Processing Dependencies

Martin John Pickering

Ph.D.

University of Edinburgh
1991

To Edith Goff

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.

Edinburgh, March 22, 1991

Martin Pickering

Acknowledgements

This thesis has only proved possible because of the enormous amount of help, encouragement and criticism provided by various people. Parts of it represent joint work, as shall now be made clear.

Many of the linguistic and formal analyses in chapters 2, 3, 4 and 5 of the thesis have emerged from joint work conducted with Guy Barry during the last two years. His influence is particularly felt in the use that is made of the categorial grammar framework. Sections which are derived directly from joint work are indicated in the text. Chapter 6 describes an experiment conducted with Janet Nicol.

I would like to thank Elisabet Engdahl and Richard Shillcock for giving me invaluable supervision and spending so much of their time on my work. Richard gave particular help with the experiment described in chapter 7. Without their assistance, I feel that it is very unlikely that I would have been able to submit this thesis at this time. I would also like to thank Terry Myers for his assistance. Even in the parts of the thesis that do not reflect joint work, Guy Barry's encouragement and criticism has been very considerable. Rosemary Stevenson directed me toward cognitive science before I began this thesis, but was also a great influence during the thesis, and together with David Kleinman, assisted me in practical aspects of the experiment in chapter 7. Ken Forster made it possible for me to visit the University of Arizona, and he and Janet Nicol helped me develop the psychological issues in my thesis. Mark Steedman encouraged much of my work in my first year in Edinburgh, and helped direct me into the general area that this thesis explores. Glyn Morrill and Mark Hepple have considerably influenced my views on categorial grammar, and Matt Crocker my views on psycholinguistics. I have also greatly benefited from conversations with and comments from Andy Barss, Kathrin Cooper, Robin Cooper, Merrill Garrett, Janne Johannessen and Antonio Sanfilippo, and many others. Finally, I would like to thank Ted Briscoe and Henry Thompson for commenting on and assessing this thesis.

I have needed lots of non-academic assistance as well. My parents, Elizabeth and Elizabeth's parents have given me the kind of support that has made it possible for me to write this thesis. I cannot thank them enough for their kindness. I owe a great deal to my family, friends and, very importantly, colleagues in the Centre for Cognitive

Science, who have made Edinburgh such an enjoyable and stimulating place to live and work.

Abstract

The thesis assumes that a theory of language should be evaluated with reference to both linguistic and psycholinguistic evidence, and that there should be a very straightforward relation between grammar and method of processing. It contends that most current linguistic theories are limited by their lack of interest in processing considerations. It concentrates on one major issue, that these theories do not have any straightforward way of representing the fact that we can begin an interpretation of an utterance before the utterance is completed. It is argued that a cause of this is the assumption that sentences should be represented by phrase structure grammar, which only allows rigid, non-overlapping constituents. It suggests that this should be replaced by a flexible notion of constituency, called dependency constituency, and based on a generalized notion of dependency.

The thesis then shows how this notion can be represented in the grammatical framework known as the Lambek Calculus, by means of a very simple restriction on permitted operations. It then shows how this allows a new characterization of strings that can be coordinated, which, it is argued, captures the relevant data better than competing theories. The thesis then proposes an account of human sentence processing based on dependency constituency. It applies the account of dependency constituency to unbounded dependencies, and suggests that this treatment is more parsimonious than alternatives based on phrase structure grammar that make use of mechanisms such as empty categories. It shows how experimental evidence that is usually interpreted in terms of theories using empty categories can be reinterpreted in terms that do not use these devices. But the next chapter goes further, and shows how it is possible to distinguish between a processing theory with empty categories and a theory without empty categories. It is claimed that the evidence supports a theory without these devices, and so a grammatical theory that bears any relation to sentence processing should avoid their postulation.

This theory is then employed as part of a psycholinguistic model concerned especially with the resolution of ambiguities. It contends that in a theory without empty categories, it is possible to collapse the processing of unbounded dependencies with the processing of other constructions in a way that is obviously parsimonious. It is then shown that the experimental evidence supports this account, and indicates how this can be interpreted in terms of dependency constituency. The conclusions are that a psychologically plausible grammar should be formulated in terms of dependency and flexible constituency, and that empty categories should be avoided.

Contents

7 Processing Subject Extractions 174

Chapter 1

Introduction

1.1 Goals and Assumptions

There is a common but often implicit assumption that standard views about linguistics can serve as an adequate foundation for a theory of the human processing of language. This indicates that there is no need for linguistics and psycholinguistics to develop in isolation, which is surely advantageous. However, in practice there is little attempt by linguists to consider whether their theories are psychologically plausible. This means that psycholinguists either have to define complex and unintuitive mappings from the linguistic framework, usually associated with grammatical competence, or else have to abandon any attempt to associate their theories with linguistic models.

The problematic relation between grammatical models themselves and the consistency of these models with language use can be found in Chomsky (1965), who assumes that a grammar should describe knowledge of language, and therefore ultimately must correspond to some form of mental representation or structure. It 'provides the basis for actual use of language by a speaker-hearer', but 'does not, in itself, prescribe the character or functioning of a perceptual model or a model of speech production'. As Bresnan (1978) points out, Chomsky holds back from assuming a direct relation between linguistic rules and operations of the processor. Chomsky is of course right that no grammatical model is a model of processing, but this is true in the trivial way that a logic cannot be computed until some method of deduction is added. Bresnan's point is that the grammar should be formulated in such a way that it is possible to describe processing which makes use of the rules of the grammar directly, in the same way that a method of deduction will use the rules of the logic. The design of the processor indicates which rules to apply and when to apply them. Although it is possible that it will not apply a given rule, it will never make use of a procedure that can be described by a rule that it does not have. If it did, there would then be a second, 'processing', grammar, distinct from the competence grammar. This simply involves the multiplying of unobservables beyond necessity. Likewise, it will not have grammatical rules that it never applies.

This thesis therefore assumes what we shall call the 'Strong Competence Hypothesis', that an adequate grammar both generates all and only the sentences of a language and also allows direct implementation of this grammar in a way consistent with performance evidence. This concept is admittedly rather vague, but is useful as part of a general theoretical orientation. It is closely related to the notion of 'transparency', as employed by Berwick and Weinberg (1984). The appropriate working methodology is to assume that the parsing apparatus is not unnecessarily complex, and therefore that there can be some relation between performance data and the appropriate form of the linguistic theory. At the same time it must be acknowledged that experimental evidence often admits multiple interpretations, and so it would be foolish to assume that such data can always straightforwardly distinguish competing linguistic accounts.

This thesis claims that there is some very important performance evidence which bears directly on the form of a linguistic account and whose implications cannot seriously be questioned. It concerns what will be termed 'incremental interpretation', which means no more than that we understand each part of an utterance as we hear it. We shall use it to refer to systems that allow us to interpret units which are not constituents generated by standard phrase structure grammar. This thesis makes the parsimonious assumption that a theory consistent with the Strong Competence Hypothesis will only allow constituents to be interpreted. In other words, the notion of a constituent will be defined uniformly across the syntactic and semantic domains. This assumption has the effect of allowing evidence from the study of incremental interpretation to be combined with standard linguistic evidence in the evaluation of grammatical theories and associated processing accounts.

The thesis argues against some common assumptions of modern grammar. Language is standardly analysed into 'rigid' constituents, which permit no overlap, so that if XYZ is a string, then it is impossible for both XY and YZ to be constituents. But rigid constituents do not in general correspond to the units that are incrementally interpreted. Rather than allowing the interpretation of some non-constituents, the thesis proposes to replace rigid constituency with a flexible notion derived from dependency relations, where two words form a constituent if there is a dependency relation between those two words. This restriction allows overlapping constituents, in contrast to the standard assumption, and we can now represent incremental processing as the formation and associated interpretation of a particular subset of these constituents.

The thesis introduces a formal means to represent this notion of constituency, based on flexible categorial grammar. It therefore in a sense operates on two levels, firstly representing dependency structure, and secondly describing a set of mechanisms that generates these structures. One effect of this is to make flexible categorial grammar motivated by general linguistic concerns, rather than the relatively small set of constructions that are usually considered.

An important implication of this approach is that no difference is assumed between 'canonical' constructions, which are generated directly by the 'base' grammar, and 'non-canonical' constructions, which are generated derivatively. For instance, in transformational grammar, actives are canonical but passives are non-canonical, and are derived from a form related to the active. This dichotomy is especially interesting in the area of unbounded dependencies, where many theories assume that an element has been moved, leaving an empty category, trace or gap behind. In most accounts of processing, unbounded dependencies involve some kind of reversing of this movement operation. This begs the question of why we would want to have non-canonical constructions in the first place. The thesis shows that such a process of reversal is very unparsimonious and in fact empirically falsifiable. If it is not used in processing then the parsimonious assumption is that it should not be used in the grammar.

Linguistics should have a direct role to play in models of sentence processing, and data from both areas should be equally relevant in the description of language. It is only by relating the grammatical theory to our mental faculties that we can hope to achieve explanatory adequacy. If there are mismatches between linguistic and performance theory then one or both accounts must be ill-formulated, or at least not described in a way that is most useful to the development of an integrated cognitive science. This thesis makes the basic assumption that a grammatical theory should display a very close relationship with principles of processing and explores the consequences of this assumption for both linguistics and psychology.

1.2 Outline of Thesis

Chapter 2 discusses conceptions of dependency and constituency and relates them to the way in which we interpret sentences. It assumes that sentence processing is essentially incremental, so that we begin to understand an utterance as soon as we begin to hear it, and that we do not have to wait until the end of a sentence or clause. It assumes that any realistic grammar must be capable of being used as the linguistic component of a model of sentence processing, and claims that traditional, rigid, notions of constituency are undesireable because they do not in general allow incrementally interpretable units to correspond to constituents.

It claims that we should employ a new, flexible notion of constituency, based on the general principle that two words are part of the same constituent if there is a dependency between them. These constituents are called dependency constituents, and, unlike phrase structure constituents, can overlap, as discussed above. It is claimed that there is no uncontroversial linguistic or psycholinguistic evidence for standard phrase structure constituency, such as for the existence of a VP which includes the verb and object arguments but not the subject. It then argues that sentence processing involves the incremental construction of dependency constituents, where new words are integrated into the present structure if they form a single dependency constituent. This is shown to be compatible with processing using phrase structure grammar, or with a traditional means of representing dependency grammar, only if it is assumed that it is possible to interpret units which are not constituents. It is claimed that it is more parsimonious to use a grammar that represents dependency constituency directly.

Chapter 3 argues that such an account is possible within flexible categorial grammar. It first introduces the simplest version of categorial grammar, known as the Ajdukiewicz-Bar-Hillel or **AB** system (Ajdukiewicz 1935; Bar-Hillel 1953). **AB** allows the representation of dependency relations, but does not permit a flexible and incrementally interpretable notion of constituency. Therefore this is replaced with another system, the Lambek Calculus or **L** (Lambek 1958). In itself this system is too flexible to represent dependency constituency, because it allows the combination of any adjacent words, but it is possible to define dependency constituency within **L**. This is done by introducing a natural deduction axiomatization of **L**, and by imposing a simple restriction on permissible rules. One important change is made from standard categorial grammar, in that modifiers are treated as having the lexical characteristics of arguments rather than functors. A system is now defined which allows the interpretation of all strings which are dependency constituents, but none which are not.

This is then used to form the basis of a new means to characterize which strings can be coordinated. The initial claim is that only dependency constituents of the same type can be coordinated. This does not allow all strings which can be given the same type to coordinate, because some strings can only be given the same type by non-dependency-preserving operations. However, this is too restrictive, and so it is finally claimed that coordination is possible so long as each conjunct is made up of dependency constituents of the same type in the same order. The chapter concludes with a discussion of incremental interpretation, and claims that the coordination evidence suggests that the processor forms all dependency constituents and nothing else. It gives an explanation of why nested constructions are hard to process in terms of dependency constituents, and argues for a necessary processing restriction on the form of any grammar which is consistent with the assumed version of the Strong Competence Hypothesis.

Chapter 4 considers the representation and processing of unbounded dependencies. It discusses the analyses that have traditionally been made within constituency-based theories, arguing that context-free grammar restricted to traditional category labels has to stipulate some extra mechanism to deal with unbounded dependencies. Transformational grammar (Chomsky 1965; 1981) assumes additional rules, levels of representation and often empty categories, whereas Generalized Phrase Structure Grammar (Gazdar, Klein, Pullum and Sag 1985) makes use of special derived categories, but both approaches can be seen as having treatments of unbounded dependencies which regard them as non-canonical constructions. It then argues that there is no simple representation of unbounded dependencies within the dependency representation introduced in chapter 2.

In contrast it is possible to represent unbounded dependencies in a simple way in flexible categorial grammar, and more interestingly, to capture the notion of dependency constituency in relation to unbounded dependencies. The chapter then discusses the differences between this formulation of categorial grammar and Steedman's Combinatory Categorial Grammar or CCG (Steedman 1987). CCG is similar to the dependency-preserving subset of **L** in many respects, but includes some non-dependency preserving operations and disallows some dependency preserving ones. Steedman attempts to capture permissible extractions and permissible coordinations simultaneously. This is shown not to be compatible with all the data, and the chapter claims that the data supports the analyses given in chapter 3 rather than Steedman's.

The chapter then considers the incremental processing of unbounded dependencies. It claims that the use of empty categories is simply unnecessary if our grammar allows incremental interpretation, and that if the GPSG treatment of unbounded dependencies is used, then there is data which forces us to use a much more complex apparatus than is generally assumed. But if this apparatus is added, there is no reason not to use it always, in which case we have a system that allows incremental interpretation and can be made very like flexible categorial grammar. Finally, it considers the experimental literature on processing unbounded dependencies, which has generally assumed the existence of empty categories and some 'gap-filling' mechanism. It shows that all the evidence can be reinterpreted without requiring empty categories.

Chapter 5 goes a stage further, and claims that the assumption that there are empty categories which are reflected in sentence processing is untenable. Unbounded dependencies are not processed by gap-filling, but instead an association is formed directly between the extracted element and its governor, which is the head of the purported empty category. In many cases, the head and the empty category are adjacent, so it is very difficult to see how the two accounts can be differentiated. But this is not always the case. If a verb subcategorizes for two post-verbal phrases, for instance, and the second is extracted, then the purported empty category is not adjacent to its head. The chapter uses the methodology of recursing this process of extraction in a way that requires a nested pattern of associations in an account with empty categories, but a disjoint pattern in an account without. The sentences do not become difficult to process however often the construction is recursed, a fact which is shown to be incompatible with the nesting account.

The chapter then considers possible escape routes for a theory with empty categories, based on the assumption that the empty category does exist but is in an unusual location. These are shown to be flawed for various reasons, and so the only conclusion is that the processing of unbounded dependencies takes place without the postulation of empty categories. Chapter 4 argued that it is more parsimonious for a grammatical theory to avoid empty categories in unbounded dependencies, but this chapter shows that their use is in fact inconsistent with the assumed version of the Strong Competence Hypothesis. Finally, the chapter shows how this this data can be modelled by using dependency constituency and within the dependency-preserving subset of **L**.

Chapter 6 considers the processing of unbounded dependencies from a more psycholinguistic perspective, considering the way in which we resolve ambiguities and the use of experimental methods. The thesis so far has made the simplifying assumption that all strings are unambiguous, but of course this is not true. Therefore the chapter considers the processing of various kinds of ambiguity, and shows how they can be represented in the categorial notation in a way that indicates when we make the wrong analysis. It then discusses how we decide which analysis to pursue, and suggests that the crucial difference is between cases where, at a particular point in a sentence, both analyses result in dependency constituents, and cases where one analysis does but the other does not. The main hypothesis is that in the latter case we always (except perhaps when there is contrary prosodic information) choose the analysis where a single dependency constituent is formed initially, but that this decision may be reversed quickly if its inappropriateness becomes apparent.

This model is then applied to the treatment of unbounded dependencies, where it is claimed that exactly the same principle applies. This is natural in the present account, which does not assume the existence of additional linguistic or cognitive mechanisms to deal with this construction. Gap-filling accounts, on the other hand, are unconstrained by the processing methods for 'canonical' constructions, and hence it is impossible to make any clear predictions about what they must be like, but the most natural accounts assume that mislocation of gaps will be costly. In fact the experimental evidence supports the hypothesis that dependency constituents are formed immediately they can be, even though this results in many analyses that have to be revised. Hence the present account, which gives a clearly falsifiable hypothesis, is in fact supported by the experimental evidence. The chapter then considers some new experimental evidence which supports the nesting argument presented in chapter 5.

