in \mathcal{S} \)

10. \( (l,v)[\alpha][l,v'] = (l,v)[\alpha][l,v'](l,v) \)

5.4 A Sample Semantic Analysis

This section shows how example (5.2) is ruled out in DIL2 as violating the \( \alpha \)-variance constraint between antecedent and deleted VP. This is carried out by demonstrating that the DIL2 translation of (5.2) fails to be assigned an interpretation. ⁸

The DIL2 translation of the antecedent clause *No man believes that Mary loves him* is:

\[
\neg \exists x \{x/d_1\} [\text{man}(x) \land \text{believe} ((m/d_2) \text{love}(d_1)(m))(x)]
\]

and the semantic representation of the e-clause *But she does* consists of the semantics of the subject NP *she* being applied to the (reconstructed) semantic representation of the ellipsis *does*:

\[
\begin{align*}
\lambda P.P(d_2) & (R(\lambda y.\text{love}(d_1)(y))) \\
= \lambda P.P(d_2) & (\lambda y'.\text{love}(d_1)(y')) \\
= \lambda y'.\text{love}(d_1)(y')(d_2) \\
& = \text{love}(d_1)(d_2)
\end{align*}
\]

So the semantics of the whole discourse is:

\[
(l,v)[\neg \exists x \{x/d_1\} [\text{man}(x) \land \text{believe} ((m/d_2) \text{love}(d_1)(d_2))(x)] \land \text{love}(d_1)(d_2)](l,v)
\]

which depends on the semantics of:

\[
(l,v)[\neg \exists x \{x/d_1\} [\text{man}(x) \land \text{believe} ((m/d_2) \text{love}(d_1)(m))(x)]](l,v)
\]

⁸For the sake of simplicity, the static translation of (1) is given here rather than the dynamic one. The static translation of a discourse with dynamic translation \( \lambda p[\phi \land p] \) is \( \lambda p[\phi \land p](\text{true}) \equiv \phi \).
and of:

$$(l,g_l_1)[\text{love}(d_1)(d_2)](l,g')$$

where $g_l_1 = (d_2 \leftarrow (l,g_l_1)[m](l,g_l_1))g_l$

Assume that $l$ denote the initial state and $g_l$ the initial context that is, $F(d)(l)$ and $F(d)(g_l)$ are undefined for all $d \in DM$. Given the semantics of DIL$_2$, the following reasoning holds.

$$(l,g_l_1)[\text{love}(d_1)(d_2)](l,g')$$

$$= (l,g_l_1)[\text{love}](l,g_l_2)[(l,g_l_3)[d_1](l,g_l_3)](l,g_l_3)[d_2](l,g')$$

where $g_l_1 = g_l_2 = g_l_3 = g'$

But as noticed above, the initial state $l$ is undefined for all $d \in DM$. Further $g_l_1$ is only defined on $d_2$. Hence the interpretation $(l,g_l_3)[d_1](l,g_l_3)$ of $\text{him}$ will fail because both $F(d_1)(l)$ and $F(d_1)(g_l_1)$ are undefined.

## 5.5 Summary

The goal of this section was to show how Sag's account of VPE could be recast within DIL. I first showed that within a dynamic framework such as DRT, the $\alpha$-variance constraint simply falls out of the model theory. To obtain similar results using DIL, I then revised DIL by first, making the assignments of values to discourse markers partial and second, providing a richer notion of context on the other. Finally, I went through an example and showed how, given this revised version of DIL, the interpretation of a discourse violating the $\alpha$-variance constraint fails.
Chapter 6

Discourse

The main part of this thesis aims at providing an analysis of gapping and VPE which accounts for both their syntax and their semantics. However it is easy to show that a full account of ellipsis cannot be given unless pragmatics is considered as well\(^1\). There are two main reasons for this. First, as we have seen in chapter 3, the interpretation of gapped clauses is heavily dependent on pragmatic factors such as processing strategy and information structure. As this issue has already been discussed in section 3.1.3, I will not consider it here. A second reason has to do with VP ellipsis: as we shall see, there are good reasons to think that VPE resolution is sensitive to discourse structure. This chapter investigates the question of whether discourse theory can be shown to explain the restrictions regulating VPE resolution. Section 6.1 presents some data about VPE which suggests that pragmatics constrains VPE resolution, whereas sections 6.2 and 6.3 indicate how a theory of discourse could help modelling those constraints.

6.1 Data

Most of the linguistic work on VP ellipsis has concentrated on giving a precise account of the relationship that must hold between the anaphor and its antecedent, and how this relationship interacts with other linguistic phenomena such as extraction, quantification and pronominalisation. In contrast, very little work has been done on defining a resolution procedure for VP

\(^1\)I include under the term 'pragmatics' fields as diverse as discourse theory, information and thematic structure theory, and processing strategy theory.
ellipsis. On the basis of examples such as (6.1), it might be tempting to expect that resolving to the nearest preceding VP is a reasonable resolution strategy.

(6.1) Mary [1 started talking]. Sarah [2 interrupted]. Mary did θ₂ too.

In this example, the elided VP is understood as referring to the second VP i.e., the VP immediately preceding the e-clause, rather than to the first one. However, as [Klein/Stainton-Ellis 1989] has shown, the antecedent of a VP ellipsis need not always occur in the clause which immediately precedes the clause containing the ellipsis. This means that the strategy which consists in resolving a VP ellipsis to the nearest preceding VP is inadequate (following Klein and Stainton-Ellis, I will refer to this strategy as RNA (Resolve to Nearest Antecedent)). The rest of this section illustrates this point by means of examples, many of which are taken from [Klein/Stainton-Ellis 1989].

Although Klein and Stainton-Ellis stress that RNA is particularly problematic for cases which involve the existence of multiple VP ellipses, the following example shows that a single VP ellipsis might also be wrongly resolved if RNA is adopted.

(6.2) “[1 would be cold and dignified], and point out that during my year in the lower sixth I had not been considered worthy, and that, frankly, I did not feel that I had altered radically now I was in the upper sixth. Hence there could be no justification in conferring the dignity upon me now. But I wasn’t θ₁. I just mumbled ‘Yes, Sir’ when he asked me.” [Garford 1962]

Here there are six intervening VPs between the elliptical VP and its antecedent so clearly, RNA would be inappropriate. A similar example is given in (6.3) where the elliptical VP is separated from its antecedent by one intervening VP sack him.

(6.3) If you don’t [1 really like him], you can sack him. But I’m sure you will θ₁.

Another typical instance of RNA violation is provided by cases where the elliptical VP precedes (rather than follows) its antecedent as in (6.4) below.

(6.4) If I can θ₁, I will [1 go running].
As indicated by the data in (6.5), backward VPE is subject to the same constraint as that suggested for pronominal anaphora in [Wasow 1972]: if the anaphor precedes its antecedent and the antecedent is in the same sentence, then the anaphor must be (syntactically) more deeply embedded that its antecedent (this constraint is called the backwards antecedent constraint (BAC)).

(6.5)  

a. Jon [i went running] after Bill did \( \theta_1 \).  
b. *Jon did \( \theta_1 \) after Bill [i went running]  
c. After Bill did \( \theta_1 \), Jon [i went running]  
d. After Bill [i went running], Jon did \( \theta_1 \).

Note that there is no obligation for a VP ellipsis occurring within the first conjunct of such a structure to resolve to the VP of the second conjunct. It can in fact refer back to some VP occurring in the preceding discourse as is illustrated by (6.6).

(6.6) You’ll [i really like him]. If you don’t \( \theta_1 \), you can sack him. But I’m sure you will \( \theta_1 \).

A last case of RNA violation involving a single VP ellipsis is given in (6.7) below.

(6.7)  

1. Jon prepared his exam thoroughly.  
2. He decided not to go on holiday.  
3. He spent the whole week studying.  
4. Every day he went to bed early.  
5. Bill did \( \theta \) too.

Here the VP ellipsis in (5) may refer to the conjunction of the VPs in (2, 3, 4) or in (1, 2, 3, 4) or to the VP in (4). RNA would fail in that it would only account for the last of these three readings.

Other cases of RNA violations involve the existence of multiple VP ellipses. Two main dependency patterns between VP ellipses and antecedents can be distinguished: crossing and nesting. These can be represented as follows.

Crossing: \( \text{VP}_i, \text{VP}_j, \theta_i, \theta_j \)
Nesting: VP, VP; 0j 0, Both patterns are problematic for RNA. In the case of crossing, the first VP ellipsis (θ₁) is separated from its antecedent by the second VP (VP₂) and the second VP ellipsis, by the first VPE: θ₁. In the case of nesting, the first VP ellipsis can be resolved by RNA but not the second because it is separated from its antecedent by θ₂. Examples of such dependencies are reproduced below from [Klein/Stainton-Ellis 1989].

(6.8) If he [1 is lucky] he [2 has bought his software from a house that [3 can help (at a price of course)]]. If he hasn't θ₂, he isn't θ₁, and may the force be with him because he will need it. (Nesting)

(6.9) It [1 was of course preposterous]; nothing so elementary [2 could possibly work]; there [3 must, one [4 would think], have been some other precautions]. One [5 would be wrong]. There weren't θ₃ and it did θ₂. (Nesting)

(6.10) I [1 promised myself I [2 wouldn't go to Manchester]] unless I first [3 opened a big stack of mail]. I didn't θ₃, so I didn't θ₂. (Nesting)

(6.11) If you [1 work hard, make the right choices and keep your nose clean], you [2 get ahead]. If you don't θ₁, you don't θ₂. (crossing)

(6.12) Xenophobia pestis, like the hard native perennial it is, bourgeons as lordly young Mediterranean male cyclists sail into oncoming traffic with such signorial arrogance that even as we swear and skid, we look round wildly for street signs to see if he'[1 s right], and we [2 are wrong] and the one-way system's [3 undergone one of its periodic reversals]. (He isn't θ₁. We aren't θ₂. It hasn't θ₃.) (crossing)

As the following example shows, still more complicated patterns are possible.

(6.13) I was [1 really thin] then, and I [2 tried some ski-pants that [3 looked really good on me], and I [4 should have bought them]. But I didn't θ₄, and now I'm not θ₁ and they wouldn't θ₃.
In summary, the above data clearly indicates that RNA is an inadequate resolution procedure for VP ellipsis. The antecedent VP need not occur in the clause immediately preceding the ellipsis. It can occur immediately after the ellipsis in certain cases, in other cases it can occur arbitrarily far away within the discourse preceding the ellipsis. In contrast with these observations, it is interesting to note that despite the long distance character of VP ellipsis, there is in fact little room for ambiguity when resolving VP ellipses. Providing that the discourse is coherent, the hearer will usually recover the intended antecedent. This suggests that some strong constraints come into play to help the hearer resolve adequately. I argue that these constraints essentially stem from the structure of discourse. The claim I make is that the antecedent of a VP ellipsis must occur in a discourse segment which is related by some discourse relation to the clause containing the VP ellipsis. To make this claim precise, a theory of discourse structure must be defined which gives some content to the notions of discourse segment, discourse relation and discourse structure. Such a theory is developed in section 5.2. Section 5.3 shows how this theory explains the discourse constraints on VPE which we just discussed.

6.2 Discourse structure and Anaphora resolution

6.2.1 Discourse theory: the state of the art

An assumption shared by many existing theories of discourse is that discourse structure constrains anaphora resolution (cf. [Grosz 1977 and Grosz/Sidner 1986] for definite NPs, [Nakhimovsky 1988] for temporal anaphora and [Webber 1990] for deictic pronouns). This in turn implies (i) that discourse has a structure, (ii) that this structure can be defined and (iii) that this structure can be used to derive constraints on anaphora resolution which explains the data.

As it turns out, the literature on discourse shows that there are almost as many ways of defining discourse structure as there are theories of discourse. For instance, in [Grosz/Sidner 1986], discourse is taken to consist of clauses, each of which is characterised by a given discourse purpose, and the resulting structure is taken to represent the intentional structure of discourse. In contrast, [Cohen 1987] suggests that discourse is organised along two distinct but complementary dimensions: the linear ordering of utterances on the one hand and a dominance relation between utterances such that if X is subordinate to Y, then X provides evidence for Y on the other. This time the resulting structure is taken to be a reflex of the argumentative structure of discourse.
Other candidates for the nature of discourse structure include Reichman’s structuring of discourse into context spaces (cf. [Reichman 1985]), Levin and Moore’s partitioning of discourse into functional units where each unit is related to a goal of the speaker(s) (cf. [Levin/Moore 1980]) and Polanyi and Scha’s tree structuring where various discourse relations are taken to connect discourse segments into bigger discourse units to form a tree (cf. [Polanyi/Scha 1984]).

So there are many views of what discourse structure is. However, only few of the theories of discourse mentioned in the preceding paragraph have something to say about the interplay between discourse structure and anaphora resolution. One of the chief references with regard to this question is [Grosz/Sidner 1986]. Sidner develops a theory of discourse structure based on three interrelated representational structures: a linguistic structure, an intentional structure and an attentional structure. The linguistic structure reflects the surface realization of discourse. The intentional structure encodes the purpose of the discourse and of its constituting segments together with the relations which hold between these purposes. It is represented by a tree in which the parent-of relation indicates that the dominated node has for purpose to satisfy part of the purpose of the dominating one. The left-sibling-of relation indicates that the purpose of the left node must be satisfied before that of the right one. Finally, the attentional structure records the discourse entities that are salient at any point of the discourse and is represented by a stack of focus spaces. Each focus space is indexed with the purpose of the corresponding discourse segment. Restrictions on anaphora resolution result from the interaction which takes place between intentional and attentional structure. The nodes of the tree representing the intentional structure of discourse are linked to the focus space stack in a way which depends on the intentional relation holding between the two discourse segments considered. If the relation is one of dominance (the purpose of the new segment contributes to the purpose of the segment which immediately precedes it in the discourse), then the stack is pushed (the focus space of the current clause is added to the top of the stack). Any anaphor contained within the new clause must find its antecedent in the top of the stack or in some of the focus spaces occurring below the top of the stack. If on the other hand the relation between the two segments considered is one of precedence (the purpose of the new segment contributes to the purpose of some segment occurring higher up in the hierarchy), then the stack is popped till the focus space indexed by the purpose of this higher segment is found.

The basic content of such a strategy is that an anaphoric link between two discourse entities can only be established if 1) some intentional relation obtains between the current segment and some other segment and 2) the antecedent is accessible from this other segment, that is, the corresponding discourse entity is either in the stack space occupied by this segment or somewhere
below in the stack.

An alternative analysis of the influence of discourse structure on anaphora resolution is presented in [Polanyi/Scha 1984]. Under this analysis, a discourse grammar consisting of a set of rewrite rules allows for a discourse tree to be constructed which reflects the structure of discourse. The rules ensure that each node in the tree has a given anaphoric potential. This anaphoric potential is the set of potential antecedents accessible at that node. This corresponds to the second fact encoded by Sidner’s system: anaphors can only refer to discourse entities which are accessible from a given segment in discourse. The second fact (that anaphors can only occur between segments that are connected by some discourse relation) is encoded by the fact that not all nodes in the discourse tree are available for extensions. Only some are: those that belong to the right-frontier \(^2\) of the current discourse tree.

So although there are many theories of discourse structure to account for the notion of discourse structure, there are only two main theories of discourse which account for the interplay between discourse structure and anaphora resolution. Like most theories of discourse, these two theories rely on the assumption that discourse has a hierarchical structure where the building blocks, called *discourse segments*, are connected together by some kind of *discourse relations* to form bigger units. Note though that again, there are many diverging views of what discourse relations are. Roughly, three types of approaches can be distinguished on the basis of the three systemic meta-functions proposed by the systemic school of linguistics ([Halliday 1985]) to explain how different linguistic devices are used to realize different goals. These meta-functions are:

- The **interpersonal** function which serves to establish and maintain the interaction between speaker and hearer.