The chapter then considers the processing of constructions involving another empty category proposed within recent transformational grammar, known as NP-trace. It constructs an argument against its psychological reality, similar to that used in chapter 5 with unbounded dependencies. This indicates that experimental evidence for the existence of this empty category needs to be reinterpreted. Finally, the chapter considers the methodology of reactivation in more detail, and argues that while it is clearly sensitive to the initial processes of integration and the formation of dependency constituents, it may also be sensitive to higher-level processes. It is suggested that other predictions of the priming methodology should be tested.

Chapter 7 is concerned with subject extractions, specifically subject relatives, and again attempts to associate processing evidence with linguistic theorizing. Descriptive evidence suggests that extractions from embedded subject position (as in *who Sue thinks loves Mary*) are marked or unusual, and it is also sometimes held that simple subject relatives (like *who loves Mary*) do not involve extraction at all. If either of these is correct, then subject relatives should be treated in a way that differentiates them from other extractions. The descriptive evidence is argued to be inconclusive, but it is claimed that the accounts of standard GB and GPSG are only tenable if subject relatives are differentiated from other relatives. The chapter then assesses flexible categorial grammar treatments of embedded subject relatives, and acknowledges that this construction creates problems for the frameworks. Steedman's treatment is compared with Moortgat's (1988a), and it is argued that Moortgat's allows the representation of dependency and incremental interpretation in ways that are forbidden by Steedman. It also does not assume that embedded subject extractions are marked or unusual, at least from the point of view of processing.

An experiment is then described which tests whether subject extractions are marked, and specifically whether the relationship of embedded to simple subject extractions is the same as the relationship of embedded to simple object extractions. It concludes that the relationship is the same, and that there is nothing especially marked about embedded subject relatives. This suggests that a grammar abiding by the Strong Competence

Hypothesis should avoid idiosyncratic treatments of subject relatives. The experimental data is then modelled within dependency constituency, using Moortgat's extension of the Lambek Calculus. This serves as an example of how linguistic theory and sentence processing can be used together without the need to make additional stipulations.

Chapter 2

Dependency, Constituency and Incremental Interpretation

2.1 Introduction

The relationship between generative linguistic theory and sentence processing is faced with the problem that the linguistic structures which are usually proposed are simply not amenable to efficient processing. For example, standard phrase structure tree representations for English sentences are predominantly right-branching, so that (as we shall see) the beginning of a sentence often does not form a constituent, and therefore we have to wait until much more of the sentence has been considered before constituents can be formed.

The term 'incremental interpretation' has been used to refer to the idea that it is possible to obtain a partial understanding for an utterance before that utterance is complete. Let us restrict it to cases where interpretations are given to strings which do not correspond to completed phrase structure constituents. Altmann and Steedman (1988; Steedman 1989) argue that the psychological reality of 'incremental interpretation' cannot seriously be disputed in the light of overwhelming evidence. We shall look at the evidence for this below, and suggest that there is little doubt that they are correct. For instance, in a study of speech comprehension, Marslen-Wilson (1973) showed that the interpretation of one constituent can influence the interpretation of another constituent before the larger constituent which they are both part of has been completed. Altmann and Steedman claim that we should abandon standard theories of grammar, and, instead, advocate an approach that allows all languages to be treated as, in some sense, 'left-branching'. This allows a parsimonious relation between grammar and processor, in the sense that all the strings that get an interpretation during comprehension are in fact syntactic constituents.

The problem is that most standard linguistics assumes that a large proportion of constructions do not have left-branching analyses. Indeed few theories will allow these analyses; the most striking exception is 'flexible categorial grammar', of which Steedman himself is a main proponent (eg 1987; Ades and Steedman 1982). But a belief that Steedman's assumptions must be linguistically flawed is found even in research trying to construct incremental models of parsing:

> A grammatical theory, whatever its connection with semantic interpretation, should assign constituent structure in a way consistent with distributional evidence, intonational contours, and so on. It is by no means clear that categorial grammar is capable of doing this satisfactorily, and it is for this reason that linguists have not usually been attracted to this framework. (Pulman 1987)

Some of the pieces of 'distributional evidence', for instance, that have been used to support traditional phrase structure rules, may be subject to serious criticism, as we shall see, and I shall claim that reformulating a few phrase structure rules is not the best way to address these issues. Instead, in this chapter, I shall claim that sentence processing is based not on phrase structure constituency, but on dependency relations. From this perspective, Altmann and Steedman's use of the term 'left-branching' is misleading, because incremental interpretation is best not based on a phrase structure grammar at all, but rather something related to a dependency grammar, where I shall claim that 'flexible' constituent structure is the result of the process of associating chains of dependencies.

Dependency grammars are linguistically motivated (see Matthews (1981) for instance), and have a very long history. They have encountered little direct criticism, but have been overshadowed as a result of the predominance of American Structuralist and Chomskyan theory within modern linguistics, which is based on phrase structure grammar (Bloomfield 1933; Chomsky 1957; 1965). However, Chomsky suggested that context-free phrase structure grammar was inadequate for the analysis of natural language, and hence he added a transformational component to the phrase structure base. But Gaifman (1965) showed that classical dependency grammar was weakly equivalent to context-free phrase structure grammar, and since no other version of dependency grammar was considered at that time, it was assumed that dependency grammar was inadequate to deal with natural language. It is now known that context-free phrase structure grammar is very nearly adequate to describe natural language, in contrast to the assumptions of Chomsky (Gazdar and Pullum 1982; Sheiber 1985; Culy 1985). Therefore the case against classical dependency grammar is much weakened.

A consequence of this discovery of the power of CFPSG is the appearance of a number of generative linguistic theories that make no use of transformations, of which the most influential has certainly been Generalized Phrase Structure Grammar (Gazdar, Klein, Pullum and Sag 1985), but there are still few generative grammars that have no basis in phrase structure rules at all. A clear exception is Word Grammar (Hudson 1984), which is an explicit dependency grammar. But a number of modern linguistic theories are closely related to dependency analyses, eg LFG, where constituent structure is becoming increasingly marginalized (Kaplan and Zaenen 1988), and HPSG (Pollard and Sag 1987), as well as categorial grammar itself (Ajdukiewicz 1935). Certainly these theories view syntax more as a means to achieve an interpretation and less as an end in itself. This is partly due to the assumption of the rule-to-rule hypothesis (Bach 1977; Montague 1973) where each syntactic rule has a semantic counterpart, and so there is no need to base an interpretation on a particular tree configuration. The construction of traditional constituent structure is a by-product at best, or else irrelevant. This is precisely the traditional claim or assumption of dependency grammars. Likewise, language comprehension can be seen as having the single goal of converting sound (or text) into meaning (and language generation of doing the reverse). There is no necessary reason to assume that we must do this by constructing a representation of constituent structure first.

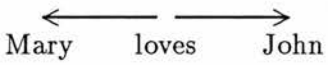
This chapter will outline a description of language, based on dependency relations. This allows the derivation of a flexible notion of constituency, where, unlike in phrase structure grammar, a string XYZ can have both XY and YZ as constituents. We can then make the strong claim that all (incrementally) interpretable units are constituents, which makes the relation between grammar and processor extremely simple, and therefore allows us to develop analyses of language that are constrained by both linguistic and processing evidence.

2.2 Dependency and Constituency

2.2.1 A Perspective on Dependency Grammar

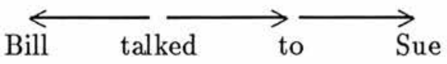
It is possible to make a simple model of grammar based on a generalized notion of dependency between words. This thesis will make no reference to units smaller than words, such as morphemes. The dependency relation is assumed to be directional, so that one of the words, the controlled or *dependent*, is regarded as dependent on the other, the controller or *head*. The primitive reason for assuming a dependency is in cases of subcategorization, with the subcategorizer being the head and the subcategorized element the dependent. We can display dependency relations by using arrows leading from heads to dependents. For instance, let us represent *Mary loves John* as below, with arrows from *loves* to both *Mary* and *John*:

(2.1)



Only one word (standardly a verb in the main clause) may not itself have a head. That word, in this case *loves*, is called the *root of the sentence*. In more complex sentences it is possible to have words which are simultaneously heads and dependents. This is true of the preposition *to* in *Bill talked to Sue*:

(2.2)



We now have what we can call a chain of subordination, where *Sue* is a dependent of *to*, which is itself a dependent of *talked*. Every word in a sentence will be connected to the root via such a chain of subordination.

Dependency grammar is of course a framework as general as phrase structure grammar. In particular there is no unanimity about the dependencies that should be represented for any given sentence. We shall assume that dependency reflects covariation or cooccurence restrictions, so that changing the form of one word in a dependency relation may have an effect on the form of the other. This includes such traditional concepts as government, where the verb is regarded as determining the case of its arguments, and agreement (or concordance) where, for instance, the subject and verb are given equivalent markings for, say, person and number. Dependency is also related to collocational restrictions (see Matthews (1981) for extensive discussion). For instance, the fact that we must (idiomatically) say *grill meat* but *toast bread*, when there is no difference between the meanings of *grill* and *toast*, indicates that there is a dependency relation between each of these verbs and its object. However, dependency grammars do not necessarily assume a dependency between all pairs of words which display cooccurence restrictions. As stated above, the primitive reason for assuming a dependency is if one word subcategorizes for another, with the subcategorizer being the head. Other pairs of words related by cooccurence may not always warrant the assumption of a dependency, as we shall see.

We have to be clear which word is the head and which the dependent. We have assumed that the verb is the head of its arguments, and shall make the general assumption that the subcategorizer (in the traditional sense) is always the head. This of course presupposes that there is a meaningful and unified definition of *head*. This is the conclusion of Hudson's (1987) reply to Zwicky (1985), and we shall follow Hudson in the claim that such a definition is possible.

Notice that no such relations hold between subject and object, so no dependency is assumed. This characterization will hold for any dependency grammar. But consider a sentence like *Bill is happy*. Because *is* is the verb, we shall treat *is* as the head and *Bill* and *happy* as its dependents. For instance *Bill* and *is* agree by both being singular. These relations are uncontroversial. However many (eg Hudson 1984) would claim that there is also a dependency relation between *Bill* and *happy*, with *Bill* being dependent on *happy* as well as *is*. There are certainly collocational restrictions, and, for instance in French, say, there is agreement between the subject and the predicative adjective. This is covariation, but must it therefore require a dependency relation?

The present approach will not assume such a dependency relation. Instead, the dependency structure of *Bill is happy* is taken to be like that of *Mary loves John*. Both *is* and *loves* subcategorize for two arguments, so both the sentences must at least have similar dependencies between the verb and its arguments. The fundamental reason for not assuming an additional dependency in *Bill is happy* is that each dependency is considered as indicating the place at which a compositional semantic operation can occur. We shall assume that the work of making a semantic association between the meanings of *Bill* and *happy* is done by the lexical semantics for *is*. Hence there is no direct association between the meanings of *Bill* and *happy*, and therefore there is not a dependency relation between these two words. In order to give a single meaning to n words we need precisely $n - 1$ compositional operations. This approach keeps absolutely distinct the compositional semantic operations, interpreted as dependencies, and the semantic representation for the sentence. For instance, we may believe that *Bill is happy* should be interpreted as the meaning of *happy* applied to the meaning of *Bill*, so that there is a direct relation between the meanings of the words, but this does not entail that there is a dependency between these words. This putative direct relation between the meanings is a result of the dependency between both of these words and *is*. Obviously, a dependency between *Bill* and *happy* would give us three dependencies, which would not be permitted between three words.

We shall also assume that an auxiliary verb, not the 'main' verb, is the root of the sentence. Miller (1985) makes this point when criticising the discussion of auxiliaries in Matthews (1981). There is covariation between auxiliary and subject, for instance (note that the syntactic properties of *have* do not change when it is used as a perfective, as in *John has eaten*, rather than as a 'main' verb, as in *John has problems*). We shall therefore assume that the auxiliary *have* works in a similar way to the auxiliary *is* above, and associates the subject and the 'main' verb in a particular way. It is irrelevant that a semantic representation might regard the 'main' verb as central (in the same way that *happy* rather than *is* is perhaps central in *Bill is happy*). Again, the subject is not dependent on both the 'main' verb and the auxiliary; this would require too many dependencies. We should note therefore that cooccurence restrictions between the subject and the 'main' verb do not imply that there is a dependency between these words. A similar point can be made with 'control' verbs like *expect*. In *Sue expected Fred to leave*, we only assume dependencies between *expected* and all three arguments; the work of associating *Fred* and *to leave* is done by the lexical semantics of *expected*.

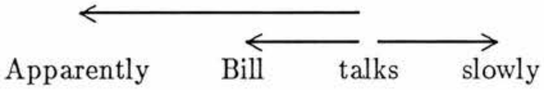
Treating dependencies as relating to a semantic representation rather than compositional operations is undermined by the fact that there is simply no clear evidence as to what the correct semantic representation is, and no clear procedure to state what relationships in this representation count as important enough to constitute dependencies.

We could then only appeal to subjective judgements about what 'feels' important for any individual sentence. This arbitrariness is not necessary when the dependencies are treated as compositional operations.

One very important consequence is that this model has a general proscription of polycephalic (ie multi-headed) dependency structures, where any word has more than one head. This is a property of classical dependency grammar, but immediately contrasts this approach with, for instance, Word Grammar (Hudson 1984), which has no such restriction. If we regard dependency structures as directed acyclic graphs, then this monocephalic (ie single-headed) framework is equivalent to the proscription of reentrancy (see Shieber 1986). It has obvious parallels to the 'single mother condition' of constituency theories, which specifies that any node can have only one mother[1] (see Sampson (1975) for a rare constituency analysis without this constraint).

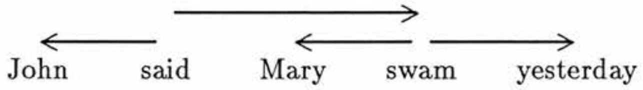
We shall also assume that the same relationship is to be found between the verb and optional modifiers as between verbs and their arguments, namely, that the verb is the head and the adjunct the dependent. This is taken to be the case whether in phrase structural terms it would be assumed to be a VP- or S-modifier. Thus the analysis of *Apparently Bill talks slowly* is:

(2.3)

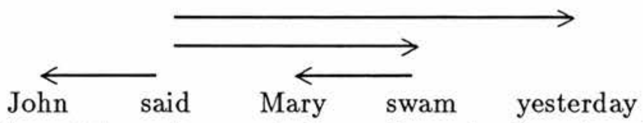


But some ambiguities are represented in this system; for instance the two readings of *John said Mary swam yesterday* are differentiated:

(2.4)



(2.5)

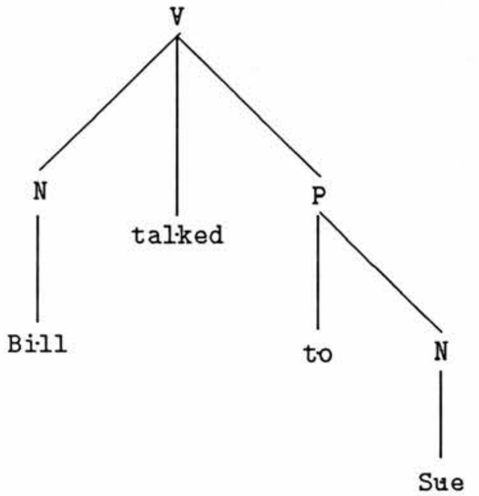


There are a number of dependency relations where the direction of dependency is controversial, for instance whether the determiner is the head of the noun or vice versa. We shall assume that the determiner is the head (see Hudson (1986) for arguments), but without being strongly committed to this.

---

[1] The S-node has no mother and corresponds to the root of the sentence.

A system of rewrite rules is possible for dependency grammar (e.g. Gaifman 1965), analogous to phrase structure rules, of the general form $X(Y_1 \ldots Y_m \; ^* \; Y_{m+1} \ldots Y_n)$. The interpretation is that X can be rewritten as $Y_1 \ldots Y_m \; X \; Y_{m+1} \ldots Y_n$ (in that order), where X is the head, represented by the asterisk, and $Y_1 \ldots Y_n$ are the dependents. We also need a rule to introduce the root of the sentence, and a set of lexical rules that allow the heads to be rewritten as lexical items. For instance, in *Bill talked to Sue*, we could have the rule V(N,*,P), which means we can rewrite V as N, V, P where V is the head and can be rewritten as *talked*. We would then need other rules P(*,N) to give us the preposition *to* and N. To rewrite both Ns, which have no dependents, we need the further rule N(*). We can then generate the dependency tree below:



Such rules are weakly equivalent to rules of context-free phrase structure grammar. They generate trees that are nearly identical to phrase structure trees that have a very 'flat' structure and have a restriction that one daughter rewrites as a lexical item (see Robinson 1970), but they fail to be strongly equivalent because they lack terminal nodes. Note that these rules must have exactly one element outside the brackets, and exactly one '*' within the brackets. A consequence of this formulation is that each dependent is *adjacent* to its head, which means that a dependent and a head cannot be separated by any elements that are not dependents of either the head or the dependent (see Fraser 1989). It is therefore not possible for dependencies to cross. The assumption of adjacency is directly related to the constraint that prohibits the crossing of branches in phrase structure trees. This requires an assumption about language universals, because the primitive notion of dependency, based on covariation, clearly does not rule out such patterns. We shall however restrict our consideration to dependency grammars with adjacency.

Monocephalic dependency grammars allow us to define a number of concepts. A is *subordinate* to B iff (i) A is a dependent of B or (ii) A is a dependent of a subordinate of B. B is *superordinate* to A iff A is subordinate to B. For instance, in *Bill talked to Sue*, *Sue* is subordinate to *talked*. A *full constituent*, with root A, consists of A plus all the subordinates of A.[2] Assuming adjacency, a full constituent must be a string. *Bill talked to Sue* has four full constituents, *Bill*, *Sue*, *to Sue* and *Bill talked to Sue*. This is obviously similar to a definition of constituent within a 'flat' phrase structure grammar. We shall use square brackets to mark full constituents (just as in phrase structure representations), so that the full constituent representation of *Bill talked to Sue* is *[[Bill] talked [to [Sue]]]*.

A more important notion is that of *dependency constituent*:

Definition 1 *A dependency constituent with root A, is a string B in which every word except A has a head within B.*

A dependency constituent is therefore a string which consists of an unbroken chain of dependencies. It is therefore a superset of full constituent. In our example, there are ten dependency constituents, the four full constituents plus *talked*, *to*, *Bill talked*, *Bill talked to*, *talked to*, *talked to Sue*. The difference between a dependency constituent and a full constituent is that in the former there is no requirement to go to the bottom of the dependency chain in all directions. *Talked to*, for example, is a dependency constituent but not a full constituent. For dependency constituents, we shall use round brackets. Obviously we shall not represent all dependency constituents at once, because in general there will be a vast degree of overlap. For instance, the dependency constituents *Bill talked* and *to Sue* would be represented as *(Bill talked) (to Sue)*, whereas the dependency constituents *Bill talked* and *Sue* would be represented as *(Bill talked) to (Sue)*. Strictly, we should label each bracket, but unless we considered overlapping dependency constituents in the same representation (which we shall have no need to do), we can ignore labelling without causing any ambiguity.

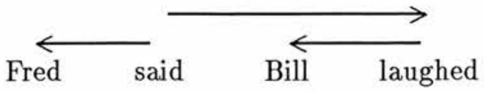
This example is rather unusual in that every string is a dependency constituent.

---

[2] The use of *root* in this sense is a straightforward generalization from *root of a sentence*.

This is not the case in *Fred said Bill laughed*, which has the dependency representation below:

(2.6)



The two dependents of *said* are *Fred* and *laughed*. *Bill*, on the other hand, is a dependent of *laughed* which is a dependent of *said*, so *Bill* is subordinate to *said* but not a dependent of it. This captures the intuition that *Fred* and *said* are related in a way that *said* and *Bill* are not. Therefore, *Fred said* is a dependency constituent, but *said Bill* is not. However, *said Bill laughed* is a dependency constituent. In fact, all the substrings of *Fred said Bill laughed* are dependency constituents except *said Bill* and *Fred said Bill*, so there are eight dependency constituents in this sentence, in contrast to the ten in *Bill talked to Sue*. Note the restriction that dependency constituents (and indeed full constituents) must be strings, and therefore cannot be discontinuous. This means that *Fred said* and *laughed* do not form a dependency constituent, even though there is an unbroken chain of dependencies here. Likewise, *gave* and *a book* do not form a dependency constituent in *I gave John a book*.

2.2.2 Comparison with Constituency

As I have shown, it is possible to define notions of constituency within a dependency-based formulation. The much more common alternative is to treat constituency as basic. In this section, I shall compare these two approaches, and then discuss the standard Transformational Grammar assumption that interpretation is dependent on syntactic structures, not rules. We have to construct a particular tree, or indeed set of trees representing different levels of structure, and then read the interpretation off these trees. I shall contrast this with the use of the rule-to-rule hypothesis, which does not require interpretation to be constructed via a particular tree.

In some respects dependency representation is more informative than constituency representation. It is not in general possible to determine dependency relations from a constituent structure representation. For instance, *Mary loves John* is standardly given the representation *[Mary [loves John]]*, with *John* forming a constituent with *loves*, to the exclusion of *Mary*. We know that there is a dependency between *loves* and *John*, but not which is the head, nor whether *Mary* is a head or not. This representation therefore does not indicate which word is the head, which is potentially a weakness, but of course we can add information to do this, as in X′ syntax (Jackendoff 1977). Also we cannot directly show covariation relations between words, like government or agreement, because the primitive way to express relations is between constituents (for instance between *Mary* and *loves John*, not *Mary* and *loves*). Again, however, we can define these relations derivatively. But a more fundamental way in which dependency grammar representations contain more information is in the flexibility of the derived notion of constituency. Constituency grammar has no notion corresponding to a dependency constituent. For instance, in *Fred said Bill laughed* there is no primitive way to express that *Fred said* forms a unit whereas *said Bill* does not. This is because *Fred said* is not a phrase structure constituent. However, as mentioned above, the intuition is that *said* goes with *Fred* in a way that it does not go with *Bill.* A central concern of this thesis is to show the considerable explanatory value in having the notion of dependency constituency.

But in other respects, standard constituency grammar is more informative than dependency grammar. This is because its constituents are more hierarchical than full constituents. In *Mary loves John*, *loves John* forms a phrase structure constituent. *Loves John* is also a dependency constituent, of course, but it has no special status, because *Mary loves* is one as well, and neither string is a full constituent. The question is whether a special division into *Mary* and *loves John* is motivated for any reason. One possibility is that it is primitive to the analysis of the sentence. Traditional linguistics usually assumes that such sentences consist of a subject and a predicate. But even assuming that this is self-evident, it is at least reasonable to think of this as relating to the semantic representation rather than the immediate structure of the sentence (as assumed for predicative adjectives and 'main' verbs above). So the primitive nature of this division is not proven. However, there is also claimed to be considerable distributional justification for this division, which we shall remark on in the next section.

Most constituency theory has assumed as in Standard Theory (Chomsky 1965) that interpretation is read off structures. Hence grammatical relations (similar to types of dependency relations) are defined configurationally. *Mary* is the subject because it is the constituent of type NP which is immediately dominated by the constituent of type S, *John* is the direct object because it is the leftmost constituent of type NP which is immediately dominated by the constituent of type VP. Other linguistically relevant notions are also defined configurationally. Fundamental to modern transformational grammars, like Government-Binding Theory (Chomsky 1981), is c-command (Reinhart 1976), which is used to explain, among other things, the restrictions that there are on coreference between anaphoric elements and potential antecedents. Defining an equivalent notion in a theory without rigid constituency would not be trivial, and therefore if c-command is crucial, then a configurational approach to grammar is supported. However, there is evidence that the relevant restrictions can be defined in other terms (see Pollard and Sag (1987), for instance). Another point is that constituency theory with such divisions can distinguish more ambiguities syntactically than can a theory with only a flat representation. As mentioned above, dependency representations do not distinguish so-called VP- and S-modifying adverbs. Therefore, the two readings of *John arrived hopefully*, 'John arrived full of hope' and 'it is to be hoped that John arrived', both simply have dependencies from *arrived* to *John* and *hopefully*. This contrasts with constituency theories which gives the two readings the representations $[[\mathit{John}]_{NP}\ [[\mathit{arrived}]_{VP}\ [\mathit{hopefully}]_{AdvP}]_{VP}]_{S}$ and $[[[\mathit{John}]_{NP}\ [\mathit{arrived}]_{VP}]_{S}\ [\mathit{hopefully}]_{AdvP}]_{S}$ respectively. But the question is why we should want to deal with this syntactically, rather than in the semantics. It is impossible to distinguish all ambiguities syntactically without radically complicating the structure. This is one reason why transformational grammars assume that there is more than one level of constituent structure. This greatly increases the amount of configurational information available. Government-Binding Theory (Chomsky 1981), for example, has a level called LF which deals with scope ambiguities.

Constituency theories which are opposed to the rule-to-rule hypothesis therefore tend towards specifying maximally hierarchical, perhaps binary-branching, syntactic structures, and may proliferate levels of representation. This means that they are extremely inflexible in the sense that it is impossible to obtain the correct interpretation for a sentence without building a highly complex and precisely specified structure. If we assume the rule-to-rule hypothesis, we are not compelled to build such structures. We can construct rigid constituent trees using phrase structure rules if we choose, as in Generalized Phrase Structure Grammar (Gazdar, Klein, Pullum and Sag 1985) for example, but we can obtain the interpretation by associating the words in an order unrelated to their constituent structure. We could in principle do this by any means at all, but, as we shall see, there are good reasons to suppose that we should pay attention to dependency structure. But first we have to be convinced that the standard arguments for phrase structure constituents are weak.

2.2.3 Linguistic Justifications for Phrase Structure Constituents

Assumptions about phrase structure grammar are based on a number of linguistic arguments. It is a standard assumption that there is a close relationship between constituents and distributional units (because this is the fundamental reason for assuming particular divisions in the first place). Hence we can determine constituent structure by applying distributional tests. This section outlines some means of distinguishing distributional units (together with one other standard test based on anaphoric properties), in order to consider whether they show that an adequate generative grammar can construct the particular types of syntactic trees proposed by standard phrase structure grammar. These tests have not all been uniformly employed by linguists, but it is the case that problems arise if non-constituents can be distributional units. These problems are especically serious if there are a number of distributional tests which distinguish non-constituents just as often as they do constituents.

I shall question whether there are any grounds for assuming that standard constituents are any better distributional units than certain supposed non-constituents. To a considerable extent, I shall concentrate on the phrase structure assumption that there is a primitive VP constituent, like *loves John* (and therefore that the subject-predicate distinction is justified on distributional grounds). The survey of tests is not exhaustive, but the general emphasis is to show that for every test that indicates *loves John* is a constituent, there is an equally good one, which may or may not have standardly been considered, which suggests that *Mary loves* is a constituent. Two of the issues discussed here will be picked up in detail in subsequent chapters.

Coordination

Coordination is often used as a test for constituency. In what is assumed to be the basic cases, conjuncts must be constituents. Furthermore, they must be constituents of like category. For instance, if *X* is grammatical in some context, and *X and Y* is grammatical in the same context, then both *X* and *Y* must be constituents, and of the same category. So if we know the category of *X*, say, we can discover the category of *Y*. If this principle has to be modified, so that non-constituents or constituents of different categories can conjoin, then the power of coordination as a test is greatly diminished. In many cases this test is adequate:

(2.7) Sue saw Bill and Fred.

Here, both *Bill* and *Fred* are clearly NPs and can coordinate. The first problem is that there are cases where constituents of unlike category can coordinate, like (2.8) below, and cases where coordination of constituents of like category sounds at best very odd, like (2.9) (where, we should note, the sentence describes a completely reasonable situation):

(2.8) John is lucky and a rogue.

(2.9) ?John was in his bed and a deep sleep.

These are presumably not a serious problem for phrase structure grammar, but do suggest that coordination is at least partly dependent on a more functional notion than simply category.

Much more of a problem is so called 'non-constituent' coordination. This is often ignored in basic textbooks (eg Burton-Roberts 1986), presumably because it is viewed as a complex phenomenon, but in fact it forces us to modify the test to such an extent that it arguably becomes useless. For instance, given the standard division into subject and predicate (as in *[Mary [loves John]]*), we should not be able to treat *Mary loves* as a conjunct, because it is not a constituent. But we clearly can, in so called 'right node raised' examples:

(2.10) Mary loves and Sue adores John.

Similarly, (2.11) is a case of non-constituent coordination, because the adverbs do not form constituents with the objects:

(2.11) John loves Mary madly and Sue passionately.

These two examples suggest that standard phrase structure grammars, which do not include constituents consisting of simply subject and verb, or object and adverb, are inadequate in themselves. We need to add new machinery (such as empty categories and/or transformations), allow very non-standard rules, or else abandon phrase structure grammar in favour of a grammar that allows these coordinations directly. The only other possibility is to rule out coordination as a test for constituency.

Fronting

Another test for constituency is that only constituents can be fronted, that is, 'moved' from their 'canonical' position to the front of the sentence or perhaps clause. For instance, we can tell that *John* is a constituent in *Mary loves John*, because the sentence *John, Mary loves* is also grammatical. This test does not overgenerate in the way that the coordination test does, but is certainly not sufficient, because there is a large range of constituents that cannot be fronted. For example, we cannot say **Loves John, Mary*, even though *loves John* is a VP. Similarly, adjectives cannot be fronted, so that **Very old, Mary knows the man* is ungrammatical. Hence the test does not give much evidence for traditional constituents.

Also it is unclear how to characterize the constituent structure of sentences with fronting. If we assume that constituents must be continuous, we cannot straightforwardly give a structure to *John, Mary loves*, for instance. As before, the problem is that *Mary loves* is not a phrase structure constituent. Again we appear to need extra machinery, like transformations or empty categories. For instance, there could be a trace after *loves*, which forms a VP with *loves*, and this trace is associated with the fronted element *John*. So fronting can be used as a test for some constituents, but leaves us with a problem over the constituent structure of a sentence with fronting.

Proforms

Another test or demonstration is that only constituents can be replaced by a proform. For instance, *John* is a constituent in *Mary loves John*, because it can be replaced by the proform *him*. Likewise *to London* is a phrase in *Fred went to London*, because it can be replaced by *there*. More interesting as a test, however, is the claim that only a constituent can serve as the antecedent of a proform. This is of course not a distributional test, but has often been considered in a similar way. For instance, in *John dislikes Mary but Mary loves him*, the antecedent of *him* is *John*, which therefore must be a constituent. Similarly, *to London* must be a constituent in (2.12) below:

(2.12) Bill went to London on Monday and Fred went there on Tuesday.

But this test is fraught with problems. For instance, proforms often refer not to an actual antecedent phrase but rather to something that has to be reconstructed from the context. For instance, *there* is presumably given the same meaning as the phrase *to London* in (2.13), even though this phrase is not actually mentioned:

(2.13) Bill lives in London but Fred never goes there.

There has nothing to be associated with in the actual sentence. Proforms must instead acquire their reference from a level of representation removed from surface form.

Another problem is that the test only works given that we already know what elements count as proforms. For instance, Radford (1988) assumes that *as* is a proform that can stand for VPs. Consider:

(2.14) John might give Mary a lift, as might Bill.

Note first that *as* cannot be replaced by a VP:

(2.15) ?*Give Mary a lift might Bill

Nor can it stand for all VPs:

(2.16) *John gave Mary a lift, as Bill.

But we do have some limited evidence to treat *give Mary a lift* as a constituent from (2.14). Therefore consider:

(2.17) John might give Mary a lift, plus Bill.

On the reading where John might give Bill a lift, *plus* may be viewed as anaphoric to *John might give a lift*, which is not a traditional constituent. In *Mary loves John,* *plus Fred*, on the reading where Mary loves Fred, a similar argument can be made about *Mary loves.* All we can do is decide that only certain (closed class) anaphoric expressions count as proforms, presumably those that refer to standard constituents, but then the test becomes completely circular.