- The **ideational** function which is concerned with ideation, that is, with the purely semantic content of language.

- The **textual** function which corresponds to the construction and interpretation of discourse as a structured unit.

Most discourse grammars integrate at least two of these three levels. However for each grammar it is usually the case that emphasis is placed on one particular aspect. In what follows, I give

\(^2\)The right-frontier of a tree is the set theoretic union of the root of this tree with the set of nodes which appears on the rightmost branches of this tree. For instance, the right frontier of the tree a(b, c(d,e)) is \{a,c,e\}
a general description of how emphasis on a particular meta-function affects the definition of discourse relations.

Approaches to discourse which are mostly concerned with the interpersonal function of language fall into three classes: those that are based on rhetorical relations, those that are based on intentional structure and those that are based on speech act theory. Rhetorical relations (or predicates) are used in [McKeown 1985; Mann 1984 and Polanyi/Scha 1984]. They indicate communicative functions such as identification, attribution, definition and are notoriously under-defined. For instance, in [Polanyi/Scha 1984], the rule for narratives states that a narrative may consist of a sequence of sentences $S_1 \ldots S_n$ such that the events $e_1 \ldots e_n$ introduced by these sentences are linearly ordered in time (i.e., $e_1 < \ldots < e_n$). Clearly, this is too naive. As shown in [Lascarides 1990], the temporal ordering of events does not always reflect the structure of discourse. For instance, the discourse in (6.14) is traditionally thought of as narrative (cf. [Dowty 1986]). However in this case the door opening is likely to follow the room being pitch dark thus contradicting the relative ordering of the corresponding sentences. So either Polanyi and Scha’s semantics for narrative is too naive or something must be added to the theory to explain why this discourse is not a narrative.

(6.14) Max opened the door. The room was pitch dark

Approaches based on speech acts include [Searle/Vanderveken 1985] and [Appelt 1985]. These are based on taking utterances as actions which have a definite purpose and are about propositions, that is, as speech acts. Following [Searle/Vanderveken 1985], conditions of use for speech acts can be defined so that the problem addressed by discourse grammars of this type is to identify what makes a string of speech acts coherent. There are several problems with this global approach to speech-act sequencing. These are discussed at length in [Bowers/Churcher 1988]. Finally, in intentional theories of discourse such as [Grosz/Sidner 1986 or Reichman 1985], discourse segments reflects the intentional structure of discourse in that they correspond to a given discourse purpose. Here again, the weak point is in the recognition procedure: how and why a common purpose is recognised is very fuzzy. The list of possible relations between discourse units is arbitrary and criteria for membership are ad hoc.

Discourse theories based on the ideational function of language are few and far between. An exception is [Halliday 1967/1985]. Halliday argues that texture, that is text coherence, results from three interacting factors: cohesion, sentential texture and text macrostructure. Cohesion
and sentential texture are the levels at which the ideational function of language contributes to discourse structure. Cohesion follows from implicit and explicit relationships between discourse units such as anaphora, ellipsis, contrast etc. It creates a link between the ideational function of language and the structure of discourse in that anaphors create semantic links between discourse units thus contributing to the building of coherent discourse segments. Sentential texture involves thematic and information structure at the sentence level. Information structure indicates how the different tone groups of the utterance contribute to partitioning the information carried by this utterance into given and new information. It concerns spoken language and will not be investigated here. Thematic structure organizes the message carried by the utterance into a rheme (what the sentence is about) and a theme (what is predicated of the rheme). Thematic structure can be used to distinguish between different types of relation between discourse units. Suppose we have two adjacent clauses S1 and S2. According to Halliday, there are four possible relations which can hold between S1 and S2 (elaboration, continuation, resumption and topic shift) depending on their relative thematic structure. S2 will be a continuation of S1 if S1 and S2 share the same theme. S2 will be an elaboration of S1 if the theme of S2 is an element of the rheme of S1. If the theme of S2 is also the theme of some utterance not immediately preceding S2, then S2 will be a resumption on S1. Finally, if none of these conditions obtains, S2 will implement a topic shift in the discourse. One of the main problems with Halliday's approach is that it crucially relies on notions which are given no formal content: the notions of theme and rheme. As a result, it is impossible to precisely determine whether his theory makes any useful prediction. This is not to say of course that the theory is useless: it does provide some interesting insights on the nature of discourse and of discourse processing. Simply, these insights must be taken for what they are: insights, not predictions.

Discourse theories based on the textual function of language are plentiful. In fact this is one of the area in which most of the work on discourse analysis and most of the approaches mentioned above integrates some knowledge of the macrostructure of texts. This is because in general, it is easier to describe the structure of text globally. The use of global techniques allows for speedy advance and applications can be produced relatively fast which produce reasonably coherent text. These techniques are mainly concerned with encoding knowledge about the discourse structure of the various genres (narrative, sonnet, television drama, conversation etc.). By contrast, they have little to say about the linguistic structure of discourse and are thus of little use when dealing with anaphora.

In short, discourse theory yields no good account of what discourse relations are. In most theories, discourse relations are taken as primitives (cf. Reichman 1985 and Grosz/Sidner
1986]) and in those theories in which they are not, the way in which they are defined is either too naive (cf. [Polanyi/Scha 1984]) or crucially dependent on notions that remains undefined (cf. [Halliday 1967]). As it is not my purpose to propose a version of discourse theory that does account for a definition of discourse relations and as I am not convinced by the validity of the definition of discourse relations given by any existing theory, I will here adopt the position most usually taken which is that discourse relations are to be treated as primitives. As regards the particular theory of discourse used to explain the data in 6.1, I will use a framework derived from Polanyi and Scha's where the main innovations are (i) that I make no assumption about how discourse relations are to be defined and (ii) that I make some additional assumptions about the influence of discourse structure on anaphora resolution and in particular on VPE resolution. Section 6.2.2 makes precise the main assumptions I make about discourse theory, discourse structure and the interaction of discourse structure with anaphora resolution. Section 6.3 shows how these assumptions can be used to explain the data listed in section 6.1.

6.2.2 Polanyi and Scha revisited

Polanyi and Scha's theory of discourse was briefly sketched in section 6.2.1. In what follows, I develop a general framework for discourse analysis which is derived from Polanyi and Scha's and makes precise the set of assumptions on which the analysis of VPE resolution presented in section 6.3 relies.

Discourse structure.

Following Polanyi and Scha's proposal for a discourse grammar but abstracting away from the particulars of their definitions of discourse relations, I make the following assumptions about discourse structure:

A1. Discourse is a structured entity whose basic constituents (or segments) are clauses and discourse relations. These basic discourse constituents group together to form larger units, called complex discourse constituents.

A2. A discourse grammar describes the structure and the well-formedness of discourse and consists of a sentence grammar complemented with a set of discourse grammar rules. Given a particular discourse, the discourse grammar will either assign to each constituent in the discourse some structural and categorial information, or reject this discourse as ill formed.
The categorial information associated by the grammar with each discourse constituent results from the knowledge embodied in this grammar. Basic discourse constituents are described both by the sentence grammar and by the lexicon: whereas a clause will be assigned a category by the sentence grammar, a discourse relation will be categorised in the lexicon (Here I assume that categories are feature-structures containing information about the phonological, morphological, syntactic, semantic and pragmatic properties of the units being described). Complex discourse constituents are described by discourse grammar rules of the form

\[ X \rightarrow Y, Z, W \ldots \]

where \( X \) is the name of the constituent defined by the rule and \( Y, Z, W \ldots \) are either basic or complex discourse categories.

A3. The categorial structure of discourse can be represented in a discourse tree where each node represents a discourse constituent and is labelled with the categorial information associated by the grammar with this constituent. Typically, the leaves of the tree will be labelled with basic discourse categories containing information about clauses and discourse relations, whilst non-terminals will be decorated with complex discourse categories describing complex discourse units.

A4. Discourse relations divide into ideational (or semantic) relations and rhetorical (or pragmatic) relations. The set of rhetorical relations consists of two disjoint subsets: the set of thematic relations including continuation, elaboration, resumption and topic shift on the one hand, and non-thematic relations on the other.

By way of example, consider the discourse in (6.15).

(6.15)  (1) Jon likes Mary but (2) she doesn't like him.

Under the above assumptions, this discourse will be assigned a structure by the discourse grammar. Since I do not specify the content of this grammar, I can only hypothesise as to what this structure might be. Intuitively, we might think of the discourse in (6.15) as consisting of three basic discourse units: the two clauses indexed (1) and (2) and the contrastive discourse relation made explicit by the connective \textit{but}. In terms of Halliday's theory of information structure, we might furthermore think of these two clauses as being related by the rhetorical relation of elaboration (because the theme of (2) i.e., \textit{she} is part of the rheme of (1) i.e., \textit{likes Mary}). This structural information can be represented as in Figure 6.1 where 1 and 2 represent the
categories associated by the discourse grammar with clause (1) and (2) respectively whereas (1,2) represents the discourse unit associated with the complex discourse segment resulting from the application of some discourse grammar rule to the sequence (1,2). The discourse relations identified by the discourse grammar to hold between (1) and (2) are omitted from the tree but notated on its right handside, whereby the notation \(R(i_1, i_2)\) indicates that the discourse relation \(R\) holds between discourse segment \(i_1\) and discourse segment \(i_2\).

![Figure 6.1: A discourse tree.](image)

**Discourse structure and anaphora resolution**

In section 6.2.1, I present two models of how discourse structure interact with anaphora resolution: [Sidner 1982] and [Polanyi 1988]. Although these two frameworks differ in the details of their implementation, at a more abstract level, they share many common features. By abstracting away from implementation details, three common assumptions can be pinpointed. Firstly, each discourse segment is assigned a given anaphoric potential (in Sidner's analysis through the use of the stack, in Polanyi's through the use of the grammar rules). Secondly, only certain nodes in the tree are open for extension by incoming discourse. Intuitively, this models the fact that as discourse proceeds, some discourse segments become 'closed' in the sense that it becomes impossible to relate them to any new discourse segments. More specifically, a closed discourse segment is characterised by the fact that it can neither enter into an anaphoric link with subsequent discourse nor be related to new discourse segments by some discourse relation. Thirdly, anaphoric links may only occur between discourse segments which belong to the same super-segment. Or to put it another way, an anaphor may only occur across discourse segments if these segments are related by some discourse relation.

Drawing on these observations, I make the following assumptions about the interplay of discourse structure and anaphora resolution.

A5. Every discourse segment is assigned by the discourse grammar a universe and an anaphoric
potential. The universe of a discourse segment indicates the set of discourse entities which this segment makes available for anaphoric reference to subsequent discourse (that is, it is similar to the DRT notion of universe). In contrast, the anaphoric potential of a discourse segment is a set of discourse entities available for anaphoric reference at that point of the discourse.

A6. Only certain discourse segments in the tree representing discourse structure may be extended by oncoming discourse. That is, some nodes in the tree are open and can be related by some discourse relation to new incoming discourse segments whereas some other segments are closed and cannot be so related. Whether or not a particular segment is open depends on three main factors i.e., the position of this segment in the discourse tree (I will take nodes on the right frontier of the tree to be open), the particular discourse relation in which this segment is involved (for instance, we shall see in the next section that resumption allows for a new discourse segment to be related to a segment (the topic) which is not part of the tree right frontier) and the type of the discourse segment in which a particular discourse segment occurs (for instance, we will see in section (6.3) that discourse segments contained in complex discourse segments which are related either by parallelism or by contrast to some other complex discourse segment are open to the constituting segments of this other complex segment).

A7. An anaphor may only relate to a discourse entity occurring in the universe of some related discourse segment. Or equivalently, the referent of an anaphor must be provided by a discourse entity which is part of the anaphoric potential of the discourse segment including the anaphor.

This gives us a framework where discourse structure influences anaphora resolution. However, to account for the range of data presented in section 6.1, the notion of anaphoric potential needs to be further defined. This is the object of the remainder of this section. In what follows, I first review some general properties of VPE illustrated by the data given in section 6.1 and show how these affect the specification of the notion of anaphoric potential. I then complement the set of assumptions made so far with some additional assumptions concerning the notion of anaphoric potential.

According to A5, an anaphoric potential is a set of discourse entities available for anaphoric reference at any given point in the discourse. Just as anaphors may denote different semantic objects, these discourse entities may be of various semantic types. For instance, they may denote individuals but also properties or propositions. In order to simplify the exposition, I will
concentrate here on that part of the anaphoric potential which has to do with VP ellipsis and ignore every other type of anaphor (this means that when talking about anaphoric potential, I will only consider those discourse entities which provide a referent for VPEs). Furthermore I will assume that the discourse entities which provide an interpretation for VP ellipses are property type entities (thus, the part of the anaphoric potential I will be dealing with will exclusively consist of property type discourse entities).

When dealing with VPE, the following facts need to be taken into account in defining the notion of anaphoric potential.

First, the discourse entity which provides the interpretation of an elided VP is not as persistent as an individual discourse entity. Consider the following discourses:

\[(6.16)\]  
\[a.\quad \text{*Jon [reads]. Mary is a writer. Peter does } \theta_1 \text{ too.}\]  
\[b.\quad \text{Jon}_1 \text{ is a poet. Mary}_2 \text{ is an accountant. } \text{He}_1 \text{ does not like her}_2. \text{ She}_2 \text{ does not like him}_1. \text{ They}_1\text{+}_2 \text{ are each other's favorite enemies.}\]

Whereas in (6.16b), the pronouns he, her, she, him and they can "skip" over sentences to establish a link with their intended antecedent, in (6.16a), the elided VP cannot establish a link with the intended antecedent reads because of the intervening clause Mary is a writer. This is not to say that a VPE must refer to an antecedent occurring in the clause immediately preceding it. Indeed the data in section 6.1 shows that this is an untenable stance. Rather the point here is that there is a sense in which property type discourse entities seem to be less persistent than individual ones. Exceptions to this general lack of persistence can usually be accounted for not in terms of linguistic structure but in terms of discourse structure (that is, it is discourse structure and not linguistic surface structure which is decisive in deciding how far apart a VPE is from its antecedent). We model the lack of persistence of property type discourse entities by not percolating them up and down the discourse tree. That is, the discourse entity associated with a given VP remains local to the discourse segment that introduces it.

Second, I want to account for the fact that VP ellipsis can refer to a summation of VPs. The following discourse illustrates this.

\[(6.17)\]  
\[a.\quad \text{Jon went to the library.}\]  
\[b.\quad \text{He borrowed a book on computer science.}\]
c. Peter did too.

The VPE in (6.17c) refers either to *borrowed a book on computer science* or to the summation of *went to the library and borrowed a book on computer science*. To account for this fact, we assume that the universe of a complex discourse segment consists of the property formed by conjoining the properties introduced by the constituting sub-segments. Such a property, I call a *conjoined property*. So in the example above, some mechanism will allow the VPE in (6.17c) to resolve either to the property type discourse entity introduced by the VP *borrowed a book on computer science* or to the conjoined property *went to the library and borrowed a book on computer science*.

Finally, a third fact that needs to be taken into account is that of backward anaphora as in *If Jon does, I will go swimming* where the anaphoric potential of the first clause includes the discourse entity introduced by the VP of the second clause. This is implemented in the grammar by allowing information about anaphoric potential to be lexicalised. That is, connectives may specify how the universe and the anaphoric potential of the units they connect relate. For instance in the case of *if* and *when* we stipulate that the anaphoric potential of the first clause is that given by the discourse rules unionized with the universe of the second clause. Thus the sentence *If Jon does, I will go swimming* is acceptable because the discourse entity introduced by *will go swimming* is part of the anaphoric potential of the first clause (*Jon does*).