Ellipsis and Sentence Fragments

Another common test for constituency is that any sentence fragment must be a constituent. For instance, *Across the road* can serve as an answer to the question *Where did Mary run?* Similarly, it is also argued that fragments that can undergo ellipsis must be constituents. But there is no principled way of knowing whether any given fragment is to be tested by what is left, or by what is missing. *John did* is a possible answer to the question *Who did the work?*, and so if *across the road* is a constituent, then *John did* should be too. If on the other hand, *John did* is taken to indicate that the constituent *the work* has undergone ellipsis, then we can also regard *Mary ran* as a constituent.[3] Either way round, the test allows a string which is not a traditional phrase structure constituent. It is also possible to find fragments where neither what is missing nor what is left is a constituent. For instance, a possible response to *What will John do for the charity?* is *Give some books*, which is not a constituent, but then the missing fragments are (presumably) *John will* and *to the charity*, which do not constitute a constituent together.

Conclusions

The above analyses show that the 'tests' for constituency in fact give support for the constituent status of supposed non-constituents. For instance, there is a lot of evidence to regard *Mary loves* as a constituent in *Mary loves John.* We can coordinate with it, we can find elements that can be regarded as proforms anaphoric to it, and it is possible to use it as a sentence fragment or to make it undergo ellipsis. In fronting, it is the kind of unit that can be left behind in the 'body' of the construction. Therefore we can make the claim that dependency constituency is a more appropriate measure of constituency than traditional constituency. As already mentioned, phrase structure constituency has no way to regard *Fred said* as any more of a constituent than *said Bill* in *Fred said Bill*

---

[3] Note also that determining the antecedent is not always straightforward either, as with proforms.

laughed, which is surely counter-intuitive. We shall not make any detailed consideration of the relation between the tests and dependency structure at this point, but shall return to some of these issues in the next chapter. The purpose of the section has been to indicate that the case for a constituent structure as defined in phrase structure terms is not very strong, and hence that there is no linguistic imperative to presume standard phrase structure models. It is worth noting this in the light of Pulman's comment above.

In the next section I shall argue for the use of having a flexible constituency representation when considering sentence processing. I shall claim that a fundamental weakness of constituency theory is that it has no primitive means of constructing such a representation; it will have to use some indirect means to obtain interpretations for units which are not constituents. Given that the other arguments for a constituency-based system are inconclusive, we claim that a dependency-based system is to be preferred.

2.3 Grammar and Incremental Interpretation

2.3.1 Introduction

In this section I shall claim that it is parsimonious for a grammar to take into account how straightforward it is to derive an account of human sentence processing from it. I shall concentrate on one clear issue of psychological plausibility, that we process language in an essentially incremental or 'left-to-right' manner. The section discusses what degree of incrementality sentence processing involves, and concludes that the construction of a unified meaning is at least possible whenever a dependency is formed. It then shows that this is straightforwardly compatible with the dependency grammar framework outlined above; parsing involves the construction of dependencies and interpreting the resulting dependency constituents. On the other hand, phrase structure grammars have to assume a mechanism for interpreting selected non-constituents, which requires the specification of a non-trivial interface between syntactic description and interpretation. Given the linguistic evidence outlined above, I claim that it is better to assume dependency grammar, which does not require such an interface.

I assume that there is no restriction on possible methods of investigating knowledge of language. Hence interesting approaches to linguistics cannot be restricted to acceptability judgements. This is surely necessary if we wish to regard linguistics as a component of cognitive science, and as striving toward explanatory adequacy. Most methodological theory probably assumes this position, but little actual research makes use of a wide range of methods to argue for particular analyses. However, modern transformational grammar does overtly assume that learnability at least is an important constraint (Chomsky 1986; Wexler and Culicover 1980). Also, arguments based on computability or computational complexity do influence the direction of some grammatical theories at least (obvious examples are Lexical Functional Grammar (Bresnan 1982) and GPSG (see Gazdar 1982)). Hence there can be no objection in principle to the use of evidence from human sentence processing. One major problem is of course that it is extremely hard to prove that experimental methodologies access what we want them to, namely the linguistic representation, rather than some level involved in parsing. It is possible that using grammaticality judgements faces similar problems, because at least some workers assume that sentences can be generated by the grammar but not be acceptable (eg Morrill 1988), or perhaps that they can be ungrammatical but acceptable (eg Chomsky 1970:193-5). If either position is correct, there is clearly no way to assess grammaticality directly from acceptability judgements, and the best way forward would seem to be to use some combination of methods. But quite probably, the problems with experimental methodologies can be more severe. So, in principle the arguments should be seen as relevant to linguistic theory, but in practice it may be hard to argue that any particular experimental results are. In later chapters I shall argue that some experimental results are directly relevant. But of more immediate importance is the claim that it is parsimonious to assume a grammar where there is a correspondence between constituents and interpretable units if at all possible. We shall now consider this point.

2.3.2 Incremental Interpretation

Arguments for Incremental Interpretation

When we process the sentence *Mary loves John*, there is a strong intuition that we interpret *Mary loves* as a unit before we encounter *John*. This conflicts with the standard assumption in constituency theory that *Mary loves* is not a constituent. Let us first assess the strength of this intuition.

One possibility is that our intuitions are wrong. This is possible, although it seems a strange mistake. In speech it might be due to the speed with which most sentences are uttered. Let us also assume that either in speech or writing we process words in the same order that they are spoken or written. There is some evidence that this is too simplistic (Bard, Shillcock and Altmann 1988) in speech, but temporally earlier words will of course have to be remembered, and we can assume that the deviation cannot be great. We shall therefore proceed with this idealization.

It is possible that our intuitions are wrong about the interpretability of *Mary loves*. This is also conceivable for *Bill talked* and *Bill talked to* in *Bill talked to Sue*, where neither of these strings is a phrase structure constituent. But this becomes far harder to believe with a classically 'right branching' sentence like:

(2.18) I saw the dog that chased the cat that bit the rat that nibbled the cheese.

A constituency analysis of this sentence will almost certainly be:

(2.19) [I [saw [the [dog [that [chased [the [cat [that [bit [the [rat [that [nibbled [the [cheese]]]]]]]]]]]]]]]].

If we assume that we cannot interpret a sentence until we have a complete constituent, we must remember all the words as individual units, without being able to perform any interpretation at all. Only after encountering the final word, *cheese*, can we do anything, and then we have to do everything at once. This is much more clearly at odds with intuitions, given that there is no problem at all in interpreting this sentence. But there is a stronger argument, not based on intuitions. If no interpretation is allowed before constructing constituents, we will be forced to remember fifteen words separately before we reach *cheese* (and then be able to recall them in reverse order). This is almost certainly an operation beyond the power of short-term or working memory (see Miller 1956), and, as we know, nested constructions of comparatively low 'grammatical' complexity rapidly become uninterpretable (Chomsky 1965):

(2.20) I saw the rat that the cat that the dog chased bit.

A 'right-branching' sentence may conceivably be easier than a nested one for some other reason, and fifteen words may possibly be within memory capacity. But we can extend the sentence still further, without making it uninterpretable:

(2.21) I saw the man that owned the dog that chased the cat that bit the rat that followed the mouse that nibbled the cheese.

If we keep extending this sentence, we must *at some point* exceed human memory or processing resources. This is based on the simple assumption that our capacities must be finite. Hence whatever we do in sentence processing, it cannot be that we always delay interpretation until we reach the end of the phrase structure constituent, as for instance Fodor, Bever and Garrett (1974) and Clark and Clark (1977) argue.

There are a number of other good arguments for incremental interpretation. An incremental system makes it far easier to interpret an incomplete utterance which does not form a standard constituent, for instance if the speaker is interrupted after saying *Mary loves*, or if there is some minor dysfluency. A listener clearly picks up some information from this. If the sentence is interpreted incrementally, we have no problem. If we cannot do incremental interpretation, then we must have a special means to interpret incomplete utterances. Another point is that it must be more efficient to interpret incrementally than to wait until a phrase structure constituent is finished, because presumably all operations take time, so we want to do as many as we can as soon as we can; the less we do incrementally, the more we will have to do at the end. Even processing *Mary loves John* non-incrementally would require us to do two operations, combining *loves* and *John*, and combining *Mary* and *loves John* one after the other when all the sentence has been heard. With (2.18) and (2.21), the problem is of course far more severe.

There is also an argument for incremental processing based on universal tendencies in word order. Steedman (1989) points to the fact that the vast majority of languages have either SOV or SVO basic word order, both of which are of course 'right-branching', assuming the existence of a VP constituent which excludes the subject NP. If there were no incremental processing, then there would be a strong tendency for languages to be harder to process than they need to be. In fact, having a flat structure would not help; we would need there to be a constituent formed from SO or SV, to allow incremental interpretation. We would expect that most or all languages would be of the form OVS or VOS (with a VP constituent). These word orders are however extremely rare (Greenberg 1966; Hawkins 1983). Hence it does not seem plausible that we process a VP before we associate it with a subject. Not having incremental interpretation is therefore very unlikely given the facts about language typology. Likewise, we would expect strongly 'right-branching' structures, as in (2.18), to be very rare, which is clearly not the case.

Finally, experimental evidence shows that we begin interpretation before we reach constituency boundaries, indeed within one or two syllables of input. Marslen-Wilson (1973) used a 'shadowing' technique, where listeners are required to repeat aloud what they are hearing as accurately and with as little delay as possible. Some subjects could shadow the passage as close as 300 ms, a latency of roughly one syllable. Marslen-Wilson corrupted some of the words in the passage, for instance changing *company* to *compsiny*, and found even the best shadowers often restored these corruptions, but crucially these restorations were more likely in predictable contexts. Hence at least some interpretation is very rapid and phrase structure constituency boundaries are not relevant.

Similar conclusions come from lexical ambiguity studies. Associative priming effects are well known for words presented immediately after one another (eg Meyer and Schvaneveldt 1971). Typically in these experiments, a string of letters is presented on a screen, and subjects have to decide as quickly as possible whether it is a word or not. This is called a lexical decision task. If the string is a word, for instance *butter*, then subjects are faster responding 'YES' when the word is preceded by an associate, such as *bread*, than when it is preceded by an less related word, such as *nurse*, or by no word at all. This preceding word can be presented visually or aurally (hence, cross-modally). Swinney (1979) studied this effect in sentential contexts. Subjects listened to sentences like:

(2.22) The man was not not surprised when he found several spiders,
roaches and other bugs in the corner of his room.

Bugs can mean either 'insects' or 'eavesdropping devices'. Hence three target words were used, *ant*, *spy* and a neutral word, *sew*. If the target word was presented immediately after *bugs*, both *ant* and *spy* were responded to faster than *sew*, so both senses were primed. On the other hand, 300 ms later, only *ant* was responded to faster than *sew*, and so by this point only the contextually appropriate sense was primed. Hence the context must have selected the relevant meaning of *bugs* within 300 ms, and therefore some interpretation of the string must have taken place.

In conclusion, there is overwhelming evidence that we interpret sentences incrementally, without having to wait until the end of phrase structure constituents. Let us now assess the effect that this has on grammatical description.

Grammatical Consequences

Our intuition that we can interpret *Mary loves* in *Mary loves John* suggests that there is no straightforward relation between phrase structure constituents and interpretable units or strings. It is therefore reasonable to hypothesise that the parser does not need to be able to construct phrase structure constituents at all. We shall return to discuss this later. An alternative assumption is that the fundamental operation is dependency formation. *Mary loves* is a dependency constituent, of course, because *Mary* is a dependent of *loves*. Let us therefore make the claim that every dependency constituent is interpretable. Because we have defined dependency relations in *compositional* semantic terms, and have assumed the rule-to-rule hypothesis, this is not unreasonable. Let us therefore define two further concepts, *Maximal Dependency Constituent* and *Incremental Dependency Representation.* The former is taken from Barry and Pickering (1990):

Definition 2 *Let* α *be any string. Suppose* α *can be divided into substrings* $\langle \alpha_1, \ldots, \alpha_n \rangle$ *(where* $n \geq 1$*) such that each* α_i *is a dependency constituent, but there is no* i *such that* $\alpha_i \alpha_{i+1}$ *is a dependency constituent. (It is easily shown that such a division must exist and be unique.) This is then called a division of* α *into maximal dependency constituents (MDCs).*

Definition 3 *The incremental dependency representation (IDR) for a sentence with words* $\langle \beta_1, \ldots, \beta_n \rangle$ *(where* $n \geq 1$*) are the maximal dependency constituents for the strings* $\langle \langle \beta_1 \rangle, \ldots, \langle \beta_1, \ldots, \beta_n \rangle \rangle$.

We have already introduced a round bracket notation for dependency constituents, and shall therefore represent MDCs and IDRs like this as well on occasion, so that, for instance, the MDCs $\langle X, Y, Z \rangle$ can be represented as *(X)(Y)(Z).*

Therefore the IDR for the sentence *Mary loves John* are *(Mary)*, *(Mary loves)* and *(Mary loves John)*, because whenever another word is added, it is integrated into a single dependency constituent. On the other hand, the IDR for *Fred said Bill laughed* is *(Fred)*, *(Fred said)*, *(Fred said)(Bill)* and *(Fred said Bill laughed)*. The important point is that *Fred said Bill* requires two MDCs.

We have claimed that every dependency constituent is interpretable, so sentences where there is never any need to construct more than one MDC should be straightforward to process incrementally. This seems to be supported. For instance, (2.18) has the IDR *(I)*, *(I saw)*, *(I saw the)*, *(I saw the dog)*, *(I saw the dog that)*, *(I saw the dog that chased)*, etc. There is no point at which we have to remember more than one MDC. The fact that (2.18) and its extension (2.21) are interpretable follows directly. Every additional word can be integrated into a single dependency constituent. In contrast, the IDR of (2.20) is not so straightforward. After *dog*, there are three MDCs, *I saw the rat that*, *the cat that* and *the dog*, which we can represent as *(I saw the rat that) (the cat that) (the dog)*. This presumably takes up or exceeds memory resources.[4] So a processing account based on dependency grammar can explain the difference between 'right-branching' and nested sentences in a straightforward way.

We can therefore make the following hypothesis, the *Dependency Interpretation Hypothesis* or *DIH*:

Hypothesis 1 *The sentence processor gives all and only maximal dependency constituents an interpretation.*

Let us make use of a meaning representation involving the lambda calculus. We shall assume the principle of compositionality, that the meaning of the whole is the composition of the meaning of the parts. We take the meaning representation to be based on function-argument structure, and we shall represent the meaning of words in bold font, so that the meaning of *Mary* is **mary**, and the meaning of *walks* is **walks**. We shall represent functional application by concatenation, with the function coming before the argument, so that **walks mary** means the meaning of *walks* applied to the meaning of *Mary*. Furthermore, we shall assume a convention of left-associativity for function application, so that **XYZ** means (**XY**)**Z**, and therefore that **loves john mary** means (**loves john**) **mary**. The lambda calculus allows abstraction

---

[4] However, not all strings with three dependency constituents are impossible to process. In German, for instance, it is possible to have a subordinate verb with three dependents before the head. What seems crucial is that these are dependents of the same head, whereas in nested constructions, the matching is between many dependents and many heads. I shall say nothing more about such constructions.

over variables as well as application. For example, λx [**loves** x **mary**] is the function that gives **loves john mary** when applied to **john**, **loves (the man) mary** when applied to **the man**, and so on. In general, an expression of the form $(\lambda y[\alpha])\beta$ will be equal to an expression of the form $\alpha[\beta/y]$, which means α with β substituted for y. The process of converting the former form to the latter form is known as β-reduction.

In the next chapter we shall consider a procedure for associating lambda calculus meaning representations with natural language expressions. For instance, we shall see that *Mary loves* can be given the meaning λx [**loves** x **mary**], even though it is not a traditional constituent.