At first sight, it seems that this analysis of backward VPE could be extended to deal with backward pronominal anaphora as illustrated in (6.18).

(6.18) Before he₁ came in, the man₁ looked into the bar.

In fact however, the proposed approach does not generalise very well. For a start, it fails to account for all those cases of pronominal cataphora which do not involve a connective (because this approach crucially relies on the presence of a connective in the sentence involving the cataphora). Examples (6.19a) and (6.19b) illustrate such cases.

(6.19) a. Behind him₁, Peter saw a snake approaching Jon₁'s hand.

   b. Himself₁, Jon₁ doesn't like.
Furthermore, it would fail to account for intersentential cataphora as in (6.20).

\[(6.20)\quad \text{Jon looked at him}_1 \text{ as he}_1 \text{ came through the door. He}_1 \text{ looked tired. Jon knew then that Paul}_1 \text{ had lost.}\]

Finally, consider the sentence in (6.21).

\[(6.21)\quad \text{Before Jon did, the man}_1 \text{ saw that he}_1 \text{ was in danger.}\]

On the strict reading of this sentence, the antecedent VP contains a pronoun he\textsubscript{1} whose antecedent (the man\textsubscript{1}) appears after the VPE. Given the semantics of DIL, this means that the discourse in (6.21) cannot be interpreted because when the discourse marker d\textsubscript{1} representing the pronoun he\textsubscript{1} is interpreted in the first clause, the current state is undefined on d\textsubscript{1}. This together with the other examples above points towards a serious shortcoming of DIL (and DRT), namely that it doesn’t obviously provide a mechanism for dealing with cataphoric constructions. I leave as an open research problem the task of determining whether or not either of these formalisms can be extended to cope with cataphora.

To reflect these three facts about VPE resolution, I make the following additional assumptions:

A8. VPs introduce in the universe of discourse categories a VP type discourse entity which provide the basis for the interpretation of VPE\textsuperscript{3}. These VP type discourse entities remain local to the discourse segment that introduced them (i.e., they are not percolated by the discourse grammar rules).

A9. The universe (i.e., set of discourse entities) of a basic discourse constituent is a singleton set containing the VP type discourse entity introduced by this constituent. In contrast, the universe of a complex discourse constituent is a singleton set containing a VP type discourse entity whose semantics is formed by conjoining the semantics of the VP type discourse entities introduced by the sub-segments\textsuperscript{4}

\textsuperscript{3}Although I will not go into any detail here about the nature of these VP type discourse entities, it should be clear that to be consistent with the rest of this thesis they need to be VP signs (rather than e.g., semantic properties). That is, they are linguistic objects defined along several dimensions including in particular syntax and semantics.

\textsuperscript{4}Note that this definition says nothing about the syntax of such complex discourse entities. This I leave
A10. The anaphoric potential of discourse constituents is specified by the discourse grammar rules on the one hand and by the lexical entries associated with the various discourse relations on the other. With regard to VPE, the discourse grammar rules specify that the anaphoric potential of a given segment simply is the universe of the discourse constituent to which this segment is related. Some discourse connectives such as if and when specify the anaphoric potential of the segments they connect in such a way that backward anaphora becomes possible between the first and the second segment.

Here an example will help illustrating the basic intuitions underlying assumptions A8-A10. Consider the discourse tree in Figure 6.2.

```
(1,(2,3))
  /
 / \
1   (2,3)
 |
2  3
```

Figure 6.2: A discourse tree.

For this tree, the universe and the anaphoric potential of the various nodes in the tree are as follows (I use the following notational conventions. VP\(i\) denotes the VP type discourse entity introduced by clause (i). Similarly VP\(1\)∧VP\(j\)∧…∧VP\(n\) represents the discourse entity of a complex discourse segment whose subsegments are \((i, j \ldots n)\):

as an open problem. My intuition is that ellipses involving summation are more like cases of anaphors whose antecedent does not coincide with some string in the discourse and where resolution involve some inferencing. In this sense, VPE involving summation are pragmatic cases of ellipses which suggest a different treatment for these particular cases. See also chapter 8 for more examples of pragmatic VPEs.
This is assuming that connectives permitting backward anaphora (e.g., if, when) are not present.
If say, the discourse part (2,3) is of the form If $S_i$, $S_j$, then the anaphoric potential of (2) will
be \{ VP1, VP3 \} rather than just \{ VP1 \}.

The import of such restrictions on the availability of VP referents should become clear in section
6.3 where examples are discussed of discourse constraints on VPE resolution and of how these
can be represented within the present framework.

### 6.3 Explaining the data

In this section, I present some examples of how the theory of discourse sketched above explains
some facts about VPE resolution which to date have remained unanswered. These facts includes:
cataphora, summation, multiple ellipses and more generally, various discourse constraints operating
on VPE. I start with some examples of non-adjacent a-clauses.

**Example 1.** (1) I would [be cold and dignified], and (2) point out (3) that during my year in
the lower sixth I had not been considered worthy, and (4) that, frankly, I did not feel (5) that I
had altered radically (6) now I was in the upper sixth. (7) Hence there could be no justification
in conferring the dignity upon me now. (8) But I wasn't 0.

According to assumption A7, one restriction on anaphora resolution is that the e-clause must
be related by some discourse relation to the a-clause. This explains first, why the antecedent of
a VPE is easily identified (It is the VP contained in the discourse segment to which the segment
containing the e-clause is related by some discourse relation) and second, why the a-clause need
not be adjacent to the e-clause (VPE resolution is constrained by the structure of discourse not
by its surface realization). So in order to explain why the VPE in (8) refers to the VP of (1),
we need to consider the discourse tree associated with the whole discourse. A possible tree for
CHAPTER 6. DISCOURSE

example 1 is as indicated in Figure 6.3. We now need to state which nodes of this tree are open that is, which nodes can be related by some discourse relation to clause (8). According to assumption A6, open nodes are nodes on the right-frontier of the tree (i.e., (1, ..., 2, ..., 3, ...) and 7) and any node which is involved in a discourse relation that licenses extension of this particular node. Here consider node 1. Intuitively, if (1) represents the topic of the discourse (i.e., roughly what is being talked about), then (2, ...) is a continuation on (1). Typically (cf. Halliday's theory of rhetorical relations), topics are open for extension even when they no longer are on the right-frontier of the tree because they can be resumed. Thus in the present case, I will take the set of open nodes to include (1). The point is that now, we have several possibilities for attaching the discourse segment introduced by clause (8). That is, RNA need not be adhered to. Now assuming that (1) provides an adequate argument for the relation of contrast indicated in (8) by the connective but, we can hypothetise that (8) will be connected to (1) in the discourse tree rather than to the immediately preceding sentence in the discourse i.e., (7). Under the present theory of discourse, this means that the anaphoric potential of (8) consists of the singleton set containing the discourse entity introduced by the predicative in (1) i.e., cold and dignified. This explains why clause (8) resolves to But I wasn't [cold and dignified].

Example 2. If (1) you don't like him, (2) you can sack him. But I'm sure (3) you will.

Here (1) and (2) may be thought of as being related by a causal or conditional discourse relation. Supposing both (1) and (2) remain open for extension then (3) may be related either to (1) or to (2). But if (3) is related to (2), the resulting discourse constituent is You can sack him but I'm sure you will which sounds very odd. Possibly the right explanation for the oddness of such a discourse is that the connective but indicating a contrastive relation imposes some requirements on the clauses it connect which are not fulfilled here. In contrast, (1) does seem to fulfill these requirements in that if you don't like him, but I'm sure you will does appear to be a coherent discourse. Thus supposing that the discourse grammar provides an account of the linguistic properties of but, (3) will be attached to (1) rather than to (2). This would explain why the VPE in (3) resolve to like him rather than sack him.

---

5. This tree of course, depends on how discourse relations are defined. Given the minimalist framework assumed here the analysis I propose can only be stated at the intuitive level. Thus in the present case, all I can say is that intuitively the relations indicated in Figure 6.3 seem to hold between the various discourse segments.

6. Again what these requirements are very much depends on the content of the discourse grammar. Following [Asher 1990] however, we may for example require that two propositions related by a contrastive relation have opposite polarity. This in turn begs the question of how polarity is to be defined and this is in fact a difficult question. Rather than assume some ad hoc definition of polarity, I simply assume that but imposes some requirements on the surface realization of the propositions it connects and that these are not fulfilled. This explains why you can sack him but I'm sure you will sounds odd.
FIGURE 6.3: Discourse tree for Example 1.
Example 3. (1) Jon prepared his exam thoroughly. (2) He decided not to go on holiday. (3) He spent the whole week studying. (4) Every day he went to bed early. (5) Bill did that too. [Prüst/Scha 1990]

As observed in [Prüst/Scha 1990], the VPE in (5) may refer either to the conjunction of the VPs in the preceding discourse (θ = VP1 ∧ VP2 ∧ VP3 ∧ VP4) or to the conjoined VPs of the continuation on (1) formed by clauses (2), (3) and (4) (θ = VP2 ∧ VP3 ∧ VP4) or to the VP in the clause immediately preceding (5) (θ = VP4). Under the present theory, this is explained by the fact that the discourse constituent formed by clause (5) may be attached to several places in the discourse tree where each attachment corresponds to different anaphoric possibilities (because the anaphoric potential of a node in the discourse tree is essentially determined by the universe of its left sister).

Suppose then that the discourse tree for the discourse comprising clauses (1), (2), (3) and (4) is as in Figure 6.4. Supposing as in [Polanyi/Scha 1988] that only the nodes on the right-frontier of the tree are open, the discourse unit for (5) may then be related either to (1,(2,3,4)), to (2,3,4) or to (4). Under assumption A9, the universe of a discourse segment consisting of several sub-segments consists of a VP type discourse entity whose semantics is formed by conjoining the semantics of the VP type discourse entities introduced by the sub-segments. Thus, in the first case, the anaphoric potential will be the universe of (1, (2,3,4)) i.e, the singleton set containing the VP type discourse entity whose semantics is formed by conjoining the semantics of the VPs in (1), (2), (3) and (4). This yields the first reading: Bill prepared his exam thoroughly, decided not to go on holiday, spent the whole week studying and every day, went to bed early. In the second case, the anaphoric potential of (5) will be provided by the universe of (2,3,4). This yields the second reading: Bill decided not to go on holiday, spent the whole week studying and every day, went to bed early. Finally, in the third case, the anaphoric potential of (5) will be the universe of (4) i.e, the VP type discourse entity introduced by went to bed early every day. This gives the last possible reading: Bill went to bed early every day.

Let us now turn to cataphora. Consider the following example.

Example 4. If (1) I can, (2) I will go swimming.

Under assumption A10, discourse connectives may specify the anaphoric potential of the segments they take as arguments in such a way that the discourse entities introduced by the second
clause are available to anaphors appearing in the first clause. Since *if* is such a connective, the discourse entity introduced by the VP *will go swimming* in example 4 is accessible from (1). This explains why the VPE in (1) can resolve to the VP in (2).

**Example 5.** I’m sure (1) you’ll like him. *If* (2) you don’t, (3) you can sack him.

Here the first argument of *if* is related to the previous discourse segment by a contrastive relation. This allows for an anaphoric link between (1) and (2) so that the VPE in (2) may take as its antecedent the VP *like him*. That the VPE in (2) is not resolved to the VP in (3) is probably due to some pragmatic factor about which I have nothing to say.

In the above examples, we have shown that discourse structure can influence VPE resolution in various ways. This explains among others, why the antecedent of a VP ellipsis need not occur in the clause immediately preceding it in the discourse. I now turn to the question raised by multiple ellipses. Multiple ellipses are discussed in some detail in [Klein/Stainton-Ellis 1989]. The main problem with this type of data is that although there seems to be no clear-cut constraints on the relative position of the multiple ellipses and their antecedents, there is nevertheless very little room for ambiguity. That is, it is usually very clear what VPE refers to what VP. Since all configurations are possible (nesting, crossing and mixed), no configurational constraint will ever be useful. The analysis I propose is this. I take cases of multiple ellipsis to be cases where a discourse relation has scope over a complex discourse segment (CDS) rather than a basic one as examined so far. Furthermore, I assume that discourse connectives which scope over CDSs lexically determine the anaphoric potential of their various subconstituents. Once these assumptions are made, VPE resolution can be shown to fall out of the linguistic requirements of discourse connectives on the discourse segments they connect. As a case in
point, consider the set of examples given in section 6.2 and repeated here for convenience.

\[\begin{align*}
(6.22) \quad & \text{a. I promised myself I [1 wouldn't go to Manchester] unless I first [2 opened a big stack of mail]. I [2 didn't], so I [1 didn't].} \\
& \text{b. A: I gave her some questions to [1 ask you] if you [2 rang her].} \\
& \text{B: I [2 did] but she [1 didn't].} \\
& \text{c. It was preposterous. It [1 couldn't possibly work]. There [2 must have been some other precautions]. But there [2 weren't] and it [1 did].} \\
& \text{d. If you [1 work hard, make the right choices and keep your nose clean], you [2 get ahead]. If you [1 don't], you [2 don't].} \\
& \text{e. I was [1 really thin] then, and I [2 tried some ski-pants that [3 looked really good on me], and I [4 should have bought them]. But I [4 didn't], and now I [1 'm not] and they [3 wouldn't].} \\
& \text{f. Xenophobia pestis, like the hard native perennial it is, bourgeons as lordly young Mediterranean male cyclists sail into oncoming traffic with such signorial arrogance that even as we swear and skid, we look round wildly for street signs to see if he [1 's right], and we [2 are wrong] and the one-way system [3 's undergone one of its periodic reversals]. (He [1 isn't]. We [2 aren't]. It [3 hasn't].)}
\end{align*}\]

In these examples, the surface linguistic structure of the complex discourse segments obey no unique pattern. In (6.22a), (6.22b) and (6.22c), VPE and antecedents illustrate a nesting pattern, (6.22d) shows a crossing pattern and (6.22e) shows a mixed pattern. So any resolution process based on configurational notions is doomed to failure. However at the discourse level some regularity seems to hold. That is, all examples above are cases in which both the set of antecedent clauses and the set of anaphoric clauses form a complex discourse segment in the sense defined in the preceding sections. Furthermore in all cases, the two CDSs are related by a relation of contrast or of parallelism. Intuitively what seems to happen is that this relation between CDSs also constrains the relations between individual clauses. Thus each constituent in the first complex discourse segment maps onto exactly one constituent in the second, and furthermore, the internal relation connecting the constituents within the complex segment is preserved in the other segment. That is, the logical structure of the two discourse segments is the same. For instance, consider example (6.22a) with discourse tree as in Figure 6.5.

If we assume that the discourse relation (call it r1) grouping (1) with (2) on the one hand and (3) and (4) on the other is an ideational relation with semantics $r1(P, Q) = \neg P \rightarrow \neg Q$, then
we have that the relation structuring the two CDSs (1,2) and (3,4) is the same i.e, $r_1$. In such a case, we say that parallelism holds between the two CDSs (1,2) and (3,4). More generally, parallelism on CDSs can be defined as follows:

**Definition 6.1** Parallelism between complex discourse segments.

Given two complex discourse segments $CDS_1$ and $CDS_2$ of the form $(1_a \ldots 1_n)$ and $(2_a \ldots 2_n)$ respectively, parallelism holds between these two CDSs iff the discourse relation structuring $(1_a \ldots 1_n)$ is the same as the one structuring $(2_a \ldots 2_n)$.