Let us apply the DIH to the processing of *Mary loves John.* After interpreting *Mary loves*, we encounter *John.* Our interpretation of this can then be given the semantic representation (λx[**loves** x **mary**]) **john**, which can be β-reduced to **loves john mary**. This is legitimate because *Mary loves John* is a dependency constituent. Let us contrast this with *Fred said Bill laughed.* The IDR is (using the round bracket notation for each MDC) *(Fred)*, *(Fred said)*, *(Fred said)(Bill)*, *(Fred said Bill laughed).* After *Bill* we cannot form a single dependency constituent, because there is no unbroken chain of dependencies, but when *laughed* is reached a single chain is formed, with *laughed* the head of *Bill* and a dependent of *said.* Hence, the Dependency Interpretation Hypothesis claims that *Fred said* is given the interpretation λf[**said** f **fred**], but that *Fred said Bill* is not given an interpretation as a unit. Most accounts, such as those based on phrase structure grammar, would agree with this, but it is of course possible to devise a method of giving this string a meaning represented in the lambda calculus. The meaning of *Fred said Bill* is that function that gives the meaning of the sentence *Fred said Bill laughed* when applied to the meaning of *laughed*, the meaning of *Fred said Bill loves Sue* when applied to the meaning of *loves Sue*, and so on. The function can be represented as λf [**said** (f **bill**) **fred**]. Hence the claim that we do not interpret n*Fred said Bill* as a unit is indeed a strong claim. However the DIH holds that the two MDCs *Fred said* and *Bill* are kept semantically distinct, as λf [**said** f **fred**] and **bill**. When *laughed* is reached, however, a single representation is formed, [**said(laughed bill) fred**].

Why should this be? The only evidence that I have produced against this is that nested constructions like (2.20) do not allow the formation of a single IDC and are hard to process. This can be explained by the fact that they require at least three IDCs, and therefore three units of representation. This may exceed working memory space. This is probably preferable to the most obvious alternative, that a single representation is formed but it is complex in some way, and therefore harder to process. This alternative can be called the *Structurally Complete Interpretation Hypothesis* or *SCIH* (after Moortgat 1988). In it every word can be interpreted immediately it is encountered, irrespective of the dependency structure. In this system, we have to work out some additional property, presumably of the semantic representation, that will tell us when a sentence becomes impossible to process. This account obviously relates less directly to any notion of dependency or constituency, and so is linguistically less attractive, but more clear evidence in favour of the DIH would come if we could show that dependency constituents were needed in order to obtain good linguistic analyses for particular constructions. But we have already seen some linguistic motivation for dependency constituency, and the processing evidence fits well with this. Incremental interpretation is therefore consistent with dependency representations, and allows us to make the parsimonious conclusion that interpretable units are always constituents.

It is possible to assign incremental interpretations to systems such as phrase structure grammar, where the interpretable units are not always constituents. In a sense, such a system is less parsimonious than one with the DIH and dependency constituency, simply because we have to specify two sets of units, one the constituents, the other the units which are given an interpretation in processing. This would be necessary if there were strong evidence that the assumptions of phrase structure grammar were correct and that any other system had fundamental linguistic flaws, but the evidence presented above suggests that this is not the case. However, we can distinguish interpretable units from constituents by assuming some structure to the syntax-semantics interface.

We shall consider some elements of Pulman (1986), who shows one possible way of getting interpretations for non-constituents, making use of a phrase structure grammar with the rule-to-rule hypothesis, where a semantic operation is associated with each syntactic rule. First, let us consider his general method. Consider again *Mary loves John.* Phrase structure grammar analyses this with the rules S $\longrightarrow$ NP VP and VP $\longrightarrow$ V NP. Let us associate an interpretation with these rules so that S $\longrightarrow$ NP VP has the interpretation **VP NP**, and VP $\longrightarrow$ V NP the interpretation **V NP**, where **V**, **VP** and **NP** are the meanings of V, VP and NP respectively. Hence we give the VP *loves John* the interpretation **loves john**, and the full sentence the interpretation **loves john mary**. Pulman assumes a left-corner parser, where a rule is applied whenever the leftmost category on the right hand side is found. The category is now taken to be the mother category needing the other right-hand-side elements. Whenever this is done, lambda abstraction is performed in the semantics, over variables corresponding to the semantics of the elements that have not so far been found. With *Mary loves John*, we find *Mary*, which is of category NP and has interpretation **mary**. We then discover that NP is the left-corner of the rule S $\longrightarrow$ NP VP, and so now assume that *Mary* is of category S, needing a VP, and having interpretation $\lambda vp\ [vp\ \mathbf{mary}]$. The semantic operation involved is type-raising, discussed in the next chapter. Next we find *loves*, of category V and interpretation **loves**, but since we have the rule VP $\longrightarrow$ V NP, we now assume that it is of category V, needing an NP, and having interpretation $\lambda np\ [\mathbf{loves}\ np]$. At this point, we can in principle (see below) apply another operation to combine these together to give us *Mary loves*, of category S, needing an NP, with interpretation $\lambda x \lambda vp\ [vp\ \mathbf{mary}]\ (\lambda np\ [\mathbf{loves}\ np](\mathrm{x}))$, which β-reduces to $\lambda x \lambda vp\ [vp\ \mathbf{mary}]\ (\mathbf{loves}\ x)$, which in turn reduces to $\lambda x\ [\mathbf{loves}\ x\ \mathbf{mary}]$. This operation is one of function composition, which will also be considered in the next chapter. We have therefore obtained an interpretation for *Mary loves*, with phrase structure rules, even though it is not a phrase structure constituent. This interpretation can then be combined with the interpretation of *John*, so that the full sentence has the interpretation **loves john mary**.[5]

This shows that it is possible to have a rule-to-rule relationship between syntax and semantics, and, given the appropriate restrictions on parsing, allow us to interpret non-constituents. However, these restrictions have to be defined independently of the rules, and some machinery is needed to allow us only to interpret the units that we wish to interpret. For instance, we can employ the same principles to give an interpretation to *Fred said Bill* in *Fred said Bill laughed*. Abbreviating somewhat, *Fred said* can be given

---

[5] The above does not constitute a description of Pulman's proposals, but rather an outline of the general framework which shows how incremental interpretation is possible within phrase structure grammar. In fact, he proposes that function composition only occurs when the number of elements that has to be remembered is quite large, so that function composition is a sense an option of last resort for the parser. Therefore he would not predict that an interpretation is given to *Mary loves*, although 'right-branching' sentences will allow some degree of incremental interpretation. Nested constructions on the other hand cannot be interpreted, because function composition is prohibited in such cases, and too many elements need to be remembered. Pulman's restrictions on combination sometimes correspond to dependency constituency, but sometimes do not. We shall not consider his precise predictions in this thesis.

a category S, needing an S, and interpretation λs [**said** s **fred**], and *Bill* category S, needing a VP, and interpretation λvp [vp **bill**]. These can be combined to give an interpretation which β-reduces to λx [**said** (x **bill**) **fred**]. So if we want to prohibit such an interpretation, we need to differentiate permissible from impermissible semantic operations, or alternatively prevent top-down information being applied in selected instances. Any such method will contrast unfavourably with the analysis in terms of dependency constituents and the DIH, because this only requires a completely bottom-up parsing operation, which assumes the existence of dependency constituents only if all the words have been found. Only the current account has a a direct relationship between syntactic constituents and interpretable units.

Let us briefly consider other alternatives to assuming the reality of dependency constituents in parsing. First, making use of the rule-to-rule hypothesis is not necessary to incrementally interpret a phrase structural theory. We could build a structure for a non-constituent fragment like *Mary loves*, by applying phrase structure rules in a top-down manner as before, so that we compute a structure $[Mary[loves\ldots]$, where the dots represent information that has not been encountered yet. We then have apply a rule which converts structures of the form [NP [V ...]] to the interpretation λx [**V** x **NP**]. The issues here are the same as with the rule-to-rule version above. If we have a totally top-down system, we simply allow every string to be interpreted, and assuming that this is not desireable, we have to impose some restrictions on top-down processes, or some restriction in what semantic operations can be performed. Alternatively, we could employ Gaifman's dependency grammar rules rather than phrase structure grammar rules. *Mary loves John* would therefore be interpreted by using the rule V(N,*,N). With or without rule-to-rule semantics, we get the interpretation **loves john mary** (we will have to index the Ns with the appropriate Ns in the semantics). In order to get an interpretation for a dependency constituent that is not a full constituent, we need to apply top-down processes again, and we have to add mechanisms, as before, if we wish to prevent the interpretation of all strings, whatever their dependency structure.

We can conclude that it is possible to give interpretations to exactly those strings which constitute dependency constituents within a phrase structure grammar, but any means that we have of doing this will also allow the interpretation of strings which are not dependency constituents unless restrictions are imposed. We have to make decisions about the degree of top-down processing available, and about the way that interpretations are given to non-constituents. Given the evidence presented above that using phrase structure grammar with rigid constituency may have limited linguistic justification, it seems preferable to employ a grammatical framework that is not based on phrase structure constituency, but which will instead allow us to generate dependency constituents as syntactic units, and will give exactly these units an interpretation in terms of the rule-to-rule hypothesis. We shall discuss what framework we want in the next chapter. But first, we must address some evidence that it has been claimed shows that phrase structure constituents are psychologically real.

2.3.3 Psycholinguistic Arguments for Constituency

We have considered the standard linguistic arguments for phrase structure constituency. Similar assumptions are often made in psycholinguistics. See Pulman's comment above, for example. Pervasive psychological concepts, such as Minimal Attachment (Frazier (1978), c.f. Kimball (1973)) are defined in terms of phrase structure tree configurations. These assumptions are obviously incompatible with the claim that constituent structure is not primitive to parsing (though they are extremely controversial themselves; see Crain and Steedman (1985); Altmann and Steedman (1988), and to some extent it may be possible to reinterpret them in non-phrase-structural terms (see Altmann 1988)). We shall return to these issues in chapter 6. We should therefore consider whether there is any psychological evidence that unambiguously supports the construction of phrase structure trees, and, if this process does occur, whether it has to occur during actual sentence processing or whether it should be seen as a by-product of a particular kind of reflection on the structure of language.

Human cognition is bound to impose some structure on a complex set of stimuli such as the words in a sentence. Therefore it is to be expected that, given some set of lexical categories, we can discover what we might term 'psychological constituents'. There is evidence that these 'psychological constituents' bear a considerable relation to linguistic assumptions about constituent structure. It is, however, less clear whether they correspond to phrase structure constituents, or to dependency and full constituents as defined in terms of our compositional dependency grammar.

Fodor, Bever and Garrett (1974:249-254) describe a number of experiments showing psychological evidence for some kind of constituent structure. For instance, Johnson (1965) made subjects learn lists of sentences, and found that the probability that they would incorrectly recall a particular word when they correctly recalled the previous word increased at the beginning of major phrases. This suggests that subjects learn sentences by segmenting them into major phrase groups, and so when a part of a phrase is learned, the rest of it tends to be learned as well. Several other studies show greater degrees of association between words in the same constituent than between words in different constituents.

This strongly suggests, as would be expected, that linguistic notions of constituency are not wholly unrelated to psychological reality. There are a number of qualifications, however. First, there is no reason why these results cannot be interpreted as showing the psychological reality of dependency relations. Words in the same constituent are in general going to be close to each other in dependency terms as well. Therefore a dependency interpretation is possible as well as a constituency interpretation.

Second, it is not clear that phrase structure constituency is supported rather than full or dependency constituency. Fodor, Bever and Garrett (1974:252,329-342) outline studies based on the 'click' paradigm (Ladefoged and Broadbent 1960; Fodor and Bever 1965). Fodor and Bever had subjects listen to sentences where a click has been inserted before, after, or at a major constituent break, and found that subjects often assumed that clicks before or after the break were closer to the break than they actually were. Again, this gives some evidence for the psychological reality of constituents, and is at least closer to being on-line and perceptually-driven than memory-based tasks. There is also evidence that the results are not simply due to acoustic properties of the sentences, but represent some deeper analysis (Garrett, Bever and Fodor 1966). However, there is no evidence that the magnitude of the effect is dependent on the number of constituent boundaries at the constituent break. Therefore it certainly will not give a very accurate measure of constituent structure. But also Bever, Lackner and Stolz (1969) found no evidence that the SV boundary was more salient than the VO boundary in a simple SVO sentence. Hence, assuming the relevance of the technique, it gives stronger evidence for a constituency representation derived from dependency grammar, where SV and VO are equally important units, than a phrase-structure representation.

Third, it is not clear that the psychological evidence reflects any representation that is actually constructed during processing rather than a structure that is derivative from a semantic representation which is built up without the direct use of any structural representation at all (as the incremental evidence would suggest). This possibility is stronger in relation to other psycholinguistic studies of constituency such as Levelt's (1970) multidimensional scaling technique. Subjects were simply asked to rate the degree of relatedness between all possible pairs of words in a sentence. Not surprisingly the results of this study bear considerable resemblance to constituent structure representations, especially given that some issues of ascertaining constituent structure are dependent on intuition. But these results could be epiphenomenal to sentence processing, or might not even indicate any reality of standard phrase structure trees at all. One likely possibility is that Levelt's results bear on the 'propositional' semantic representation, not a syntactic representation or compositional semantic operations. For instance, noun phrases are universally regarded as both basic syntactic constituents and as semantic units, and hence would be represented as such within a propositional representation such as a logical form. For example, Levelt found close connections between *the* and *boy*, and between *a* and *dollar*, in *the boy has lost a dollar*. This result is open to different interpretations. But one possibility is that Levelt's study shows how we ascribe structure to a sentence if we are asked to do it, and that we base this at least partly on a conceptual representation of the sentence.

These arguments show that there is no conclusive psycholinguistic evidence that we construct a structural representation in sentence processing, or that the constituents proposed in phrase structure grammar are reflected in the mental representations we form during sentence processing. This removes a possible objection to the claim that there is a consistent account of sentence processing based on dependency grammar.

2.4 Summary

In this chapter we have argued that a linguistic theory based on dependency relations is superior to one based on phrase structure constituents for allowing a straightforward representation of the process of language comprehension. Dependency grammar allows constituency representations to be generated directly from the dependency relations, so the ability to make reference to constituency is not lost. Many of the phrase structure trees prevalent in contemporary constituency theory will not be directly generated, but then the syntactic arguments for their existence are weak. There is good evidence that language comprehension involves the interpretation of dependency relations, and therefore the most parsimonious theory will have these relations directly represented in the grammar.

This is consistent with the compositional-semantic dependency formulation proposed in this chapter. But we do not have any adequate characterization of the generative rules that we need. We need rules that will at least allow us to form dependency relations whenever they occur. We can then construct a linguistically motivated generative theory of grammar which will allow us to account for language comprehension in a direct manner. The grammar will then embody a powerful form of the Strong Competence Hypothesis.

Chapter 3

Categorial Grammar and Dependency

3.1 Introduction

This chapter outlines a linguistic framework in which dependency and dependency constituency can be defined, and which has the power to allow a transparent representation of incremental interpretation. It introduces the notion of a (directional) categorial grammar, and then shows that there is a close relation between the simplest type of categorial grammar and dependency grammar. However, this system is inadequate because it fails to allow the degree of incremental interpretation that we have found necessary. Hence the chapter then considers an extended categorial framework, the Lambek Calculus, and then shows how to represent dependency relations and dependency constituency within this framework. This system allows us to give an interpretation to the dependency constituents. The chapter then shows how this description of dependency constituency within the Lambek Calculus allows an elegant account of which strings can be coordinated. It concludes by considering what degree of incremental interpretation is actually employed by the sentence processor, and applies this to the analysis of processing complexity.

3.2 An Outline of Categorial Grammar

3.2.1 Basic Concepts

In categorial grammar, expressions of a language are given categories or syntactic *types*, defined recursively from a set of basic types. A directional categorial grammar defines its types as follows:

a. If X is a basic type, then X is a type.
b. If X and Y are types, then X/Y and $Y\backslash X$ are types.

An expression of type X/Y is one which combines with an expression of type Y to its immediate right to form an expression of type X. An expression of type $Y\backslash X$ is one which combines with an expression of type Y to its immediate left to form an expression of type X.[1] The basic set of types will have at least S, NP and N as members. NP is a type for the same set of expressions as it is in phrase structure grammar. This includes names such as *John* and *Mary*, and complex definite expressions like *the president of Romania.* S is also used as in phrase structure grammar, e.g. *Mary loves John* is of type S. N is the type of noun, roughly equivalent to N′ in X′ theory (Jackendoff 1977), and includes, for instance, *president of Romania.* The types for other expressions are defined derivatively. For example, *loves John* has the type NP\S, because it can combine with an NP to its immediate left to give a sentence of type S. Similarly, the verb *loves* has the type (NP\S)/NP, because it can combine with a NP to its immediate right to give an expression of type NP\S.[2] Determiners such as *the* have the type NP/N, because they can combine with an N to their immediate right, such as *president of Romania*, to give an NP. In this way we can give types to any expression we choose. In a non-directional categorial grammar (e.g. Ajdukiewicz 1935), only one slash (/) is used, and an expression of type X/Y can combine with a Y either to its right or its left to give a type X.