The discourse relation of contrast can be similarly defined as follows:

**Definition 6.2** Contrast between complex discourse segments.

Given two complex discourse segments $CDS_1$ and $CDS_2$ of the form $(1_a \ldots 1_n)$ and $(2_a \ldots 2_n)$ respectively, contrast holds between these two CDSs iff the discourse relation structuring $(1_a \ldots 1_n)$ is the complement of the relation structuring $(2_a \ldots 2_n)$ (The notion of a complement relation is illustrated by some examples further below. Its exact definition will be left open however since as already mentioned, the question of how contrast it to be defined is a difficult one which could not possibly be given a satisfactory answer here).

Given these definitions, I can now formulate the following hypothesis about multiple ellipses.

Multiples VPEs are grouped into a complex discourse segment ($CDS_2$) which is either parallel to or contrastive with another complex discourse segment ($CDS_1$).
Furthermore, for each e-clause in \(CDS_2\), the a-clause is in \(CDS_1\) and both clauses cover the same argument role with regard to the discourse relation structuring the two CDSs.

Here some remarks are in order. First note that as already mentioned in section 6.3, the surface ordering of linguistic clauses need not reflect their semantic ordering. For instance, in *Jon fell. Max pushed him*, the discourse relation relating the two discourse segments is that of causality and the semantics of the resulting discourse segment is \(cause(e_2, e_1)\) where \(e_x\) denotes the main event introduced by the discourse segment \(S_x\). In this particular case, the surface ordering of clauses \((S_1, S_2)\) does not coincide with the semantic ordering of the corresponding events with respect to the connecting discourse relation. We say that the first discourse segment \((S_1)\) covers the second argument role of the *cause* relation structuring the complex discourse segment *Jon fell. Max pushed him*.

Note further that for certain relations the order of the arguments is irrelevant. Coordination sometimes is such a relation: *Jon runs and Peter walks has the same meaning as Peter walks and Jon runs*. I refer to as *symmetric* any relation for which the order of the arguments has no special significance. By contrast a *non-symmetric* relation is such that its arguments cannot be permuted without modifying the meaning of the global expression.

Let us now come back to the data above. (6.22a), (6.22b) and (6.22d) are examples of cases where the internal relations involved are non-symmetric. The table below indicates for each example the relationship between surface linguistic structure, discourse structure and anaphoric links. In the row ‘Linguistic structure’, the numbers refer to the relative position of clauses in the discourse and negation indicates whether or not the proposition is in the negative mood. The row ‘Discourse relation’ hypothesises on the relation \(r_i\) which cements each complex discourse constituents, while the row ‘Logical structure’ indicates how each basic discourse constituent contributes to the semantics of the complex discourse constituent in which it is included. I use the notational convention \(r_1^i\) to indicate that \(r_i^j\) is in some sense the complement of \(r_i\). Finally, the row ‘Anaphoric links’ indicates which clauses are anaphorically related where \(x/y\) indicates that clause \(y\) is anaphorically related to clause \(x\).
Examples (6.22b) and (6.22d) are cases where the CDSs are related by contrast whereas (6.22a) is a case of parallelism between CDS. The notion of logical structure underlying the table is only meant to illustrate the point that some semantic parallelism occurs between antecedent and anaphoric CDS. Let us now consider cases where the internal relation involved is symmetric as given in (6.22c), (6.22e) and (6.22f).

Here examples (6.22c) is a case of parallelism between complex discourse segments whereas (6.22f) and (6.22e) are cases of contrast.

These two tables provide more support for the informal hypothesis proposed above that the relation of parallelism holding between the two CDSs propagates down the discourse tree to the subconstituents in the sense where each subconstituent of the first CDS is anaphorically related to the subconstituent of the other CDS which plays the same logical role in the logical structure of the CDS. This is best illustrated by a diagram. Consider for instance example (6.22a). We can represent the various relations holding between the different parts of discourse by the diagram in Figure 6.6.

This diagram shows a simplified discourse tree in which integers stands for discourse segments and capital letters (A, B ...) indicate the argument role filled by this particular discourse segment. For instance, the leftmost node of the tree is labelled 1_B which means that the
semantics of the first discourse segment covers the second argument role of the relation \( r_1 \) structuring 1 and 2 into the complex discourse segment \( (1,2) \). The curved links between basic discourse segments indicate anaphoric links. The important point here is that (i) the two VPE clauses are contained within a complex discourse segment \( (3,4) \) which is parallel to another CDS i.e, \( (1,2) \) and (ii) anaphoric links only obtain between segments which cover the same argument role with respect to \( r_1 \) (2 and 3 cover the A-role of \( r_1 \) whereas 1 and 4 cover the B-role).

Using this representation scheme, the two other cases of multiples VPE involving non-symmetric relations (i.e, (6.22b) and (6.22d)) can be pictured as in Figure 6.7 and 6.8, where Figure 6.7 illustrates another case of nesting type dependencies between ellipses and antecedents whereas Figure 6.8 shows a case of crossing.

Similarly, the symmetric cases can be represented as shown in Figure 6.9, 6.10 and 6.11.

To conclude, it seems that some notion of mapping between CDSs is required to account for the facts just described about multiple VPEs. In the present grammar such a notion can be implemented at the level of attentional structure defined by the notion of anaphoric potential. Suppose that the universe of each subconstituent is percolated up to the top node of each CDS together with the name of the argument to which it corresponds. Then categories can refer to this information to encode constraints on VPE resolution. For instance, the category associated
Figure 6.7: Discourse tree for example (5.18b).

Figure 6.8: Discourse tree for example (5.18d).
Figure 6.9: Discourse tree for example (5.18c).

Figure 6.10: Discourse tree for (5.18f).
with the discourse relation of parallelism could specify that when the relation scopes over CDSs, the anaphoric potential of each subconstituent in the rightmost CDS equals the universe of the corresponding subconstituent in the leftmost CDS.

### 6.4 Summary

In this chapter, I have given some motivation for claiming that VPE resolution is constrained by the structure of discourse. I proceeded by assuming a general framework for discourse analysis derived from [Polanyi/Scha 1984]. With regard to VPE resolution, the core of the analysis is that discourse structure and not linguistic structure, defines what VPs are available for anaphoric reference at any point in a discourse. Once some general assumptions are made about the nature of discourse structure and on how this new notion of structure influences discourse anaphora, the data listed in section 5.1 to illustrate some unsolved puzzles about VPE resolution can be explained.

A caveat is in order here. Because I do not develop a discourse grammar as such, I cannot claim that constraints on VPE resolution fall out of the theory which would be embodied in such a grammar. More specifically, the main shortcoming of the proposed analysis is that it crucially relies on the notion of discourse relation and that this notion is given no formal content. I
believe this is unavoidable given the state of the art within discourse theory. Nevertheless, I hope to have given a reasonably convincing account of why it is reasonable to think that VPE resolution may be constrained by discourse structure.
Chapter 7

Computation

One of the motivations for choosing a unification-based grammar as a linguistic framework is that is easily implementable and that it allows for syntax and semantics to operate in parallel. In this chapter, I first discuss existing computational treatments of ellipsis and show that none of them account for the various linguistic constraints on VP and verbal ellipsis discussed in this thesis. I then go on to show how the analysis of ellipsis presented in this thesis can be implemented within a system where most of the syntactic and semantic constraints discussed in chapter 3, 4 and 5 are accounted for. In conclusion, I compare the proposed approach with the systems described in the first part of the chapter, discuss its advantages and shortcomings, and point to directions for further research.

7.1 Existing algorithms for ellipsis resolution

Computational approaches to ellipsis resolution fall into two main classes: a first class which concentrates on the syntax and semantics of ellipsis, which I will refer to as the 'linguistic approach', and a second class which is more concerned with the role of pragmatics in ellipsis resolution and which I will refer to as the 'pragmatic approach'. With regard to the specific problem of VP and verbal ellipsis resolution, three main approaches can be distinguished which either provide an account of at least one of these phenomena, or could be adapted to do so. These are described in [Webber 1978; Allen/Perrault 1978; Robinson 1981] — Webber's is a linguistic
approach and the other two are pragmatic approaches. In this section, I briefly review these approaches, whilst in the next section, I compare them with the type of approach I propose.

7.1.1 Pragmatic approaches

[Allen/Perrault 1978]: A pragmatic approach based on modelling goals

Allen and Perrault [1978] develop a model of dialogue understanding which handles ellipses that cannot be resolved on the basis of purely linguistic information. The main idea here is that knowledge of the speaker’s goal is crucial for language processing. For instance, if I am in a railway station and somebody asks me the following question:

(7.1) The 3.15 train to Windsor?

then I can infer from my knowledge about trains that the goal of the speaker is to know where the 3.15 train to Windsor leaves from. I can therefore reinterpret the question in (7.1) as meaning What platform does the 3.15 train to Windsor leave from? Thus, by knowing about the goals of the utterer, I can make the necessary inference to correctly interpret an elliptical sentence.

An interesting aspect of Allen and Perrault’s analysis is that it does not deny the role of syntax and semantics in resolving ellipsis. Rather, it takes pragmatics to be an efficient means of restricting the search space of possible alternatives for ellipsis resolution. Another attractive point of this system concerns the integration of pragmatic information. With regard to VP ellipsis, an interesting issue would be to investigate whether the goal planning process underlying Allen and Perrault’s analysis could be used to help resolve cases of VPE in which the a-clause does not precede the e-clause. For instance, consider the discourse in (7.2).

(7.2) I am [VP, trying to [VP, get the 4.15 to London]]. If I do θ₂, will I be there by 6?

Here knowledge of the speaker’s goal could be used for resolution. This goal is indicated in the last clause where the speaker indirectly expresses his wish to be in London by 6. Given this goal, it is reasonable to assume that the VP ellipsis θ₂ resolve to VP₁ (i.e., get the 4.15 to London) rather than to VP₁ (i.e., try to get the 4.15 to London). With respect to the analysis
developed in chapter 6, an interesting issue raised by Allen and Perrault's approach is then, whether the goal-based approach could be used to explain the constraints on VPE resolution observed in chapter 6, section 6.1; and further, whether such an approach should either replace or complement a discourse grammar type approach.

On the problematic side, there are two main criticisms which can be levelled against Allen and Perrault's approach. Firstly, although they acknowledge the role of syntax and semantics in natural language processing, it remains unclear from their paper just how syntax and semantics interact with pragmatics. Secondly, it seems (although they do not go into great details about the linguistic part of their system) that the linguistic component of their system is somewhat rudimentary. As a result, it seems unlikely that most of the linguistic constraints on VP and verbal ellipsis discussed in this thesis can be accounted for. In particular, it is unclear first, how semantic phenomena such as sloppy identity and the alphabetic variance constraint can be implemented and second, whether the syntactic constraints (e.g., missing head restriction, major constituent restriction etc.) regulating VPE and gapping can be accounted for.

[Robinson 1981]: A pragmatic approach based on modelling focus

In [Robinson 1981], an approach is presented which reduces the search space for verb phrase referents by using the notion of focus developed in [Sidner 1979]. The idea behind using this notion of focus is that among the set of possible antecedents produced by a given discourse, some are more salient than others and therefore more likely to be anaphorically linked to some anaphor in subsequent discourse. Sidner developed an algorithm which given a piece of discourse, predicts an ordered list of discourse foci: these indicate in a preferential order the potential antecedents available for subsequent anaphors.

Robinson combines the notion of focus with that of task structure. The system she develops acts as an expert assistant participating in a dialogue about assembling an air compressor. The task structure of this specific domain is used to support natural language processing. It is modeled using a tree representing the relations obtaining between tasks and subtasks (dominance indicates that the mother node is a task whose accomplishment depends on the realization of the subtasks labelling the daughters nodes; precedence indicates the temporal ordering between tasks). This tree is then used as a top-down constraint when searching for the place in the dialogue structure where an utterance fits, and establishing the referent of anaphoric expressions. Each node in the tree is associated with an action schema which specifies the participants of that action as well as the preconditions for and the effect of performing the action. Sidner's notion
of focus is incorporated in the system in that each node in the tree is associated with a given focus space which is then used to resolve anaphors: the current focus is that associated with the node corresponding to the immediate next step in the task plan. Essentially, the focus shifts with the task being discussed. Any object referred to anaphorically in the discourse is likely to be in the list of foci available at that point in the task structure. Robinson refers to focus and task structure constraints as contextual constraints. The system also allows for syntactic and semantic constraints (called utterance constraints) to be used bottom-up while searching for an appropriate antecedent.

Robinson's system is attractive in that it allows for the integration of task structure, focus structure and some syntactic and semantic knowledge. However, as in the Allen and Perrault treatment, the main problem is that the role played by the linguistic component is rather minimal. In effect, utterance constraints use linguistic knowledge to guide the search. For instance, if the ellipsis is in the present tense and progressive aspect then the referent must be provided by an action in the goal plan which has been previously mentioned but only just started. By contrast, if the ellipsis is in the past tense, the most recent direct goal will be checked first. As these examples illustrate, the nature and use of the linguistic knowledge embodied in Robinson's system are somewhat idiosyncratic and again, it is unlikely that the system can be extended to account for such fine-grained constraints as are discussed in the present thesis.

7.1.2 A linguistic approach

Webber [1978] proposes a unified account of the resolution of various types of discourse anaphora including definite NPs, pronominal anaphora, one-anaphora and VP ellipsis. One characteristic of the system she proposes is that it is serial. That is, resolution is taken to proceed in a production line fashion where several stages are necessary for completing the task and these stages have to be realized in a given order. More specifically, Webber's system comprises four successive processing stages, which can be summarised as follows.

Stage 1: The incoming sentence is input to the parser, which returns a phrase-structure tree for this sentence.

Stage 2: This phrase-structure tree is then input to a first interpreter, called the level 1 interpreter, which produces a semantic representation for the sentence using information both from the parse tree and from the lexicon. This first semantic representation (called the level
1 interpretation) is an intermediate representation in the sense that it does not contain all the information necessary for resolving anaphors.

Stage 3: The level 1 interpretation is mapped into a level 2 interpretation where pronouns, definite NPs, quantifier scope and elided VPs are resolved. This provides the basis for the construction (in stage 4) of a discourse model, which Webber takes to be central to anaphora resolution: this discourse model contains a record of the entities that a discourse or text makes available for anaphoric reference.

Stage 4: The level 2 interpretation is used to create the entities evoked by the input sentence; these will then be used for resolving anaphors occurring in subsequent discourse.

In this system, then, anaphors are resolved by searching the discourse model for an appropriate entity where appropriateness depends on the type of anaphor to be resolved. With respect to VP ellipsis, Webber lists ten conventions which she claims must be observed to derive all and only the appropriate entities for a given instance of VP ellipsis. The most important ones are that the antecedent of the VPE must be a property-type discourse entity, and that the subject of the elided VP should fill the same logical role with respect to the ellipsis trigger as the trigger's own subject. The latter restriction is added to ensure that the voice of the ellipsis is the same as that of the antecedent. Note also that sloppy/strict ambiguities are dealt with in the transition from level 1 to level 2 interpretation, whereby any pronoun appearing in the antecedent VP which is anaphorically related to the subject NP will trigger the formation of two discourse entities: one property in which the pronoun is represented as a bound variable, and another where it is coreferential with the subject NP.