---

[1] This convention is the one proposed by Lambek (1958) and differs from that used by, for instance, Steedman (1987), who uses $X\backslash Y$ in place of $Y\backslash X$.

[2] Note the implicit 'VP-ordering' here; the object NP is combined with before the subject NP. We shall return to this later.

3.2.2 AB Grammar[3]

The simplest form of (directional) categorial grammar is known as the Ajdukiewicz-Bar-Hillel or **AB** system, after Ajdukiewicz (1935) and Bar-Hillel (1953) (who introduced directionality). The only rule of combination allowed is called *functional application*, by virtue of its semantics, in its forward and backward forms. We represent function application by juxtapostion, with f standing for a functor and y standing for an argument, with fy meaning f applied to y. The syntax is indicated before the colon, the semantics is indicated after the colon:

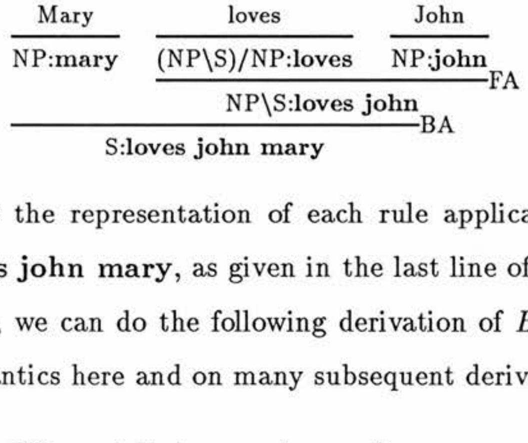
Forward Application (FA) $X/Y : f \;\; Y : y \Rightarrow X : fy$

Backward Application (BA) $Y : y \;\; Y\backslash X : f \Rightarrow X : fy$

We shall continue to assume left-associativity.

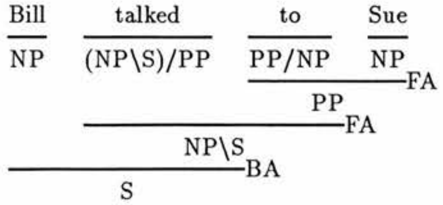
Ades and Steedman (1982) propose a notation for categorial grammar analyses, which amounts to an inversion of phrase structure trees. In the directional system we can derive *Mary loves John* as follows:

(3.1)



Note the representation of each rule application. The meaning of this sentence is **loves john mary**, as given in the last line of the derivation. If we add PP as a basic type, we can do the following derivation of *Bill talked to Sue*. We shall miss out the semantics here and on many subsequent derivations for perspicuity:

(3.2)



This has the meaning **talked (to sue) bill**. In this example, the preposition *to* is given the type PP/NP, which means that it is a PP lacking an NP to its immediate right. The verb *talked* is given the type (NP\S)/PP, because it is a sentence lacking a PP to

---

[3] This section is derived from joint work with Guy Barry. See Barry and Pickering (1990) (which is bound into this thesis).

its immediate right and an NP to its immediate left.

Representation of Constituency and Dependency

AB grammar forces us to associate elements in a particular, rigid way, like phrase structure grammar. We can represent the constituent structures of our examples as *[Mary [loves John]]* and *[Bill [talked [to Sue]]]*, though the constituent labels, which can be read off the derivations, will be unconventional (eg NP\S rather than VP). So **AB**, plus a suitable lexicon, could be regarded as having the rule S ⟶ NP NP\S, which is equivalent to S ⟶ NP VP. But unlike most versions of phrase structure grammar, **AB** is strictly binary branching. For instance, a phrase structure grammar may have a rule VP ⟶ V NP PP, which could have no equivalent in **AB**.

Categorial grammar is, as we have seen, based on the association of functors and arguments, not heads and dependents. However we can draw parallels between heads and functors, and between dependents and arguments. We can give the following obvious definition (where the semantic type $Y \to X$ collapses the two syntactic types X/Y, $Y\backslash X$):[4]

Definition 4 *When a string of a functor type* $Y \to X$ *combines with a string of an argument type* Y *to form a string of type* X*, the functor string is the phrasal head and the argument string the phrasal dependent.*

In *Bill talked to Sue*, *talked to Sue* is of type NP\S and *Bill* of type NP. Hence *talked to Sue* is the phrasal head, and *Bill* is the phrasal dependent. Likewise, *to Sue* is the phrasal dependent of *talked*, because *to Sue* has type PP, and *talked* has type (NP\S)/PP. Because these definitions are phrasal, they do not relate directly to the lexical notions of head and dependent defined in the previous chapter. But we can derive these indirectly:

Definition 5 *When a word of a functor type* $Y_1 \to (\ldots \to (Y_n \to X)\ldots)$ *combines with strings of argument types* $Y_1, \ldots, Y_n$ *to form a string of type* X*, the functor word is the lexical head (and the argument strings the phrasal dependents).*

In our example, *talked* is now the head and *Bill* and *to Sue* the dependents. We can now derive lexical dependents:

---

[4] → is used in the semantics, and should not be confused with the longer ⟶ used as the rewrite symbol in phrase structure rules.

Definition 6 *A lexical dependent is the lexical head of a phrasal dependent.*

For example, *to* is a lexical dependent of *talked*, because *talked* has a PP argument and *to* has the type PP/NP. This means that *to* is the lexical head of the PP. We now have lexical definitions of head and dependent, and so **AB** grammar can represent dependency relations, albeit in a very indirect manner. It is far more natural to regard it as having *phrasal* heads and dependents, as in definition 4, because it has phrasal functors and arguments.

Incremental Interpretation

Although we can reconstruct dependency relations within **AB** grammar, we cannot develop a flexible notion of constituency which allows the interpretation of these constituents. In *Bill talked to Sue*, we must associate *talked* with *to Sue* before we can associate *talked to Sue* with *Bill*. The only constituents that **AB** grammar will generate are single words plus *to Sue*, *talked to Sue* and *Bill talked to Sue*. We cannot get *Bill talked*, for instance. In fact, **AB** constituency is a superset of full constituency (for instance, it includes *talked to Sue*, which is not a full constituent), but a subset of dependency constituency. For instance, *Bill talked to* is a dependency constituent but not an **AB** constituent.[5] Of course, it is only possible to give a type and an interpretation to a constituent. Therefore *Bill talked to* cannot be interpreted. In general, **AB** grammar does not allow us to give types to dependency constituents and is therefore inadequate as a representational system for incremental interpretation.

3.2.3 The Lambek Calculus

Introduction

Given the definition of X/Y as an X lacking a Y to its immediate right and $Y\backslash X$ as an X lacking a Y to its immediate left, we can see that **AB** grammar does not allow all valid inferences. For instance, if we have an expression that combines with an immediately following Y to form an X, which is immediately followed by an expression that combines with an immediately following Z to form a Y, we can infer that we have an expression that combines with an immediately following Z to form an X. Therefore

---

[5] Note that this is not due to the assumption of VP-ordering.

the inference $X/Y \ Y/Z \Rightarrow X/Z$ is valid in terms of the meanings of the directional slashes, but is not valid in **AB** grammar. The Lambek Calculus **L** (Lambek 1958) is a much more general system and does allow this inference. Indeed it allows all intuitively valid inferences. In this section we shall present an outline of its powers, without going into the details of the mathematics involved. For a much more detailed presentation see Moortgat (1988:ch 1). We shall then discuss the system in relation to dependency, constituency and the representation of incremental interpretation.

Instead of introducing **L** directly, let us consider a small set of rules which are all valid with respect to the meanings of the directional slashes (and therefore **L**-valid):

Forward Application (FA) $X/Y : f \ \ Y : y \Rightarrow X : fy$

Backward Application (BA) $Y : y \ \ Y\backslash X : f \Rightarrow X : fy$

Forward Composition (FC) $X/Y : f \ \ Y/Z : g \Rightarrow X/Z : \lambda v[f(gv)]$

Backward Composition (BC) $Z\backslash Y : g \ \ Y\backslash X : f \Rightarrow Z\backslash X : \lambda v[f(gv)]$

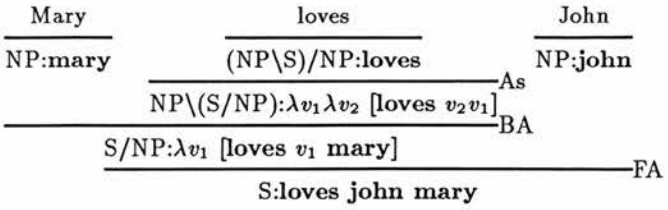
Associativity (As) $(Z\backslash X)/Y : f \Leftrightarrow Z\backslash(X/Y) : \lambda v_1 \lambda v_2[f v_2 v_1]$

Forward Composition[6] has been discussed above. Backward Composition is simply a reversal of the forward form. Associativity is also valid, because an expression looking for a Y to its immediate right to form an expression looking for a Z to its immediate left to form an X is equivalent to an expression looking for a Z to its immediate left to form an expression looking for a Y to its immediate right to form an X. In other words, it does not matter whether the expression to the left or the expression to the right is found first. The first two rules are called *binary* rules because they combine two strings, whereas associativity is a *unary* rule because the type of a single string is changed. Let us consider some derivations using this set of rules, on the sentences *Mary loves John*, *Bill talked to Sue* and *Fred said Bill laughed*. As application is one of our rules, the **AB** derivation of *Mary loves John* given in (3.1) is still valid. However, the following derivation is also valid:

---

[6] *Composition* is a specific operation, and should not be confused with the use of *composition* to refer to the general combination of meanings, which is the sense demanded in the expression *compositional semantics*.

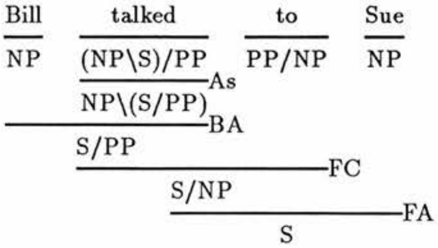
(3.3)



By using the rule of associativity on the verb, we have allowed it to combine with its subject argument before its object argument. There is no change to the meaning ascribed to the sentence, though of course it is built up in a different way. We obtain the same result by two different routes. This extremely important property is known as *derivational equivalence* (or *spurious ambiguity*).

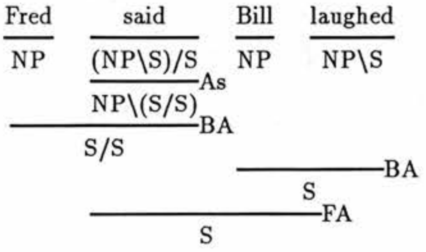
This set of rules allows us to ascribe types and hence meanings to strings such as *Mary loves*, in contrast with **AB** grammar. We can also give types to *loves John*, to each individual word and to the whole sentence. It is therefore possible that this set allows us to ascribe types to all those strings which form dependency constituents. Note also that (3.3) forms the IDR *(Mary)*, *(Mary loves)*, *(Mary loves John)*, and is therefore incrementally valid. Hence it may be that this set of rules allows us to generate the IDR for any sentence. We can, for instance, derive the IDR for *Bill talked to Sue*, which is *(Bill)*, *(Bill talked)*, *(Bill talked to)*, *(Bill talked to Sue)*. The incremental derivation is given below:

(3.4)



This derivation shows that we can give a type (and therefore a meaning) to the string *Bill talked to*, which was not possible in **AB**. Now consider *Fred said Bill laughed*, where *Fred said Bill* is not a dependency constituent. The maximally incremental derivation using this set of rules is given below:

(3.5)



This time we cannot combine *Fred said* and *Bill*, but instead have to wait until we reach *laughed*. So it appears that this set of rules does not allow at least some combinations that violate dependency constituency. Hence the IDR for this sentence, *(Fred)*, *(Fred said)*, *(Fred said)(Bill)*, *(Fred said Bill laughed)* is captured by this set of rules.

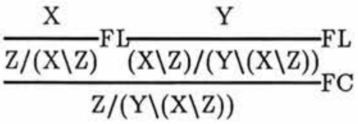
However, the rules of application, composition and associativity, are only a subset of rules valid in terms of the meanings of the slashes. Also valid is *lifting*:

Forward Lifting (FL) $X : x \Rightarrow Y/(X\backslash Y) : \lambda v[vx]$

Backward Lifting (BL) $X : x \Rightarrow (Y/X)\backslash Y : \lambda v[vx]$

The forward version is valid because an expression of type X followed by an expression of type $X\backslash Y$ yields an expression of type Y, and the backward version is valid because an expression of type X preceded by an expression of type Y/X yields an expression of type Y. The problem is that adding this rule makes the calculus structurally complete, in the technical sense that it is possible for any adjacent elements to combine. This is shown in the following proof:

(3.6)



This shows that combinations which are valid in terms of the meanings of the directional slashes are in general possible between words in ways which do not form dependency constituents. The combination of *Fred said* and *Bill* above is a particular example of this. Since these combinations are all **L**-valid, the Lambek Calculus does not directly represent dependency constituency. But because **L** can give a type to any string, we can attempt to define dependency constituency within **L**. This would however not prevent us giving types to strings which were not dependency constituents if we have any need. We shall start by outlining an axiomatization of **L**, which takes us beyond specific rules.

3.2.4 Natural Deduction Axiomatization of L

In this section I shall give an non-technical outline of the natural deduction axiomatization of **L** presented in Morrill, Hepple, Leslie and Barry (1990). The essential fact about this system is that when trying to prove an inference, it is temporarily possible to introduce a new assumption if one of the rules licences the subsequent withdrawal or discharging of the assumption. There are two rules for each of the two connectives / and \. The *elimination rule* corresponds to functional application directly: a proof of type X/Y followed by a proof of type Y, or a proof of type $Y\backslash X$ preceded by a proof of type Y yields a proof of type X. These rules can be written as follows:

(3.7) a. $$\frac{\overset{\vdots}{X/Y} \quad \overset{\vdots}{Y}}{X}/\mathrm{E}$$ b. $$\frac{\overset{\vdots}{Y} \quad \overset{\vdots}{Y\backslash X}}{X}\backslash\mathrm{E}$$

The semantics is simply function application. Hence the subsystem of **L** using only the elimination rules corresponds to the **AB** calculus, with /E being equivalent to forward application and \E being equivalent to backward application.

However there is also a complementary pair of *introduction rules.* The introduction rule for / states that where the rightmost assumption in a proof of the type X is of type Y, that assumption may be withdrawn to give a proof of type X/Y. Similarly, the introduction rule for \ states that where the leftmost assumption in a proof of type X is of type Y, that assumption may be withdrawn to give a proof of type Y\X.[7] These rules can be written as follows:

(3.8) a. $$\frac{\begin{array}{c}[Y]^n\\ \vdots\\ X\end{array}}{X/Y}/\mathrm{I}^n$$ b. $$\frac{\begin{array}{c}[Y]^n\\ \vdots\\ X\end{array}}{Y\backslash X}\backslash\mathrm{I}^n$$

We coindex the assumption and the introduction rule used to discharge the assumption, purely as an aid to perspicuity. Note however that this notation does not embody the conditions that have been stated, namely that, in /I, Y is the rightmost undischarged assumption in the proof of X, and, in \I, Y is the leftmost undischarged assumption in the proof of X.[8]

---

[7] Note that only one assumption can be ever be withdrawn by an introduction rule. In this respect this system differs from the usual natural deduction rule of *conditionalization.*

[8] In addition, the Lambek calculus carries the condition that in both /I and \I the sole assumption

The meaning of the result is given by *functional abstraction* over the meaning of the discharged assumption, which can be represented by a variable of the appropriate type, as defined in chapter 2. In general, when constructing a proof, we make assumptions whenever the elimination rules on their own are not enough to prove what we want, and we discharge the assumptions when they are no longer necessary.