The main criticism that can be leveled against Webber's analysis concerns the serial architecture of the system. The argument is both linguistic and psycholinguistic. From the psycholinguistic perspective, we have already seen that there is some evidence for thinking that language processing is an incremental process involving the concurrent action of several sources of information (cf. [Swinney 1979; Marslen-Wilson/Tyler 1980 and Altmann 1986]). In particular, there is some evidence that syntax and semantics operate in a parallel rather than in a serial fashion. From the linguistic point of view, the main problem is that syntax and semantics are completely dissociated from each other. But as we have seen (cf. chapter 3, 4 and 5), there are many cases of VP ellipsis where the correlation between syntax and semantics cannot be neglected. For instance, in the case of pseudo-gapping or of antecedent contained deletion, the antecedent must be a VP 'missing' an NP (either because of the linguistic context in which the
VPE appears as in the case of pseudo-gapping, or because of a syntactically controlled gap as in the case of antecedent contained deletion). In this case, the semantics of the ellipsis will be a relation rather than a property. The important point is that the appropriate relation must be identified on syntactic grounds: it must be the semantics of a VP missing an NP, not say of a sentence missing an NP. More generally, the serial use of syntax and semantics makes it hard to imagine how Webber's approach might be extended to deal with gapping, where the interplay of syntax with semantics is particularly strong.

7.2 A UCG-based approach to ellipsis resolution

The main shortcomings of the systems just described can be summarised as follows. In Webber's system, very little is said about the role of pragmatics and the serial architecture of the system is linguistically and psychologically implausible. In the two pragmatic systems, the role of syntax and semantics is under-specified and the interaction of these modules with pragmatics is minimal. But as we have seen, one of the main characteristics of verbal and VP ellipsis is precisely that syntax, semantics and pragmatics all interact to determine the acceptability and the resolution of ellipses. Moreover, there is psychological evidence that the human sentence processing mechanism use semantic information well before the end of a sentence. This suggests that an incremental, multi-level process of interpretation is taking place in which the various levels of linguistic and extra-linguistic knowledge concur to assign the utterance an appropriate meaning. Now as mentioned in chapter 2, one of the motivations for adopting a unification-based grammar as a linguistic framework is that it allows for such a multi-level process to be modeled on a computer. In what follows, I show how the linguistic analysis of gapping and VPE proposed in this thesis can be implemented — I proceed by first giving a general overview of the algorithm determining ellipsis resolution. To illustrate the workings of this algorithm, I then run through a simple example\(^1\).

7.2.1 Resolution

A computational treatment of VP and verbal ellipsis should fulfill two tasks. Firstly, it should implement the various linguistic constraints regulating the distribution of these ellipses (this determines the acceptability of ellipses). Secondly, it should provide a mechanism by which el-

\(^1\)A more detailed description of the PROLOG implementation of UCG and the resolver is given in the Appendix.
Ellipses will be resolved (this determines the interpretation of ellipses). Traditionally in linguistics, these two tasks are kept separate: linguistic constraints are viewed as being part of the linguistic knowledge (or competence) of native speakers, whereas the resolution process is seen as part of a more general process (or performance) by which speakers make use of their linguistic knowledge.

So far, I have mainly been concerned with describing the linguistic constraints regulating the distribution of VP and verbal ellipsis and with modelling the speaker's competence. Here, I turn to model her performance, that is, I describe the process by which competence is used. One of the characteristics of the implementation I propose is that it respects the theoretical separation made in linguistics between performance and competence: whereas the grammar encodes the competence of an idealised speaker, the resolver implements her performance. The remainder of this section describes this resolver.

It should be clear from chapter 6 that VPE is a discourse phenomenon. Similarly, gapping is shown in chapter 3, section 3.1.1 to occur across speakers in discourse. In other words, both VPE and gapping can be viewed as discourse phenomena. Thus here I take the view that ellipsis resolution is part of the discourse parsing process. This raises the question of how discourse parsing is defined. Ideally, discourse parsing, like sentence parsing, should be a process by which the linguistic knowledge embodied in the (discourse) grammar is put to use with respect to a given string. That is, given a discourse grammar DG and an input string String, the parser should assign a categorial structure to String, that reflects the linguistic knowledge encapsulated in the grammar. However, as discussed at length in chapter 6, I do not develop a discourse grammar as such (rather I make some general assumptions about discourse structure and its interaction with anaphora resolution on the basis of which, I then proceed to give a tentative explanation of constraints on VPE resolution). So, in implementing a resolver for VP and verbal ellipsis, I do not model discourse structure. This means that the intuitive explanation of VPE resolution developed in chapter 6 is not integrated in the resolver. Given this caveat, the basic algorithm for VP and verbal ellipsis resolution can then be described as follows.

For each discourse segment $DS_i$ with UCG category $Sign_i$, the discourse parser will have constructed a discourse edge $Edge_i$ of the form $[S_i, E_i, Sign_i, Tree_i]$ where $S_i$ and $E_i$ specifies the starting and the ending vertices of the edge and $Tree_i$ is the derivation tree associated by the grammar with $DS_i$.

Given a pair of discourse segments $DS_1$ and $DS_2$ with edges $Edge_1$ and $Edge_2$ respectively, the basic resolution algorithm is as follows. First, the anaphoric status of each edge is assessed.
• If $DS_1$ is a sentence and $DS_2$ contains a VP ellipsis, then resolve($Edge_1$, $Edge_2$, upe).

• Otherwise, if $DS_1$ is a subordinated sentence which contains a VP ellipsis and $DS_2$ is a sentence, then resolve($Edge_2$, $Edge_1$, upe).

• Otherwise, if $DS_1$ is a sentence and $DS_2$ contains a gapped clause then resolve($Edge_1$, $Edge_2$, gapping).

• Otherwise, resume discourse parsing.

The first two clauses deal with cases where a VP ellipsis is part either of the second or of the first string. The second case caters for cases of backward anaphora, where the backward anaphora constraint mentioned in chapter 6, section 6.1, must be enforced (hence the additional restriction that the VPE clause must a subordinate clause; alternatively, if a 'real' discourse grammar were used, this restriction could follow from the linguistic information associated with subordinating words such as and when). The third clause deals with cases of gapping. The predicate resolve($Edge_a$, $Edge_e$, AnaphorType) is then defined as follows.

• If $AnaphorType = vpe$ and $Sign_e:disgap = VP_e$, then (1) define the three functions $\mathcal{D}$, $\mathcal{D}^{-1}$ and $\mathcal{R}$ as indicated in chapter 4, section 4.2.2 and (2) unify $VP_e$ with some VP in the derivation tree of the a-clause.

• If $AnaphorType = gapping$ and $Sign_e:sem = and(Sem_e,R(Sem_e))$, then (1) define the three functions $\mathcal{D}$, $\mathcal{D}^{-1}$ and $\mathcal{R}$ as indicated in chapter 4, section 4.2.2, (2) substitute the categories of the gapping remnants into the derivation tree of the a-clause thus yielding a tree ($Tree_e$) for $DS_e$, and (3) unify $Sem_e$ with the semantics of the UCG sign associated with the mother node of $Tree_e$.

• Otherwise fail.

### 7.2.2 A sample derivation

In this section, I run through some simple examples and show how gapping and VP ellipsis are analysed within the framework developed in the preceding chapters. The aim here is to give a clearer picture of how the different sources of linguistic information interact to determine the acceptability and the interpretation of an ellipsis. That is, the examples are meant to illustrate first, how the various linguistic constraints regulating VP and verbal ellipsis interact within
the present grammar and second, how resolution occurs. I start with an example involving VP Ellipsis. As we have seen, there are both syntactic and semantic constraints on the distribution of VP ellipsis. At the syntactic level, the voice and the aspect of the antecedent VP must obey the requirements set by the auxiliary contained in the e-clause; on the semantic level, the semantic representation of the ellipsis must be an alphabetic variant of the semantic representation of the antecedent. Furthermore, this semantic representation must be reconstructed from the semantic representation of the antecedent VP via the resolution function R. From the resolution point of view, the antecedent VP must be part of the derivation tree of a clause which is related by some discourse relation to the e-clause. How is all this implemented? Consider the following example:

(7.3) Jon is running but Peter is not.

First, the discourse parser will split this sentence into three sub-parts: *Jon is running*, *but* and *Peter is not*. The corresponding edges will be added to the discourse chart and resolution will occur when the edge for *but Peter is not* is combined with the edge for *Jon is running*. To see how the various constraints mentioned in the preceding paragraph are implemented, consider the signs associated with the various parts of the discourse. The sign for the auxiliary *is occurring in the e-clause* is as indicated in (7.4)².

²For simplicity, the MCD attribute introduced in chapter 3, section 3.1.2 is ignored here and in the following examples. Furthermore, to account for tense, aspect and modality, I modify the sem attribute to include the relevant attributes. The idea here is that following [Fenstadt et al, 1985], we can think of a feature structure as yielding a schema from which the semantics of the corresponding string can be derived (rather than its semantic representation per se).
This sign encodes most of the linguistic constraints on VPE summarised above: it requires first, that the antecedent VP bears an aspectual feature which is compatible with the auxiliary is i.e., progressive (this is encoded through the path equation \( \text{sem:disgap} = \text{progressive} \)); second, that the voice of the antecedent VP be identical with the voice of the VP in the e-clause (\( \text{synt:voice} = \text{sem:disgap:sem:voice} \)); and third, that the category of the antecedent VP be S\( \backslash \)NP (\( \text{sem:disgap:synt} = s[\text{VOICE}] \backslash \text{NP} \)). From the semantic point of view, the only constraint is that the semantics of the VPE be \( R(\text{Sem}2) \) where \( \text{Sem}2 \) is the semantics of the antecedent VP. That is, the semantic representation of the ellipsis must be reconstructed from the semantic representation of the antecedent. Nothing is said about the alphabetic variance constraint. This is because as we have seen in chapter 5, this constraint falls out of the semantics and thus need not be implemented at the representational level. To complete the implementation therefore, it would be necessary to implement a DIL\(_2\) interpreter which would check whether discourses
can be assigned a truth-value. In cases where the alphabetic variance constraint is violated, interpretation would fail and the sentence rejected as ungrammatical.

The signs for the a-clause *Jon is running* and the connective *but* are as represented in (7.5) and (7.6).

(7.5)

\[
\begin{array}{l}
\text{pho: jon + is + running} \\
\text{syn: s[active]} \\
\quad \text{aspect: progressive} \\
\quad \text{tense: present} \\
\quad \text{sem: modality: \bot} \\
\quad \text{formula: } \{d1/j\} \text{run}(j) \\
\quad \text{disgap: \bot} \\
\text{order: \bot}
\end{array}
\]

(7.6)

\[
\begin{array}{l}
\text{pho: but} \\
\text{syn: sent/} \\
\quad \text{syn: s} \\
\quad \text{sem: Sem1} \\
\quad \text{order: post} \\
\quad \text{sem:formula: but(Sem1,Sem2)} \\
\quad \text{order: OrderF}
\end{array}
\]

Assuming that the edge for *but* combines with the edge for *Peter is not*, we then get the following sign for the string *but Peter is not*: 
Next, the edge for *Jon is running* combines with the edge for *but Peter is not* and at this stage resolution occurs. Two tasks need to be performed: first, the parallelism between a- and e-clause must be established (to account for the possibility of a sloppy interpretation in the e-clause) and second, the antecedent VP must be identified. According to chapter 6, the antecedent VP must occur in a discourse segment which is related by some discourse relation to the a-clause. However, as already mentioned in section 7.2.1, this part of the analysis is not implemented because the theoretical foundations needed for determining which discourse relations hold between which segments are still too vague. Hence here I make the simplifying assumption that the a-clause immediately precedes the e-clause and the antecedent VP is any VP in the derivation tree of the a-clause which satisfies the requirements set by the *disgap* attribute. In the present case, this means that the antecedent will be the VP *running* with semantics $Sem_2 = \lambda x. \text{run}(x)$. $D$ and $R$ are then defined on the basis of the two derivation trees associated with the a- and the e-clause respectively. Once these two tasks have been performed, the semantic representation $\lambda P.\{d_2/p\} P(d_2)(R(Sem_2))$ of the e-clause can be obtained. In this particular case, it will be: $\{d_2/p\} \text{run}(d_2)$. 
Gapping is treated in much the same way. Suppose that the string to analyze is as in (7.8).

(7.8) Jon likes Sarah and Peter Mary.

Then, the three basic discourse edges will cover Jon likes Sarah, and and Peter Mary respectively. On application of the product rule, the sign associated with Peter Mary will be as in (7.9).

\[
\begin{align*}
\text{pho} &: \text{Peter} \\
\text{synt} &: \text{np} \\
\text{sem} &: \text{peter} \\
\text{order} &: \text{Order1}
\end{align*}
\]

\[
\begin{align*}
\text{pho} &: \text{Mary} \\
\text{synt} &: \text{np} \\
\text{sem} &: \text{mary} \\
\text{order} &: \text{Order2}
\end{align*}
\]

The sign for and which can combine with such category is thus the sign given in (3.33) repeated here for convenience.

\[
\begin{align*}
\text{pho} &: \text{and} \\
\text{synt} &: a/ \\
\text{sem} &: (\text{Sem1}, R(\text{Sem2})) \\
\text{order} &: \text{post}
\end{align*}
\]

\[
\begin{align*}
\text{pho} &: \text{Pho1} \\
\text{synt} &: a \\
\text{sem} &: \text{Sem1} \\
\text{order} &: \text{post}
\end{align*}
\]

\[
X^2^\ast
\]

where Sem2 is some unknown semantics which will be provided by the resolver (and is in effect, the semantics of the sign associated with the mother node of the reconstructed tree for the e-clause).

On application of the rule of functional application, the edge for and will combine with the edge for Peter Mary and the resulting edge will then combine with the edge covering the a-clause Jon likes Sarah. Again at this stage, parallelism is established and R is defined. Furthermore, the principles regulating the construction of derivation trees require that the categories associated
with the gapping remnants be substituted into the tree of the a-clause, or a-tree. In so doing, the syntactic constraints on gapping are checked and the semantics \( \text{Sem2} \) needed by the resolver for assigning an interpretation to the e-clause is recovered. The semantics \( R(\text{Sem2}) \) can then be computed so the meaning of (7.8) will be Jon likes Sarah and Peter likes Mary.

This concludes the illustration of how the different sources of information are made to interact within the present framework. The next section points to various shortcomings in the implementation and outlines some future directions for research.

7.2.3 Shortcomings and comparison with other approaches

In comparison with the systems described in section 7.1, a first advantage of the approach I propose is that it yields a very fine-grained linguistic analysis of VP and verbal ellipsis. In particular, it accounts for syntactic constraints on gapping such as the missing head restriction and the major constituent constraint, it yields a general treatment of sloppy identity and it can cater for donkey anaphors (because the semantics is a dynamic one). With the exception of sloppy identity, all of these phenomena are ignored by the systems described in 7.1. Furthermore, although sloppy identity is dealt with in Webber's system, this is done along lines inspired from [Sag 1980], which as discussed in chapter 4, section 4.2.3, is too restrictive (in particular, Sag's and consequently, Webber's approach cannot deal with sloppy identity involving gapping).

A second advantage of the present approach is that being based on a monostratal grammar, it allows for language processing to be an incremental, multi-dimensional process rather than a serial one. Consider Webber's system for instance. As we have seen in section 7.1.2, Webber's system encompasses four distinct stages. First (stage 1), the input string is parsed and a syntactic tree is produced. Second (stage 2), a level 1 interpretation is derived from this syntactic tree. Third (stage 3), the level 1 interpretation is transformed into a level 2 interpretation whereby this transformation ensures that VPE, pronouns, definite NPs and ambiguous quantifier scoping are resolved. Finally (stage 4), the level 2 interpretation is used to update the current discourse model with some new discourse entities which will then be used for the resolution of subsequent anaphors. Now, in the present approach these four stages are conflated into two: the parsing process and the resolution process. That is, because a multi-level grammar is used, parsing ensures that in effect the first two stages (syntactic and semantic analysis) are conflated into one. Resolution, which occurs at parsing time, implements the third stage (transformation from level 1 to level 2 interpretation) and because a dynamic semantics is used, stage 4 can also
be dispensed with (though admittedly, the semantics used cannot deal with as many types of anaphors as Webber's system can). Besides considerations of elegance and transparency, the main advantages of the more 'compact' type of approach to resolution permitted by a monstral grammar, there are, as already mentioned in section 7.1.2, two main motivations for adopting such an approach: first, it is psychologically more plausible and second, it is linguistically more adequate.

This is not to say, that the present approach resolve all problems. There are in fact several important shortcomings here. A first, most flagrant, shortcoming is that nothing is said about the role of pragmatic knowledge in general and of discourse structure in particular. As we have seen in chapter 6, this sort of knowledge is particularly crucial in accounting for constraints on VP and verbal ellipsis. Although, little work has been done to date on integrating pragmatic knowledge into unification-based grammars, some progress has been achieved in this direction which are described in [Elhadad 1990]. A possible further direction for research then would be to investigate how ideas from Elhadad could be used to augment the present implementation with such pragmatic knowledge as is needed for the treatment of ellipsis. With respect to VPE more particularly, another possible direction would be to adopt Polanyi's discourse grammar (cf. [Polanyi/Scha 1984]) and extend it to account for the discourse constraints on VP ellipsis observed in chapter 6 (with the caveats made in chapter 6, which is that the account of discourse relations provided by Polanyi and Scha is still somewhat rudimentary).

Knowledge about intonation also plays a major role in the interpretation of gapped clauses (cf. [Kuno 1976; Sag 1980 and Steedman 1989]). Again this raises the question of how knowledge about prosody could be integrated into the present framework. Here work by [Bird 1990] and [Scobbie 1991] indicates that attribute-value grammars can be made to integrate such knowledge. It would be interesting to examine whether their work can be used to account for the complex interactions of intonation, syntax, semantics and pragmatics which characterises gapping.

Thus in general, a direction for further research regarding computation would be to investigate how various linguistic and non-linguistic sources of information can be made to interact with an attribute-value grammar such as UCG(2).
In this thesis, I have examined in some detail two elliptical constructs: VP ellipsis on the one hand, and gapping on the other. As was the aim, I have developed a UCG(2) based analysis for these two phenomena which accounts for a reasonable range of data. In conclusion, I would like to take stock first of what has been achieved and second, of what has remained unsolved.

The results achieved by this thesis can be summarised as follows. Firstly, gapping and VPE have been shown to be subject to constraints from different linguistic levels. While the main emphasis has been on describing the role of syntax and semantics, the importance of pragmatics has also been addressed (though to a much lesser degree). Furthermore, I have shown that whereas syntax plays a crucial role in determining gapping acceptability (cf. chapter 3, section 3.1.2), VPE is more affected by semantics (cf. chapter 5). Similarly, I have given some evidence for the claim that whilst VPE resolution is constrained mainly by discourse structure (cf. chapter 6), the interpretation of gapping is mostly affected by processing and intonational factors (cf. chapter 3, section 3.1.3). In short, I have presented a range of facts which taken together all point towards some essential differentiation between VPE and gapping.

Secondly, I have shown how UCG(2) could be extended to deal with VPE and gapping. From the syntactic point of view, I have indicated how the categorial syntax embodied in UCG(2) could be extended to cover these two constructs. The question of how this approach compares with other existing approaches to gapping and VPE is discussed at length in sections 3.1.4 and 3.2.3 respectively. From the semantic point of view, I have investigated the issue of how a dynamic
semantics could be used to determine the interpretation of VPE and gapping thus highlighting two facts. First, a dynamic semantics allows for Sag's alphabetical variance constraint on VPE to simply fall out of the model theory; any syntactic check on the semantic representation is thus made unnecessary. Second, the discourse information modeled by a dynamic semantics introduces further constraints on the interpretation process which need to be taken into account when developing a proposal for interpreting elliptical constructs. That is, the proposal must take care that the information about discourse entities introduced by the a-clause is not lost when processing the semantics of the e-clause.

Thirdly, I have given a tentative explanation of how discourse structure constrains VPE resolution.

Fourthly, I have developed an implementation which first, reflects the content of the theoretical analysis developed in this thesis and second, indicates how the implementation of UCG(2) could be extended to cover gapping and VP ellipsis.

In the remainder of this chapter, I would like to discuss two questions which have not been addressed so far (or have only been partially addressed). First, how do VPE and gapping relate to each other? And further, how do VPE and gapping relate to other types of anaphors/ellipses? This question is addressed in section 8.1. The second question I would like to consider is that of remaining problems and directions for further research: What are the shortcomings of the present approach? What should be revised? What should be extended? etc. This question is discussed in section 8.2.

8.1 Ellipses and anaphors

In this thesis, I have studied in detail the linguistic constraints regulating verbal and VP ellipsis and proposed a unification-based, monostratal analysis of both phenomena. I have argued that although semantically, VPE and gapping have a lot in common, syntactically they are very different: whereas VPE involves no particular syntactic treatment, gapping on the contrary requires a treatment which is very special to this construction. This difference raises the obvious question: is there any justification for treating VPE and gapping differently? Or put another way, what is the crucial distinction which is to explain the diverging properties (cf. sections 3.1.1 and 3.2.1) characteristic of the two types of ellipsis, and the resulting differences in their syntactic
treatment? This naturally leads to a more general interrogation about the nature of VPE and gapping that is, the question of how VPE and gapping relate to other types of anaphors such as stripping, sentential-it, this-anaphora and pronominal anaphora. In this section, I propose a classification of anaphors which gives some explanation first of why VPE and gapping behave differently and second, of how VPE and gapping relate to other types of anaphora.

Two classification schemes for anaphors which include VPE and gapping are given in [Hankamer/Sag 1976; Sag/Hankamer 1984] on the one hand, and [Chao 1987] on the other.

Hankamer and Sag (henceforth, HS for short) argue for a typology which is based on the interpretive properties of anaphors. They distinguish between two types of anaphors: those that reference linguistic objects on the one hand and those that reference ‘more general sorts of mental representations’ on the other. In other words, there are anaphors which must reference a linguistic constituent and those that need not do. The latter they call deep (or model interpretive) anaphor, the former, surface anaphor (or ellipsis). HS further argue that this fundamental dichotomy explains why distinct types of anaphors have distinct properties. More specifically, they make the following claims:

Surface anaphors (VPE, sluicing, gapping, stripping) may not be pragmatically controlled: they require a linguistic antecedent. Their interpretation is determined by reference to the linguistic representation associated with the antecedent. This explains first, why surface anaphors typically require a certain parallelism in syntactic form between anaphor and antecedent and second, why they exhibit the missing antecedent phenomenon (see [Grinder/Postal 1971; Bresnan 1971]).

Deep anaphors (pronominal anaphora, sentential-it, null complement anaphora (NCA)) may be pragmatically controlled. Their interpretation is determined by reference to the interpretation of the antecedent i.e., some semantic object in the discourse model constructed by the discourse participants. As a result, the interpretation of a deep anaphor does not need to refer to any representation of the antecedent expression, be it syntactic or semantic. This explains why deep anaphors are subject neither to syntactic nor to semantic

---

1[Schachter 1977] and [Williams 1977b] contend that the classification proposed in [Hankamer/Sag 1976] is inadequate but as pointed out in [Sag/Hankamer 1984]. [Hankamer 1978] counters [Schachter 1977] while William’s proposal is shown to be untenable in [Sag 1979].

2HS describes as ‘pragmatically controlled’ (or deictic) any anaphor whose interpretation depends not on some linguistic antecedent, but on some aspect of the nonlinguistic environment. For instance, a deictic pronoun is a pragmatic anaphor.
HS’s classification is summarised in the following table:

<table>
<thead>
<tr>
<th></th>
<th>DEEP anaphor</th>
<th>SURFACE anaphor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anaphors</td>
<td>Pronouns, S-it, NCA</td>
<td>VPE, sluicing, gapping, stripping</td>
</tr>
<tr>
<td>Defining property</td>
<td>may be</td>
<td>requires a linguistic antecedent</td>
</tr>
<tr>
<td></td>
<td>pragmatically controlled</td>
<td></td>
</tr>
</tbody>
</table>

While on the whole HS’s classification is rather convincing (for a detailed argumentation of their claims, see in particular [Hankamer/Sag 1976]), their analysis leaves two questions unanswered. The first question is: Why do VPE and gapping have very different syntactic properties? (for instance, the syntactic distribution of a VPE clause relative to its e-clause is freer than that of a gapped clause). The second question is: Why is it possible for a VPE to have an inferred antecedent whereas this is impossible for a gapped clause? (Cases of VPE with inferred antecedent are given in (8.2) further below).

Now consider Chao’s analysis. Like HS, Chao proposes a dichotomic classification for anaphors. Unlike HS however, this dichotomy is not based on the way in which distinct types of anaphors are interpreted. Rather, Chao’s proposal is based on a syntactic distinction: that of whether or not the anaphor has a syntactic head⁴. Chao further argues that this dichotomy explains why the two groups of anaphors have distinct structural, distributional and interpretive properties. In particular, she claims that headless anaphors, notated H~, display a property cluster which is typical of anaphors (in the GB sense of the term) whereas H+ anaphors exhibit properties which make them more like pronouns. Briefly, these properties are as follows. First, in H~ ellipses, the missing material can be characterised in terms of discontinuous strings of constituents. This contrasts with H+ ellipsis where the missing material must always be defined as a single syntactic constituent. Second, whereas H+ ellipses are not subject to strong syntactic constraints (such as syntactic parallelism of a- and e-clause), H~ ellipses are. This is to be expected if H~ ellipses are thought of as patterning with anaphors since the latter require a local syntactic antecedent. Third, in case of H+ ellipsis, the ellipsis may precede (backward anaphora) or contain (antecedent contained deletion) the antecedent. This is not true of H~ ellipses and again is characteristic of pronouns rather than anaphors. Chao’s classification can be summarised as follows:

⁴The notion of syntactic head assumed by Chao is that present in GB.
Clearly, Chao's classification is well supported. However, like HS's it leaves some questions unanswered. The main problem is that as HS have shown, there are some important differences between VPE on the one hand and sentential-it and NCA on the other. In particular, VPE requires a certain syntactic parallelism between a- and e-clause whereas NCA and sentential-it do not. This is illustrated by the following examples:

(8.1)  

a. *Sue was flattered by Tom and Sarah did too. (VPE)  
b. Sue appears to be thought by everyone to be pregnant but I don't believe it.  
   ([Lakoff 1968])  
   $it = \text{Sue is pregnant}$  
c. A: I don't see why you even try to win he game.  
   B: Because, I am convinced that I'll succeed [Hankamer/Sag 1974]  
   NCA = in winning the game  

(8.1a) shows how a simple difference in voice rules out VPE (the a-clause is passive whereas the e-clause is active). This contrasts with (8.1b) and (8.1c) where the anaphor refer to semantic objects which do not correspond to any constituent in the a-clause. Thus in one case (VPE), the syntactic parallelism required between a- and e-clause is rather strong whilst in the other (NCA and sentential-it) it is non-existent. Whilst, in HS's analysis, this is accounted for by the fact that VPE is taken to be a surface anaphor whereas sentential-it and NCA are classified as deep anaphors, in Chao's classification VPE, NCA and sentential-it all belong to the same class and nothing explains why they have different properties.

In summary, although HS's and Chao's classifications are both well motivated and interesting, they nevertheless fail to account for some important differences between VPE on the one hand, and NCA and sentential-it on the other. To explain these differences, I propose a third classification which as we shall see, is in fact a synthesis of the first two. I start by examining

For a detailed presentation of evidence supporting these two claims, the reader is referred to [Hankamer/Sag 1976] and [Sag/Hankamer 1984].
precisely those properties which distinguish gapping from VPE on the one hand, and gapping from sentential-\textit{it} and NCA on the other.

Consider the discrepancies between gapping and VPE. These are ignored in HS but underlined both in Chao and in the present thesis and can be summarised as follows: first, the distribution of gapping is very constrained whereas that of VPE is not (cf. sections 3.1.1 and 3.2.1). Second, VPE may have a non-linguistic antecedent whereas gapping may not. This is illustrated by the following examples (taken from [Webber 1978]):

\begin{enumerate}
\item[8.2] a. Jon and Mary danced together last night and Mary wants to $\emptyset$ again tonight. \\
\hspace{1em} $\emptyset =$ dance with Jon \\
\item b. Jon wants to go to Africa and Mary wants to go to India but neither of them can $\emptyset$ because money is too tight. \\
\hspace{1em} $\emptyset =$ ?? do what s/he wants to do \\
\item c. Catch 22 is a book that Philip wanted to buy and he did $\emptyset$ when he was waiting at Kings Cross station. \\
\hspace{1em} $\emptyset =$ buy Catch 22
\end{enumerate}

In each case here, the antecedent of the VPE has to be inferred — that is, it does not form a linguistic entity. In other words, all these cases may be seen as cases of VPE having a pragmatic rather than a linguistic antecedent. So in what follows, I will assume that VPE may take either a linguistic or a pragmatic antecedent. Note that gapping always requires a linguistic antecedent (that is, examples such as (8.2) cannot be construed for gapping)\footnote{For more evidence supporting this claim, see [Hankamer/Sag 1976].}. Now consider the differences between VPE on the one hand, and sentential-\textit{it} and NCA on the other. One main difference is that if the antecedent of the anaphor is a linguistic one, then it is syntactically constrained by the syntax of the anaphor only in case of VPE. In particular, the antecedent of a VPE must satisfy (some of) the subcategorisation requirements of the VPE auxiliary and must agree in voice (active or passive) with it. This is not true of sentential-\textit{it} and NCA: whether their antecedent is provided by the pragmatic or the linguistic context is irrelevant; the syntax of NCA and of sentential-\textit{it} remains unaffected by that of the antecedent and vice-versa. The reason for this is quite simply that sentential-\textit{it} and NCA remain syntactically invariant whatever their antecedent is.
At this stage, three properties have been identified which summarise the main differences distinguishing VPE from gapping, NCA and sentential-it. On the basis of these properties, we can now define a new classification for anaphors which divides them into three distinct classes as indicated by the following table:

<table>
<thead>
<tr>
<th></th>
<th>Syntax (1)</th>
<th>Syntax (2)</th>
<th>Antecedent</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-it</td>
<td>-</td>
<td></td>
<td>P</td>
<td>1</td>
</tr>
<tr>
<td>NCA</td>
<td>-</td>
<td></td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>VPE</td>
<td>+</td>
<td>-</td>
<td>P &amp; L</td>
<td>21</td>
</tr>
<tr>
<td>Gapping</td>
<td>+</td>
<td>+</td>
<td>L</td>
<td>22</td>
</tr>
<tr>
<td>Stripping</td>
<td>+</td>
<td>+</td>
<td>L</td>
<td></td>
</tr>
</tbody>
</table>

The three distinguishing properties are characterised in the first three columns: Syntax(1) indicates whether or not the anaphor is sensitive to syntax; Syntax(2) indicates whether the distribution of the anaphor/antecedent pair is syntactically restricted or not; finally, the 'Antecedent' column indicates the type of antecedent required by the anaphor — an L marks anaphors which require a linguistic antecedent, P indicates a pragmatic antecedent, and L & P means that the anaphor may take either a linguistic or a pragmatic antecedent.

On the basis of this table, anaphors can be divided into three sub-groups: those that are not affected by syntax and whose antecedent is pragmatic (class 1), those that are affected by syntax but whose antecedent may be either linguistic or pragmatic (class 21), and those that are subject to even stronger syntactic constraints and consistently require a linguistic antecedent (class 22).