It is straightforward to prove that the four rules, application, composition, associativity and lifting are valid. Application corresponds directly to elimination, as shown in (3.7) above. Proofs of a version of each of the other rules are given below:

Composition:

$$(3.9)\quad \cfrac{\cfrac{X/Y \qquad \cfrac{Y/Z \quad [Z]^1}{Y}/E}{X}/E}{X/Z}/I^1$$

Associativity:

$$(3.10)\quad \cfrac{\cfrac{\cfrac{[Z]^2 \qquad \cfrac{(Z\backslash X)/Y \quad [Y]^1}{Z\backslash X}/E}{X}\backslash E}{X/Y}/I^1}{Z\backslash (X/Y)}/I^2$$

Lifting:

$$(3.11)\quad \cfrac{\cfrac{X \quad [X\backslash Y]^1}{Y}\backslash E}{Y/(X\backslash Y)}/I^1$$

The proofs for composition and associativity involve the assumption of a type that is used as an argument in the derivation. Let us call this abstracting over an argument. The proof for lifting, on the other hand, involves the assumption of a type that is used as a functor in the derivation, which we shall call abstracting over a functor. This distinguishes lifting from all the other rules. In fact it is possible to prove application, composition and associativity by abstracting over a functor, but it is not necessary. Here we show one such proof, for forward application:

---

in a proof cannot be withdrawn, so that no types are assigned to the empty string.

(3.12)
$$\dfrac{X/Y \quad \dfrac{\dfrac{[X/Y]^1 \quad Y}{X}/E}{(X/Y)\backslash X}\backslash I^1}{X}\backslash E$$

This proof is intuitively more complex than the simpler proof for application, which simply involves elimination. On the other hand, there is no proof for lifting that does not involve abstracting over a functor.

Let us return to *Mary loves John*, making reference to the natural deduction rules defining **L** rather than the more limited sets of rules considered above. It is possible to derive this using just the elimination rules, combining *loves* with *John* first.[9] But we can also combine *Mary* with *loves*. We have to make use of an introduction rule, but we do not have to abstract over a functor:

(3.13)
$$\dfrac{\dfrac{\dfrac{\text{Mary}}{NP{:}\mathbf{mary}} \quad \dfrac{\dfrac{\text{loves}}{(NP\backslash S)/NP{:}\mathbf{loves}} \quad [NP{:}x]^1}{NP\backslash S{:}\mathbf{loves}\ x}/E}{S{:}\mathbf{loves}\ x\ \mathbf{mary}}\backslash E}{S/NP{:}\lambda x\,[\mathbf{loves}\ x\ \mathbf{mary}]}/I^1$$

We assumed the existence of an NP to the right of *loves*, in order to allow us to combine *loves* with *Mary*. Therefore we subsequently had to discharge the assumption by using the introduction rule /I.

However, it is also possible to combine *Mary* with *loves* by abstracting over a functor. The semantics is of course the same:

(3.14)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{Mary}}{NP{:}\mathbf{mary}} \quad [NP\backslash S{:}f]^1{:}}{S{:}f\ \mathbf{mary}}\backslash E}{S/(NP\backslash S){:}\lambda f\,[f\ \mathbf{mary}]}/I^1 \quad \dfrac{\dfrac{\text{loves}}{(NP\backslash S)/NP{:}\mathbf{loves}} \quad [NP{:}x]^2}{NP\backslash S{:}\mathbf{loves}\ x}/E}{S{:}\mathbf{loves}\ x\ \mathbf{mary}}\backslash E}{S/NP{:}\lambda x\,[\mathbf{loves}\ x\ \mathbf{mary}]}/I^2$$

The second assumption is used as an argument, like the assumption in (3.13), but the first assumption is used as a functor. Therefore we can perform the derivation by abstracting over a functor, but it is not necessary. The combination of *Mary* and

---

[9] Again assuming 'VP-ordering'

loves can be performed with or without abstracting over a functor. As we saw in the two proofs of application, in (3.7) and (3.12), if there is a proof that does not involve abstracting over a functor, then an equivalent proof with such an abstraction appears more complex than necessary.

Mary loves is a dependency constituent. On the other hand, *Fred said Bill* in *Fred said Bill laughed* is not a dependency constituent. It is not possible to perform the necessary combinations without abstracting over a functor:

(3.15)

$$\dfrac{\dfrac{\dfrac{\text{Fred said}}{\mathrm{S/S}} \qquad \dfrac{\dfrac{\text{Bill}}{\mathrm{NP}} \quad [\mathrm{NP\backslash S}]^1}{\mathrm{S}}\backslash E}{\mathrm{S}}/E}{\mathrm{S/(NP\backslash S)}}/I^1$$

This is therefore comparable with lifting, because it is impossible to derive lifting without abstracting over a functor. It does not involve a particularly large number of operations, and there seems to be no part of the derivation which is redundant. This contrasts with (3.14), where *Mary* undergoes unnecessary forward lifting. The operations that take the type of *Mary* from NP to S/(NP\S) involves a diversion, because in order to do this we assumed NP\S, when in fact we already have NP\S, formed by combining the type of *loves* with the assumed NP, to the immediate right of the type of *Mary*. This diversion is technically known as a *redex*. When this is removed we get the proof in (3.13), known as the *contractum*. A *normal form* proof is one that involves no redexes.[10] (3.13) is a normal form proof, as is (3.15), but (3.14) is not in normal form, because it can be simplified to (3.13).

It looks as though the distinction between assuming a functor and assuming an argument is central to the definition of a dependency constituent within the Lambek Calculus. More specifically, a dependency constituent can only be formed if the normal form proof does not involve abstraction over a functor. We shall consider whether this is the right generalization in the next section.

---

[10] It can be shown that there is always a unique normal form for a proof. See Morrill et al for a detailed exposition.

3.3 A Definition of Dependency within L[11]

We have seen how **AB** can be taken to equate phrasal heads and functors, and phrasal dependents and arguments, but we have also seen how to derive lexical notions of head and dependent. Hence we can show a direct relationship between **AB** grammar and dependency grammar. However, these definitions are of limited use, because we cannot in general give an interpretation to the combination of (adjacent) lexical heads and dependents in **AB**. In other words the dependencies are purely descriptive and do not define possible compositional operations. This is obviously not a problem in **L**, because it possible to give an interpretation to the combination of any adjacent elements. Let us therefore consider how to formulate a definition of dependency within **L**.

3.3.1 A Limitation Regarding the Semantics

The dependency representations considered in chapter 2 are closely related to predicate-argument structure. Hence, for instance, there is no difference between the representation of a quantified definite description and a name; both are simply regarded as NPs. In order to draw detailed parallels between categorial grammar and dependency representations it is clearly necessary to regard quantified and non-quantified expressions to have the same type for syntactic purposes. Indeed this would seem to be apparent given the great similarity between their distributional properties. Syntactically, then, we shall regard both simply as NPs. Therefore, the analyses will be indeterminate with respect to scope, for instance, and in addition it is reasonable to make other type assignments on syntactic grounds without taking semantic considerations into account, as in the following section.

The lambda-expressions which are used below as translations of the expressions of natural language are therefore not disambiguated, but may be associated with several related meanings. Some further process of translation is therefore required, but we shall say nothing about this. A similar point is made by Morrill (1988:161-4), who gives some examples of how a disambiguated meaning representation can be derived from such expressions. This ties in with the fact that we can in some sense 'understand' expressions without resolving all scope ambiguities, for instance:

---

[11] Parts of this section are derived from joint work with Guy Barry, and are related to section 2 of Barry and Pickering (1990) (bound into this thesis).

(3.16) Some boy talks to most girls at every party.

It is difficult to get any reading at all for this sentence, and yet we clearly can give it some kind of interpretation. This indicates that cognitively there is a division between two types of interpretation, and so it appears correct to differentiate two levels of representation. This thesis will assume that semantic considerations should not influence the types given to the expressions combined to form the first level of interpretation. However, no attempt is made to consider how to generate full interpretations, for instance where quantified expressions are involved.

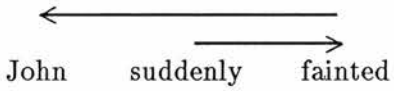
3.3.2 Modifiers

In the discussion of **AB**, we assumed that the relation between lexical types and dependency relations was straightforward. This is of course a simplification, and depends on assumptions about the categorial lexicon and on the direction of dependency. For instance, we assumed that verbs serve as functors, and therefore do not have basic types. In *Mary loves John*, *loves* is assumed to be of type (NP\S)/NP, and *Mary* and *John* of type NP. Therefore the lexical type for *loves* is functional over the lexical types of the two NPs. An alternative is to treat NPs as sentential modifiers taking verbs as their arguments (Karttunen 1986; Whitelock 1988). We could therefore associate functors and dependents, and arguments and heads. We shall not take this position in the present account.

More serious (from the point of view of elegance) is the possibility that some heads act as functors, but others serve as arguments. There is one large problem area, which is the analysis of modifier constructions, such as those involving adjectives, adverbs and relative clauses. The reason for this is semantic. It is assumed that the meaning of an transitive verb, for instance, maps the meaning of an NP to the meaning of a VP, and hence the transitive verb serves as the functor and the NP as the argument. However, the standard assumption is that the meaning of a modifier maps the meaning of an expression of a particular type to the meaning of another expression of the same type. Hence adjectives and relative clauses are given types $N \rightarrow N$ and adverbs are given type $(NP \rightarrow S) \rightarrow (NP \rightarrow S)$ or $S \rightarrow S$. This is semantically natural; in an expression like *fake gun*, for instance, we would assume that the adjective *fake* maps from the set of guns to the set of fake guns, rather than the other way round.

However, dependency theory standardly assumes that modifiers are dependents. Hence there is a class of expressions which appear to be functors and dependents. Hudson (1987) makes the point that most syntactically motivated definitions of head coincide with the notion of semantic functor, but that the modifier-modified relation seems to be the exception to this. The question is whether to accept this inelegant conclusion and to assume that modifiers are heads, or to assume that modifiers are arguments. We shall try to avoid the first possibility. Treating modifiers as heads goes against all standard assumptions about headedness. For instance, X′ syntax (Jackendoff 1977), now standard in constituency theories such as GB and GPSG, assumes that the modifier is not the head. In a rule like VP ⟶ VP AdvP, it is the modified element which is the head, because it has the same category type as the mother. In dependency grammar treating the modifier as the head would make many constructions violate adjacency, such as when the modifier comes between subject and verb:

(3.17)



Regarding modifiers as heads appears very strange at best.

Therefore I shall argue that modifiers should be treated as arguments. As indicated in the previous section, the types assumed here are not totally determined by semantic considerations, so there is no reason to assume that semantics should determine the type of modifiers. I shall present two arguments below that modifiers are arguments in the syntactic sense, and hence that their semantic status as functors is found only at the same level as the level where quantifiers obtain scope.

First, treating modifiers as functors forces a radical dichotomy between complements and modifiers. But in fact there are a large number of borderline cases which suggest that there should only be a gradual distinction. Discussions of the terms (Pollard and Sag 1987:134-139; Matthews 1981:123-141) generally consider a number of criteria which greatly overlap but do not exactly coincide. Modifiers in general must have a 'looser' relationship with the head than complements. There may be collocational restrictions between a head and a complement, not determinable on general semantic grounds. For instance, we *toast bread* but *grill meat*, although the event described by the two verbs does not differ. There cannot be collocational restrictions between verbs and modifiers, say, so that if we can *toast on Friday* then it is equally natural to *grill on Friday.* A complement may be obligatory or will at least have to be recoverable from the context. For instance, we cannot say *I was watching* in isolation. Therefore *watch* subcategorizes for a direct object, which can in some circumstances be latent. Finally, there are restrictions on the number of complements. For instance, some verbs do not take direct objects (e.g. *exist*), and no complement can be added freely. A prototypical modifier will have none of these properties. If the complement-modifier distinction were clear, with all these criteria in agreement, we could reasonably regard modifiers as functors and complements as arguments.

However, there are many borderline cases, such as directional elements. In *I dumped the books on the table* it is unclear whether *on the table* is optional or obligatory: can we say *I dumped the books* in isolation? It is possible to use different prepositions with *dump*, for instance, replacing *on* with *onto*, so at least in this respect there are no collocational restrictions, which is characteristic of modifiers. *Put* requires a preposition (*I put the books* can only be elliptical), and yet has little preference for *on* over *onto.* With *place*, the prepositional phrase is more optional, but if it is used, there is a strong (collocational) preference for *on* rather than *onto.* It can be difficult to show whether prepositional phrases are indirect objects, on the one hand, or modifiers on the other. This is most apparent with the preposition *to*, which can behave like a modifier when it is used directionally, for instance with *carry. I carried the books* is as complete as *I carried the books to Mary*, and *?I carried Mary the books* is odd. On the other hand, many uses of *to* are clearly as complements, for instance with *send.* There seems to be a continuous scale of relatedness between a complement-modifier and a verb, not a sudden distinction.

Second, there are adverbs which are not free in their distribution, and hence are not simple modifiers. The middle construction allows subcategorization for adverbs, so we can say *This book reads well* but not **This book reads.* The dependency grammar analysis treats *well* as a dependent of *reads* in a completely standard way. But in standard categorial grammar, with complements as arguments, we must either introduce a new type for *well* which verbs like *read* subcategorize for, or else assume that these verbs subcategorize for a complex type, S\S or (NP\S)\(NP\S). Either possibility is unparsimonious. In fact there are also a few nouns that subcategorize for adjectives or relative clauses. For instance, we can say *That is a serious matter*, or *That is a matter which I want to clear up*, but not **That is a matter.*

We shall therefore assume that modifiers have atomic types, and that modifiable types are functions over zero or more premodifiers and zero or more postmodifiers. This is similar to the HPSG account (Pollard and Sag 1987). In order to achieve this we shall extend the categorial machinery with a 'Kleene Star' structural operator, so that X^* means 'zero or more occurrences of X'. We shall classify noun premodifiers as PrN, noun postmodifiers as PoN, verb premodifiers as PrV and verb postmodifiers as PoV. Thus lexical nouns will be categorized as (PrN*\N)/PoN*, from which the types N, PrN\N, PrN\(PrN\N), N/PoN etc. are derivable. Similarly intransitive verbs will be categorized as (PrV*\(NP\S))/PoV*, transitive verbs as ((PrV*\(NP\S))/PoV*)/NP, and so on. *Reads* can now be given the type (PrV*\(NP\S)/PoV)/PoV*, but the inner PoV must have a feature restricting it to certain postmodifiers (thus preventing **this book reads on Friday*, for example). We shall usually only write the type that is appropriate to the particular derivation. For example, *man* will be given the type N in *the man walks*, PrN\N in *the old man walks*, N/PoN in *the man who I like walks* and so on. Hence we can keep the association between functors and heads, and arguments and dependents, in the lexical type assignments.

3.3.3 Dependency in the Lambek Calculus

This system now allows a direct correspondence between head and functor, and between dependent and argument. Let us now use these type assignments within the Lambek Calculus. We still have a problem, as although **L** allows incremental interpretation, it lets the roles of functor and argument be reversed. First assume that head-dependent relations are based on the functor-argument relations implicit in lexical type assignments (ignoring whether we are considering phrasal or lexical heads and dependents). Then in *John walks*, the function *walks* of type NP\S is applied to the argument of type NP, and so *walks* is the head and *John* the dependent. This is just as in **AB** grammar. But we can lift the NP to give us the higher type S/(NP\S) for *John*, so now *John* appears to be the head.

Therefore if we wish to retain the functor-head/argument-dependent correspondence, we must disallow any operations that reverse the roles of functor and argument. If we allow lifting in a derivation, for instance in the combination of *John* and *walks*, then the lifted element can get a functional type that it did not have before. We must therefore disallow the operation of lifting. We can contrast this unary rule with the rule of associativity, which does not reverse the roles of functor and argument, but rather operates on a functor to change round the order of combination of the functor with an argument to the left and an argument to the right. We therefore do not want to prohibit associativity.[12] Composition likewise does not reverse functor and argument.