Now the question naturally arises of how this new classification relates to HS's and Chao's typologies. The resemblances and differences between the three classifications can be summarised as follows:

<table>
<thead>
<tr>
<th></th>
<th>HS</th>
<th>Chao</th>
<th>Proposal</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-it</td>
<td>Deep</td>
<td>H+</td>
<td>1</td>
</tr>
<tr>
<td>NCA</td>
<td>Deep</td>
<td>H+</td>
<td>1</td>
</tr>
<tr>
<td>VPE</td>
<td>Surf</td>
<td>H+</td>
<td>21</td>
</tr>
<tr>
<td>Gapping</td>
<td>Surf</td>
<td>H-</td>
<td>22</td>
</tr>
<tr>
<td>Stripping</td>
<td>Surf</td>
<td>H-</td>
<td>22</td>
</tr>
</tbody>
</table>

That is, the new partitioning turns out to correspond with HS's and Chao's classifications in
the following way. First note that the two main classes (class 1 and class 2 where class 2 is the union of class 21 with class 22) of my classification correspond to HS's Deep/Surface classification. Recall that the Deep/Surface dichotomy distinguishes between anaphors whose antecedent may be pragmatic on the one hand and anaphors that require a linguistic antecedent on the other, whereas class 1 and class 2 refer to whether or not the anaphor is constrained by syntactic factors. The connection between the two classifications is thus a natural one. Surface anaphors requires a linguistic antecedent and can thus be subject to linguistic and in particular, syntactic constraints (this explains why class 2 of my classification coincides with HS's surface anaphors). In contrast, deep anaphors have as antecedent some semantic object and thus cannot be syntactically constrained (this explains why class 1 of my classification coincides with HS's deep anaphors).

Now consider the sub-classification of 2 into 21 and 22. This time the distinction coincides with Chao's H+/H− dichotomy. That is, H− anaphors are those anaphors which are most strongly constrained by syntax (class 22 in my classification) and requires a linguistic antecedent whereas H+ anaphors are syntactically freer (class 21 in my classification) and may take a pragmatic antecedent.

So the relationship between the three classifications is a direct and simple one. Moreover, note that it also resolves the conflict raised by VPE between Chao's and HS's classification, thus making them compatible: class 21 covers precisely those type of anaphors (i.e, VPE) where HS and Chao differ (whereas the H+/H− distinction mostly coincides with the deep/surface contrast, concordance disappears on the subject of VPE which HS classify as surface anaphor and Chao as H+).

Since there is a direct relationship between HS's and Chao's classification on the one hand and mine on the other, the properties and predictions made by the first two classifications can be integrated in the new classification thus giving it more support. We can summarise the relation between anaphors, properties and anaphoric classes as follows:

Class 1: Sentential-it, Null complement anaphora

Syntax plays no role.

The antecedent may be pragmatic.

Class 21: VPE

Syntax may play a role (if the antecedent is linguistic). The anaphor has a syntactic head
which in the case of VPE we take to be the VPE auxiliary. If the antecedent is linguistic, it consists of a single constituent. The distribution of the anaphor relative to the antecedent is relatively free; in particular, the anaphor may either precede or contain the antecedent. The antecedent may be either pragmatic or linguistic. At the interpretive level, the anaphor behave more like a pronominal than like a reflexive.

**Cass 22:** Gapping, stripping.

Syntax plays an important role. The anaphor lacks a syntactic head. The antecedent may consist of several constituents. The distribution of the anaphor is strongly constrained. The antecedent must be linguistic. The anaphor behaves more like a reflexive than like a pronominal.

### 8.2 Further directions

In this section, I indicate further areas of research that follow from work done in this thesis. Section 8.2.1 focuses on semantic issues related to gapping and VPE which have not been addressed in the body of the thesis. Section 8.2.2 summarises the role of pragmatics and intonation in processing VPE and gapping.

#### 8.2.1 Semantic aspects

[Oehrle, 1987] discusses the following data:

\[(8.3)\]

\[\begin{align*}
    a. & \quad \text{Kim didn't play bingo and Sandy sit at home all evening.} \\
    b. & \quad \text{Kim didn't play bingo and Sandy didn't sit at home all evening.} \\
    c. & \quad \text{Mrs J can't live in Boston and Mr. J in LA.} \\
    d. & \quad \text{Mrs J can't live in Boston and Mr. J can't in LA.}
\end{align*}\]

The gapped sentences \((8.3a)\) and \((8.3d)\) can be interpreted in two different ways: one in which the aspects of interpretation associated with the auxiliary appearing in the a-clause have wide scope over the two conjuncts, and one in which these same aspects have narrow scope with respect to the connective \(\text{and}\). Under the first reading, \((8.3a)\) means: it is not the case that
Kim play bingo and Sandy sits at home all evening — that is, for (8.3a) to be true it must hold that either Kim plays bingo or Sandy sits at home all evening but the two propositions may not both be true. Similarly, (8.3c) only holds if it is not the case that both Mrs J lives in Boston and Mr J lives in LA. The narrow scope reading is the one where (8.3a) means: Kim didn’t play bingo and Sandy didn’t sit at home all evening and (8.3c) holds if Mrs J can’t live in Boston and Mr. J cannot live in LA.

Oehrle points out that the wide scope reading is difficult to treat compositionally because the surface ordering of words does not coincide with the semantic scope of the semantic element they contribute to the meaning of the sentence as a whole: the element that has wide scope over the two conjuncts is in fact properly contained within the first conjunct. But this objection is only valid if it is assumed that the scope of semantic operators such as modals, negations and quantifiers is to be directly calculated from the surface string, that is, if a strict version of compositionality is adhered to. If this restriction is relaxed and it is assumed (as is the case in most computational grammars) that some scoping algorithm will allow for all possible readings to be generated, then the problem can be solved.

There are several scoping algorithms which one could try to combine with UCG(2). Some perform scoping on the fly ([Pollard 1989b]), others do it all at once and take as input the semantic representation associated with a whole clause or sentence ([Hobbs/Shieber 1987; Keller 1987 and Lewin 1990]). The issue of how semantic operators should be scoped and under which constraints is a highly complex one and I will not attempt here to define a scoping algorithm for UCG(2). I will simply suggest that an algorithm along the lines defined in [Cooper 1990:188-192] for a modified version of HPSC could well be appropriate as it allows for quantifier scoping to be handled lexically and by means of unary rules. The basic idea there is that the semantics of NPs is quantified out of the current sentence semantics and stored as the value of an attribute called qstore. Various devices ensure that the value of qstore percolates up to the top of the tree so that the store of a clause will contain the semantics of all NPs occurring in this clause. Some unary rules then ensure that the semantics of these NPs are quantified in, either at VP or at sentence level, and in such a way that all possible scope ambiguities are accounted for. The reason I judge Cooper’s algorithm to be compatible with UCG(2), is that as UCG(2) it is strongly lexicalist so that all the required information can be encoded either in the lexical entries or in the unary rules triggering the quantifying in. More specifically, note that given a sentence formed out of two conjuncts, the unary rule for quantifying in may apply either at the clause level or at the sentence level. In the first case, the quantifiers have narrow scope with respect to the connective whereas in the second, they have wide scope over it. Providing that like quantifiers,
modals and negations are quantified out to be quantified in at the sentence level, then we get precisely what was needed for the examples in (8.3).

8.2.2 Pragmatic aspects

As we have seen, there are a number of ways in which pragmatics affect VPE and gapping. This section summarises these and points to directions for further research regarding the interaction of ellipsis with pragmatics.

With regard to gapping, two pragmatic factors intervene to determine the acceptability and the interpretation of the ellipsis: first, the informational structure of the a-clause and second, the processing strategy of speakers. The role of informational structure was first brought to attention in [Kuno 1976], where the claim is that sentences containing gapped clauses are only acceptable when uttered in a context establishing an appropriate topic or presupposition. For instance, the sentence in (8.4a) provides the appropriate context for (8.4b).

(8.4)  
(a) Who did what?
(b) Kim played bingo and Sandy sat at home all evening.

More generally, the claim is that the missing semantics in the gapped clause precisely corresponds with either the current discourse topic or some recent presupposition available from the context. As [Steedman 1989] pointed out, this is equivalent to saying that the missing information in the e-clause closely corresponds to the 'given' information of the a-clause (as defined in [Halliday 1967]), or the 'topic' in the Prague School sense of the term (cf. [Hajicova/Sgall 1987]).

Given these observations about the interaction of given/new information (or topic/comment) and gapping interpretation, the question naturally arises of how this interaction is to be accounted for in the grammar. This question is further complicated by the fact that gapped clauses are characterised by a strongly marked intonational pattern where gapping remnants and their a-correspondents bear focal stress ([Sag 1976; Russell 1987 and Steedman 1989]). Roughly, the interplay between intonational pattern, informational structure and gapping interpretation can be summarised as follows. The focal stress characterising the a-correspondents in the a-clause means that the topic (or given information) of the a-clause can be easily identified: it is that part of the utterance which bears no stress. Furthermore, this topic coincides with the missing
semantics in the e-clause.

In this thesis, I have given no account of how these different factors interact nor of how they relate to the other grammar components of syntax and semantics. Although some work has been done on the interaction of informational and intonational structure with respect to gapping (cf. [Russell 1987]) on the one hand, and on the interplay between semantics and informational structure on the other ([Steedman 1989; Russell 1987]), a lot more work need to be done to answer the following questions: What is the relative role of syntax on the one hand and informational structure on the other in determining the interpretation of gapped clauses? Should informational structure be reflected in the semantic representation of an expression? and if so, how is this to be integrated into a dynamic semantics? On a more general level, the question arises of how informational structure, intonational patterns and topic information are to be integrated into UCG(2). This I leave as an open problem.

Another pragmatic factor which influences gapping is processing strategy (cf. section 3.1.3). Although [Kuno 1976] proposes some principles designed to explain some preferences in the way gapped clauses are interpreted, more work needs to be done on identifying the general principles underlying processing strategy. This I will leave as a task for psycholinguists.

Now consider the effect of pragmatics on VPE. With regard to VP ellipsis, pragmatics intervenes in two distinct ways: first, as discussed in chapter 5, discourse structure constrains VPE resolution and second, as illustrated in (8.4) inferencing and world knowledge sometimes are needed to interpret VPE. Although chapter 5 attempts to provide an account of how discourse structure affects VPE resolution, it is clear that much work still has to be done on defining a theory of discourse which could precisely accounts for discourse phenomena in general and VPE resolution in particular. Similarly, the question of how world knowledge is to be represented and of how it is used in producing precisely those inferences which are need for language processing remains an open one.
Appendix A

Implementation

In this appendix, I briefly describe how UCG can be implemented in PROLOG. I then go on to show how this basic framework can be extended to integrate the syntactic and semantic analysis of verbal and VP ellipsis developed in chapter 3, 4 and 5 of this thesis. Finally, I point out some shortcomings in the present implementation and indicate how it could be improved.

Caveat: for simplicity, I will often conflate the formalism and its PROLOG representation when discussing the program. For instance, rather than saying that Sign is the PROLOG representation of a UCG sign, I will simply say that Sign is a UCG sign. It should be clear from the context which is being meant of the formalism or of the PROLOG code.

A.1 Implementing UCG(2)

In this section, I give a brief description of how I implemented the basic UCG(2) framework on the basis of which a multi-level treatment of VP and verbal ellipsis can be developed. A fuller, more comprehensive implementation of UCG(2) is discussed in [Zeevat et al 1987].
A.1.1 The lexicon

Since UCG(2) uses a system of closed types to define linguistic attributes, an obvious way of implementing UCG(2) signs is to model them using PROLOG terms. Type declarations are coded as follows:

```prolog
type_declaration(Type, Attributes).
```

where Type represents a type and Attributes is a list whose elements represent the attributes defined for that type. For instance, the type declaration for a UCG(2) sign states that any UCG(2) object of type sign is a feature structure with four attributes: pho, synt, sem and order. This is encoded in PROLOG as follows:

```prolog
type_declaration(sign, [pho, synt, sem, order]).
```

A series of type declarations defines for each attribute used within UCG(2) the type of this attribute. Given these type declarations, a UCG(2) sign is simply encoded as a list of four attributes pho, synt, sem and order whose structure is recursively constrained by their respective type declarations. For instance, the sign for *man* is encoded in PROLOG as follows:

```prolog
[man, noun, man/I, ]
```

which is the PROLOG encoding of the feature structure in Figure A.1. Note that the semantics

```
pho : man
synt : noun
sem : man/I
order : 1
```

Figure A.1: UCG sign for *man*.

of this sign is represented using unification rather than predication (that is, we have *man/I* rather than the expected $\lambda x.\text{man}(x)$. This is contrary to the claim, made in chapter 2, that
the semantics used is based on predication rather than unification. The reason for this is a purely practical one. The implementation described here is only intended to illustrate how the theoretical analysis of ellipsis developed in this thesis can be implemented. It pretends neither to exhaustiveness nor to efficiency. With regard to the semantics, I have opted for a unification-based approach simply because it is simpler and faster to implement, not because I believe it to be the right approach. In fact, it means that the treatment of constituent coordination (as opposed to non-constituent coordination) is ad hoc. However, for illustrating how my analysis of ellipsis works, it is sufficient.

A.1.2 The organisation of the lexicon

Since UCG(2) is a strongly lexicalist framework, it is important that the organisation of the lexicon be maximally efficient in avoiding redundancies and expressing linguistic generalisation. The way this is realised in the present implementation is by adopting a PATR-like architecture for the lexicon. This involves allowing for path equations, templates and lexical rules.

Path equations find a simple encoding in PROLOG which is best illustrated by an example. For instance, to indicate that the morphological case of a given sign $X$ is nominative, we simply write:

$$X: \text{synt:morph} \leftrightarrow \text{nominative}$$

which means that the sign $X$ has a synt attribute whose value is itself a feature structure which contains the attribute morph whose value is nominative.

Templates are defined using path equations and calls to other templates. For instance, the template represented by the feature structure in (A.2) is encoded in PROLOG as indicated in (A.3). Finally, lexical rules are also used. Again they find a direct implementation in PROLOG whereby a lexical rule of the form $FS1 \Rightarrow FS2$ (where $FS1$ and $FS2$ are variables ranging over UCG(2) signs) is simply implemented as:

```
lex_rule(FS1, FS2) :- ...
```

where ... indicates any sequence of path equations and template references.
Figure A.2: UCG sign for an intransitive verb.

```
template(iv,Sign) :-
    Sign:synt:res:cat <=> sent,
    Sign:synt:active <=> NP,
    NP:synt:cat <=> np,
    NP:synt:feats:case <=> nom,
    NP:order <=> post,
    Sign:synt:res:feats <=> NP:synt:feats,
```

Figure A.3: PROLOG encoding of the template @intransitive.
A.1.3 Compiling the lexicon

To increase efficiency at parsing time, the lexicon is compiled prior to parsing so that the retrieval of lexical information reduces to lexical look-up. This is achieved by partially-executing all templates and lexical entries. Partial execution consists in performing some resolution steps in advance. A resolution step replaces a literal with the body of a matching clause under the unifying substitution of the literal and the head of the clause. Partially executing a clause (which is called a program clause) with respect to a set of predicates (called auxiliary predicates) thus consists in resolving the clause on literals containing these auxiliary predicates and recursively partially executing the resolvent. For instance, suppose we partially execute a program clause of the form:

\[ p(\text{X}) : - q(\text{X}, \text{Y}), r(\text{Y}). \]

and there is a matching clause for \( q(\text{X}, \text{Y}) \) which is:

\[ q(a, b). \]

Then partially executing \( p(\text{X}) : - q(\text{X}, \text{Y}), r(\text{Y}) \) with respect to the auxiliary predicate \( q \) yields:

\[ p(a) : - r(b). \]

where \( q(\text{X}, \text{Y}) \) has been replaced by the body of the matching clause (which here is null) and the unifying substitution of the literal with the head clause replaces \( \text{X} \) and \( \text{Y} \) in the program clause with \( a \) and \( b \) respectively.

In effect, partially executing UCG(2) lexical entries boils down to replace prolog clauses of the form:

\[ \text{word(Word, Sign)} : - \ldots \]

where \( \text{Sign} \) is an uninstantiated variable and \( \ldots \) indicates a sequence of template calls and path equations, by a unit clause of the form:

\[ 1 \text{For a more detailed explanation of partial execution and its use in compilers, see [Pereira/Shieber 1987].} \]
word(Word,Sign).

where Sign has become instantiated to a UCG(2) sign and encapsulates all the information contained in the path equations and template calls which before partial execution constituted the body of the clause. For instance, partially executing the template in Figure A.5 above yields the following unary clause:

template(iv, [_, sent'[_,WB,PERS,]/ [_, np'[_,WB,PERS,nom], X, post], P/I, _]).

Note also that type checking occurs when the lexicon is compiled. In effect, path equations implicitly encode the type of the feature structures they involve. For instance, a path equation with path X:synt:cat is only well-formed if X is a UCG(2) sign because only UCG(2) signs are defined as having the synt attribute. Moreover, the value of the synt attribute must be a feature structure of type synt because only these feature structures are defined on the cat attribute. When path equations are evaluated, the information about the internal structure of UCG(2) attributes contained in the type-declarations is made use of in such a way that if a path equation violates the requirements set on UCG(2) linguistic objects by these type declarations, compilation fails. This ensures that errors in the lexicon are spotted at lexical compilation time rather than at parsing time.

A.1.4 The grammar rules

Like UCG(2) feature structures, the grammar rules of UCG(2) find a natural encoding in PROLOG. Thus the binary rule of functional application described in detail in chapter 2 is encoded as follows (this is the forward version of functional application; the backward version is similar except that the argument precedes the functor and the order attribute of the argument and of the active part of the functor is post rather than pre):

fa_rule(fa,
[Pho,(Synt/[PhoA,SyntA,SemA,pre]),Sem,Order],
[PhoA,SyntA,SemA,pre],
[[Pho,PhoA],Synt,Sem,Order]).
A.1.5 The parser

To suit the lexicalist character of UCG(2), I use a bottom-up shift-reduce parser. This is augmented with a chart to keep a record of the sub-constituents found during the parse and to avoid recomputation of partial results.

A.2 Extending UCG(2)

To integrate into UCG(2) the analysis of gapping and VP ellipsis developed in this thesis, some modifications need to be introduced. First, a number of minor changes must be made to cater with the 'incomplete' syntax characterising ellipsis. These are fairly trivial and include: adding a product rule to the set of UCG(2) rules; adding a category for coordinating conjunctions which takes as arguments a complete sentence on the one hand and a product category on the other; and adding a lexical rule which will transform ordinary auxiliaries into VPE auxiliaries. Given the introduction given above, the way in which these additional features might be incorporated should be fairly obvious and so will not be discussed here.

A second more potent set of changes that need to be made concerns the licensing and the interpreting of elliptical constructs. As discussed in chapter 7, section 7.2.1, I take these to occur at the level of discourse (rather than at the sentential level). Thus, to account for the fact that ellipsis is a discourse phenomena, I augment UCG(2) with a discourse parser which will allow ellipsis resolution to take place whenever two sentences are combined to form a bigger discourse unit (I take clauses of category $S$ to be the basic building block of discourse units).

In what follows, I describe how the discourse parser works and how resolution occurs.

A.2.1 The discourse parser

A caveat is in order: recall (cf. chapter 7, section 7.2.1) that in this thesis, the discourse parser is only designed to isolate clauses and make clear when ellipsis resolution is to take place. It does not (as should any 'real' discourse parser) output any representation of the structure of discourse. Neither does it say anything about discourse relations. This discourse parser is essentially a shift-reduce parser augmented with a chart. In this chart an edge will be of the
form:

discourse_edge([Start, End, Sign, Tree])

where Start and end indicate the starting and the ending vertices of the edge respectively, Sign is the ucg(2) sign associated with this particular edge and Tree is the corresponding derivation tree. The basic edges in the chart cover either a string in the discourse which is recognised by the parser as a sentence (it is then called a sentence edge) or a connective such as and or but (called a connective edge). For instance, given the discourse:

(A.1) Jon likes Mary and Peter does too.

the discourse parser will start by building three basic edges as indicated in Figure A.4. Discourse

![Figure A.4: Discourse chart with basic edges only.](image)

parsing then proceeds by alternatively shifting and reducing edges just as with any ordinary shift-reduce parser. The top call of the discourse parser is thus:

```
d_parse([], [], M) :- !.
d_parse(Discourse, [], M) :-
    s_parse(Discourse, DiscRest, SignList), !,
    New is N+1,
    d_shift(N, New, SignList), !,
    d_reduce(N, New), !,
    d_parse(DiscRest, New, M).
```

where Discourse is a list of words corresponding to the discourse to be analysed. The role of s_parse(Discourse, DiscRest, SignList) is to take this list of words and split it into two sub-lists Wordlist and DiscRest such that WordList corresponds either to a sentence or to
a connective, i.e., to a basic discourse edge and DiscRest is the list of words remaining to be parsed. SignList contains the set of signs associated with the basic discourse edge. d_shift then ensures that as many discourse edges are shifted on the discourse chart as they are signs in SignList. These edges are then reduced by d_reduc. Finally a recursive call to d_parse ensures that the rest of the discourse (DiscRest) is also parsed. Parsing stops when the list of words to be parsed is empty.

Given this algorithm, the discourse in (A.1) will yield the discourse chart in Figure A.5.

![Discourse chart](image)

Figure A.5: Discourse chart.

### A.2.2 The resolver

Ellipsis resolution occurs when two discourse edges are reduced and one of them contains an elliptical clause. It is at this stage in parsing that the licensing and interpreting of both VPE and gapping takes place. More generally, whenever two discourse edges are tried for reduction, two cases can be distinguished: either one of the edges to be reduced contains an ellipsis and then the ellipsis must be resolved\(^2\) or there is no ellipsis and the two edges may combine without any need for resolution. This is encoded in PROLOG as follows.

\[\text{resolve}(\text{Sign1}, \text{Tree1}, \text{Sign2}, \text{Tree2}, \text{Sign}, \text{NewSign}) :-\]

\(^2\)Here, I make the simplifying assumption that the a-clause is that covered by the other edge to be reduced. That is, any case of VPE where the a-clause does not immediately follow or precede the e-clause is not dealt with. The reason for this is that as discussed in chapter 5, the a-clause of a VPE clause can only be identified using knowledge about the structure of discourse and that as already mentioned such knowledge is not available to the present parser.
APPENDIX A. IMPLEMENTATION

identity_anaphor(Sign1, Sign2, AnaphorType, AnaphorDirection), !,
order_trees(Tree1, Tree2, AnaphorDirection, TreeList),
resolve(AnaphorType, AnaphorDirection, TreeList, Sign, NewSign).

resolve(Sign1, Tree1, Sign2, Tree2, Sign, Sign).

where Sign1, Tree1, Sign2 and Tree2 are the signs and trees of the two edges to be reduced and Sign is the result of combining Sign1 and Sign2 using some UCG(2) rule. When no ellipsis is present, the second clause of \texttt{resolve/6} applies and the sign of the resulting edge is just Sign. In contrast, if one of the two edges to be reduced contains an ellipsis, the predicate identity_anaphor will succeed and resolution will be performed. More specifically, given two signs Sign1 and Sign2, identity_anaphor will return some information about the type of the anaphor (gapping or VPE) and the direction of the anaphor (backward or forward). order_trees then simply order the trees of the two input edges in a list (TreeList) where the first tree is the tree of the a-clause and the second, that of the e-clause (so if a case of backward VPE is analysed the order of the trees in TreeList contradicts the surface ordering of the corresponding clauses). Finally, a call to resolve/5 ensures that the ellipsis in the e-clause is resolved and that the sign NewSign associated with the resulting edge contains a resolved semantics (rather than an unresolved one).

Predictably, resolve/5 consists of two clauses, one of which deals with VPE resolution and the other with gapping. Consider first, the clause for VP ellipsis.

\texttt{resolve(vpe, AnaphorDirection, [ATree,ETree], MotherSign, NewSign) :- }
\begin{verbatim}
  build_delta(ETree, ATree),
  mother_node(ETree, ESign),
  ESign:sem <=> ESem,
  ESign:disgap <=> VP,
  node(ATree,VP),
  VP:sem <=> ASem,
  get_esem(ASem,ESem,ESem1), !,
  r(ESem1, ESem2),
  get_mother_sem(AnaphorDirection, MotherSign, ESem2, NewSign).
\end{verbatim}

The first clause build_delta builds the function \(D\) needed to ensure the resolution of sloppy pronouns (cf. chapter 4) by recursively walking through the two trees on which \(D\) is defined.
APPENDIX A. IMPLEMENTATION

222

(that is, the tree of the a-clause and that of the e-clause). mother_node(Tree,Sign) returns the category (here, ESign) associated with the mother node of the tree Tree (in this case the tree of the e-clause). ESign:sem and ESign:disgap return the value of the semantic and of the discourse-gap attributes respectively, for the sign (ESign) associated with the elliptical clause. node(Atree,V) searches the tree of the antecedent clause for a VP (VP) which unifies with the value of the discourse-gap of the e-clause. This VP provides the semantics (ASem) needed to resolve the VP ellipsis. get_esem(ASea,ESe",ESe»l) then ensures that this semantics is unified with the place holder provided by the unknown predicate occuring in the semantics of the VPE clause. r(ESem1, ESem2) then computes the final semantic representation of the e-clause by applying the function R to ESem1 to yield ESem2. Finally, get_mother_sem(AnaphorDirection, MotherSign, ESem2, NewSign) ensures that the semantics of the sign (MotherSign) resulting from the combination of the two edges contains the resolved semantics (ESem2) of the e-clause.

Resolution for gapping is very similar except that the substitution operation has to take place. This is ensured by the call to substitute/3 which takes the a-tree and the e-tree to yield a new tree ETreel which both licenses the gapped clause and provides the basis for its interpretation. The resolution procedure for gapping is as follows:

\[
\text{resolve(gapping, AnaphorDirection, [ATree,ETree], MotherSign, NewSign) :-
build_delta(ETree, ATree),
substitute(ETree,ETree,ETreel),
mother_node(ETreel,ESign1),
ESign1:sem <=: ESem1, !,
r(ESem1, ESem2),
get_mother_sem(AnaphorDirection, MotherSign, ESem2, NewSign).}
\]

In general then, there is a simple PROLOG implementation of the ideas developed in the body of this thesis. The relation between theory and implementation can be summarised as follows. Consider syntax first. In the case of VPE, the analysis was that a VPE auxiliary has a value for its disgap attribute which indicates what kind of UCG(2) object the antecedent must be. In particular, it places some requirements on the syntactic category (VP, VP/NP or X[+PRD]) of the antecedent, on its voice and on its aspectual features. This is implemented in the resolver by the two clauses ESign:disgap <=: VP and node(Atree,VP) which in effect require that the value of the disgap attribute in the e-clause unifies with that the category of some node in the a-tree. In the case of gapping, the main syntactic constraint is that the substitution of
the gapping remnants into the tree of the a-clause must yield a grammatical sentence. This is implemented in the call to the substitute predicate whose definition in prolog closely reflects the definition of σ given in chapter 4.

From the semantic point of view, the theoretical analysis supports two main ideas: first, that the interpretation of elliptical constructs is a function \( R \) of the meaning of the antecedent, and second, that sloppy pronouns are pronouns which are constrained by strong parallelism requirements. These two ideas find a simple implementation in the resolver just described. Parallelism is modelled by the call to the procedure build_delta which establishes which NP in the a-clause is parallel with which NP in the a-clause. The fact that the interpretation of elliptical constructs is a function of the meaning of the antecedent is encoded in the clause \( r(ESem_1, ESem_2) \) which can be thought of as meaning \( R(ESem_1) = ESem_2 \).

A.3 Possible improvements

The implementation described above is mainly intended to illustrate how the ideas developed in this thesis can actually be implemented. It does not purport to give an exhaustive treatment of the linguistic and extra-linguistic constraints regulating ellipsis. As such, it presents many shortcomings and could be improved in several ways and at different levels. As discussed in chapter 2, several extensions are possible within a unification-based framework which increase expressivity and allow for linguistic generalisations to be better captured. These include, in particular, the possibility of expressing negation and disjunction of features, the use of set and list values, and the introduction of relational dependencies. In the present implementation, none of these extensions are available. However, the general setup of the lexicon (the use of templates and path equations and the compiling of the lexicon) is such that integrating these extensions is possible without major changes to the global architecture of the system. Essentially, all that needs to be done is first, to define how these extensions are to be treated by the lexicon compiler; and second, to decide on the semantics of these extensions and their implementation in prolog.

The first task (defining how these extensions are to be treated by the lexicon compiler) is relatively easy and could be carried out by having the lexicon compiler replace any value which does not lend itself to straightforward unification by a variable, say \( X \), in the lexical sign which is being compiled. This variable is then associated with the corresponding value and added to the sign in a list of constraints which is checked upon at parsing time. So, for instance, the
lexical entry in (A.6) below will yield after compilation a PROLOG term of the form given in (A.7).

\[
\text{template}(iv, \text{Sign}) ::=
\]

\[
\text{Sign:synt:res:cat } \leftrightarrow \text{ sent}, \\
\text{Sign:synt:active } \leftrightarrow \text{ NP}, \\
\text{NP:synt:cat } \leftrightarrow \text{ (np or pp)}, \\
\text{NP:order } \leftrightarrow \text{ post}, \\
\text{Sign:synt:res:feats } \leftrightarrow \text{ NP:synt:feats}, \\
\text{Sign:sem:arglist } \leftrightarrow \text{ NP:sem}.
\]

Figure A.6: Example of disjunctive feature value.

\[
\text{template}(iv, [\_, s\text{-FEATS}/[\_, \text{Cat}\text{-FEATS}, X, \text{post}], \text{P/I}, \_]: \\
[\text{constraint}(X, \text{np or pp})])
\]

Figure A.7: Compilation of disjunctive feature value.

For the second task (deciding on the semantics of the various possible extensions and on their PROLOG implementation), a good summary of how set and list values, relational dependencies, disjunction and negation can be handled is presented in [Cooper 1990].
Bibliography


Asher, N. [1990]. Abstract objects, semantics and anaphora. Ms, Centre for Cognitive Science, The University of Texas, Austin.


Chierchia, G. and Rooth, M. [1984] Configurational notions in discourse representation theory, NELS 14, Graduate Linguistic Student Association, University of Massachusetts, Amherst, 49-63.


Daskal, M. and Katriel, T. [1977] Between semantics and pragmatics: the two types of But


Feynman, R. [1987] Surely, you are joking Mr. Feynman.


BIBLIOGRAPHY


Kasper, W., Moens, M. and Zeevat, H. [1990] Anaphora resolution in ACORD.


Lambek, J. [1958] The mathematics of sentence structure. American Mathematical Monthly,


BIBLIOGRAPHY


Steedman, M. [1989] Gapping as constituent coordination. MS-CIS-89-76, University of
BIBLIOGRAPHY