We shall capture this notion in the following definition:

Definition 7 *A derivation in* **L** *is dependency-preserving iff there is a means of performing this derivation in the natural deduction axiomatization that does not assume a type which serves as the functor in an elimination inference ('abstracting over a functor').*

If there is a derivation that does not involve abstraction over a functor, then it will be the normal form derivation. This is therefore equivalent to:

Definition 8 *A derivation in* **L** *is dependency-preserving iff the normal form for this derivation in the natural deduction axiomatization does not assume a type which serves as the functor in an elimination inference.*

Because the assumption has to be discharged before the end of the derivation and because the semantics for the introduction rule is functional abstraction, a dependency-preserving derivation will have a lambda-calculus meaning representation which does not involve abstraction of a variable that occurs as a functor within it. This means that $\lambda f\ [f\ \mathbf{john}]$ and $\lambda f\ [\mathbf{said}\ (f\ \mathbf{bill})]$ do not represent the meanings of dependency-preserving derivations. On the other hand, $\lambda x\ [\mathbf{loves}\ x\ \mathbf{mary}]$ and $\lambda y \lambda x\ [\mathbf{loves}\ xy]$ do represent the meanings of dependency-preserving derivations. But such derivations of course require adjacency, so not all meaning representations that involve no abstraction over a functor will have dependency-preserving derivations (or indeed any derivation at all).

---

[12] Note that a system allowing associativity pays no attention to 'VP-ordering' since, for instance, NP\(S/NP) and (NP\S)/NP are an associative pair.

These definitions therefore disallow the unary operation of lifting, but allow the operation of associativity, and in general they allow us to retain the functor-argument relations implicit in lexical type assignments, and hence correspondence with head-dependent relations. In other words, assuming correspondence, it stops the head-dependent relation being reversed. But it has the further important consequence of preventing adjacent but unrelated words from being treated as functor and argument, and hence in a head-dependent relationship. An example of this is the combination of *Fred said* and *Bill* in *Fred said Bill laughed*, which is only possible by abstracting over a functor, as in (3.15). So it restricts methods of combination to those that respect both dependency and direction of dependency.

Let us now give some example derivations. As before, we assume determiners to be heads of nouns, and also make the standard semantically-inspired categorial grammar assumption that determiners are functors over nouns. This therefore preserves the correspondence relation. Consider the six normal form derivations below. The three in (3.18) do not assume a type that serves as the functor in an elimination inference, but the three in (3.19) do make this assumption. Likewise, only the last three involve abstraction over a variable that occurs as a functor in the meaning representation. Let us introduce the convention of typing the variables in the meaning representation, and use f to refer to all variables that serve as functors in the meaning representations:

(3.18) a.

$$\dfrac{\dfrac{\text{the}}{\mathrm{NP/N}} \quad \dfrac{\text{dog}}{\mathrm{N}}}{\mathrm{NP}}\,/\mathrm{E}$$

the dog

b.

$$\dfrac{\dfrac{\dfrac{\text{Mary}}{\mathrm{NP}} \quad \dfrac{\dfrac{\text{loves}}{\mathrm{(NP\backslash S)/NP}} \quad [\mathrm{NP}]^1}{\mathrm{NP\backslash S}}\,/\mathrm{E}}{\mathrm{S}}\,\backslash\mathrm{E}}{\mathrm{S/NP}}\,/\mathrm{I}^1$$

λx^{NP}[loves x mary]

c.

$$\dfrac{\dfrac{\dfrac{\text{will}}{\mathrm{(NP\backslash S)/(NP\backslash S)}} \quad \dfrac{\dfrac{\text{see}}{\mathrm{(NP\backslash S)/NP}} \quad [\mathrm{NP}]^1}{\mathrm{NP\backslash S}}\,/\mathrm{E}}{\mathrm{NP\backslash S}}\,/\mathrm{E}}{\mathrm{(NP\backslash S)/NP}}\,/\mathrm{I}^1$$

λx^{NP}[will (see x)]

(3.19) a.

$$\dfrac{\dfrac{\dfrac{[\mathrm{NP/N}]^1 \quad \dfrac{\text{dog}}{\mathrm{N}}}{\mathrm{NP}}{/\mathrm{E}} \quad \dfrac{\text{runs}}{\mathrm{NP\backslash S}}}{\mathrm{S}}{\backslash\mathrm{E}}}{\mathrm{(NP/N)\backslash S}}{\backslash\mathrm{I}^1}$$

$$\lambda f^{\mathrm{N}\rightarrow\mathrm{NP}}[\mathbf{runs}\ (f\ \mathbf{dog})]$$

b.

$$\dfrac{\dfrac{\dfrac{\text{said}}{\mathrm{S/S}} \quad \dfrac{\dfrac{\text{Bill}}{\mathrm{NP}} \quad [\mathrm{NP\backslash S}]^1}{\mathrm{S}}{\backslash\mathrm{E}}}{\mathrm{S}}{/\mathrm{E}}}{\mathrm{S/(NP\backslash S)}}{/\mathrm{I}^1}$$

$$\lambda f^{\mathrm{NP}\rightarrow\mathrm{S}}[\mathbf{said}\ (f\ \mathbf{bill})]$$

c.

$$\dfrac{\dfrac{\dfrac{[\mathrm{((NP\backslash S)/NP)/NP}]^1 \quad \dfrac{\text{Mary}}{\mathrm{NP}}}{\mathrm{(NP\backslash S)/NP}}{/\mathrm{E}} \quad \dfrac{\text{John}}{\mathrm{NP}}}{\mathrm{NP\backslash S}}{/\mathrm{E}}}{\mathrm{(((NP\backslash S)/NP)/NP)\backslash(NP\backslash S)}}{\backslash\mathrm{I}^1}$$

$$\lambda f^{\mathrm{NP}\rightarrow(\mathrm{NP}\rightarrow(\mathrm{NP}\rightarrow\mathrm{S}))}[f\ \mathbf{mary}\ \mathbf{john}]$$

Notice that (3.18b) and (3.19b) could be part of derivations for *Mary loves John* and *Fred said Bill laughed*, where, as we have indicated, *Mary loves* is a dependency constituent, but *said Bill* is not. It is worth emphasising at this point that the meaning representation of a dependency-preserving derivation cannot involve the abstraction of a variable that serves as a functor within it. This is not the same as a claim that such a meaning representation cannot involve a variable of functional type; what is crucial is that it does not *serve* as a functor in the derivation. Consider the derivation of *John will* below:

(3.20)

$$\dfrac{\dfrac{\dfrac{\text{John}}{\mathrm{NP}} \quad \dfrac{\dfrac{\text{will}}{\mathrm{(NP\backslash S)/(NP\backslash S)}} \quad [\mathrm{NP\backslash S}]^1}{\mathrm{NP\backslash S}}{/\mathrm{E}}}{\mathrm{S}}{/\mathrm{E}}}{\mathrm{S/(NP\backslash S)}}{/\mathrm{I}^1}$$

This derivation assumes a functional type, but the type is not abstracted over. The associated meaning representation is $\lambda x^{\mathrm{NP}\rightarrow\mathrm{S}}[\mathbf{will}\ (\mathbf{see}\ x)]$. The variable is of functional type, but it is not a functor in the representation.

3.3.4 Dependency Constituency in the Lambek Calculus

In chapter 2, we defined a dependency constituent as a string in which every word except the root has a head within the string. We now wish to redefine dependency constituent in terms of the derivation involved. A dependency constituent is a string for which there is a derivation that involves only dependency-preserving operations. More formally:

Definition 9 *A dependency constituent of type X is a string (under a particular reading) whose normal form derivation in* **L** *is dependency-preserving.*

Notice that we define dependency constituent relative to a particular type. The importance of this will be seen in the discussion of coordination below. We have shown that the underlined substrings in (3.21) below all have normal form derivations which do not involve the assumption of a functor type:

(3.21) a. The dog runs.

b. Mary loves John.

c. John will see Mary.

On the other hand, the normal form derivations for the underlined substrings in (3.22) do assume a functor:

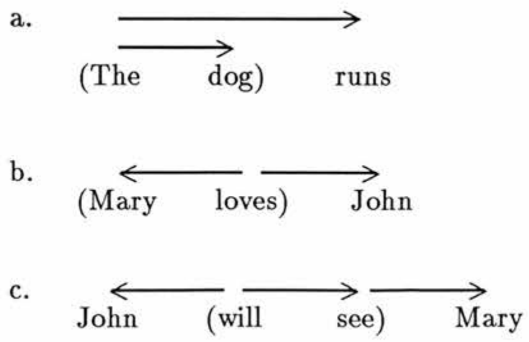
(3.22) a. The dog runs.

b. Fred said Bill laughed.

c. I showed Mary John.

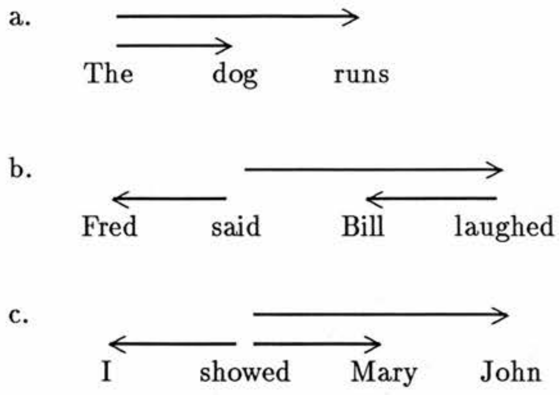
Hence the underlined strings in (3.21) are dependency constituents, but those in (3.22) are not. This is also found when we draw the patterns of dependency associations for these strings. For the examples in (3.21), we indicate the relevant dependency constituents, in which every word except the root must have a head within the string (by definition 1):

(3.23)



But we can see that the relevant strings in (3.22) are not dependency constituents:

(3.24)



We shall use the natural deduction definition of dependency constituency. A proof that the two definitions of dependency constituent are equivalent is beyond the scope of this thesis.

3.3.5 Implications of the Choice of Head

There are remaining problems about whether a particular string is a dependency constituent or not, because there may be disputes about which word is the head and which the dependent. However these are purely questions about lexical assignments. We shall assume that determiners are functors, as argued above. More importantly, we have argued that we should treat modifiers as arguments, not functors. Let us show how this affects which strings are regarded as dependency constituents. We shall claim that these choices make intuitive sense, but shall argue in the next section that assuming the modifier to be the argument is linguistically useful. Consider for example:

(3.25) the tall man

(3.26) the man who saw me

If modifiers are regarded as functors, the underlined substring in (3.25) will be a dependency constituent, but the one in (3.26) will not be. On the other hand, if modified elements are regarded as functors, the underlined substring in (3.25) will not be a dependency constituent, but the one in (3.26) will be. The relevant proofs and lambda-terms follow, with the a. examples treating modifiers as functors and the b. examples treating modified elements as functors:

(3.27) a.

$$\dfrac{\dfrac{\dfrac{\textbf{the}}{\mathrm{NP/N}} \qquad \dfrac{\dfrac{\textbf{tall}}{\mathrm{N/N}} \quad [\mathrm{N}]^1}{\mathrm{N}}/\mathrm{E}}{\mathrm{NP}}/\mathrm{E}}{\mathrm{NP/N}}/\mathrm{I}^1$$

$\lambda x^{\mathrm{N}}[\textbf{the}\ (\textbf{tall}\ x)]$

b.

$$\dfrac{\dfrac{\dfrac{\textbf{the}}{\mathrm{NP/N}} \qquad \dfrac{\dfrac{\textbf{tall}}{\mathrm{PrN}} \quad [\mathrm{PrN\backslash N}]^1}{\mathrm{N}}\backslash\mathrm{E}}{\mathrm{NP}}/\mathrm{E}}{\mathrm{NP/(PrN\backslash N)}}/\mathrm{I}^1$$

$\lambda f^{\mathrm{PrN \rightarrow N}}[\textbf{the}\ (f\ \textbf{tall})]$

(3.28) a.

$$\dfrac{\dfrac{\dfrac{\textbf{the}}{\mathrm{NP/N}} \qquad \dfrac{\dfrac{\textbf{man}}{\mathrm{N}} \quad [\mathrm{N\backslash N}]^1}{\mathrm{N}}\backslash\mathrm{E}}{\mathrm{NP}}/\mathrm{E}}{\mathrm{NP/(N\backslash N)}}/\mathrm{I}^1$$

$\lambda f^{\mathrm{N \rightarrow N}}[\textbf{the}\ (f\ \textbf{man})]$

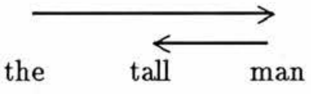
b.

$$\dfrac{\dfrac{\dfrac{\textbf{the}}{\mathrm{NP/N}} \qquad \dfrac{\dfrac{\textbf{man}}{\mathrm{N/PoN}} \quad [\mathrm{PoN}]^1}{\mathrm{N}}/\mathrm{E}}{\mathrm{NP}}/\mathrm{E}}{\mathrm{NP/PoN}}/\mathrm{I}^1$$

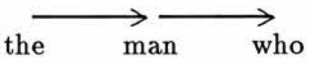
$\lambda x^{\mathrm{PoN}}[\textbf{the}\ (\textbf{man}\ x)]$

There is a clear intuitive connection between *the* and *man*, but not between *the* and *tall*, and between *man* and *who*, but not between *the* and *who*. This therefore supports the choice of modifier as argument. These can be associated with the dependency representations in (3.29) and (3.30) below rather than those in (3.31) and (3.32):

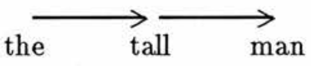
(3.29)



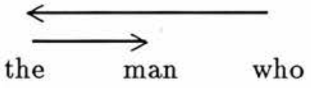
(3.30)



(3.31)



(3.32)



It is worth noting here that the string underlined in (3.33) cannot be a dependency constituent, whatever the choice of head in modifier constructions:

(3.33) John loves $\underline{\text{Mary madly}}$.

Again, the derivation in a. below treats the modifier as functor, while that in b. treats the modified element as functor:

(3.34) a.

$$\dfrac{\dfrac{\dfrac{[(\mathrm{NP}\backslash \mathrm{S})/\mathrm{NP}]^1 \quad \dfrac{\textbf{Mary}}{\mathrm{NP}}}{\mathrm{NP}\backslash \mathrm{S}}/\mathrm{E} \quad \dfrac{\textbf{madly}}{(\mathrm{NP}\backslash \mathrm{S})\backslash(\mathrm{NP}\backslash \mathrm{S})}}{\mathrm{NP}\backslash \mathrm{S}}\backslash \mathrm{E}}{((\mathrm{NP}\backslash \mathrm{S})/\mathrm{NP})\backslash(\mathrm{NP}\backslash \mathrm{S})}\backslash \mathrm{I}^1$$

$$\lambda f^{\mathrm{NP}\rightarrow(\mathrm{NP}\rightarrow \mathrm{S})}[\textbf{madly}\ (f\ \textbf{mary})]$$

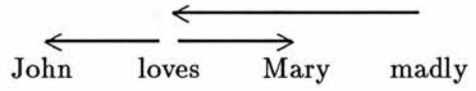
b.

$$\dfrac{\dfrac{\dfrac{[((\mathrm{NP}\backslash \mathrm{S})/\mathrm{PoV})/\mathrm{NP}]^1 \quad \dfrac{\textbf{Mary}}{\mathrm{NP}}}{(\mathrm{NP}\backslash \mathrm{S})/\mathrm{PoV}}/\mathrm{E} \quad \dfrac{\textbf{madly}}{\mathrm{PoV}}}{\mathrm{NP}\backslash \mathrm{S}}/\mathrm{E}}{(((\mathrm{NP}\backslash \mathrm{S})/\mathrm{PoV})/\mathrm{NP})\backslash(\mathrm{NP}\backslash \mathrm{S})}\backslash \mathrm{I}^1$$

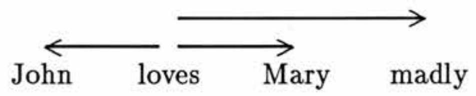
$$\lambda f^{\mathrm{NP}\rightarrow(\mathrm{PoV}\rightarrow(\mathrm{NP}\rightarrow \mathrm{S}))}[(f\ \textbf{mary})\ \textbf{madly}]$$

There are two dependency representations, but because they only differ on the direction of the relationship between *loves* and *madly*, *Mary madly* is not a dependency constituent in either case:

(3.35)



(3.36)



3.4 Coordination[13]

3.4.1 Flexibility and Constituency

I showed in chapter 2 that coordination cannot be used in any straightforward way as evidence for standard phrase structure constituents. For instance, *Mary loves* is a conjunct but not a constituent in *Mary loves and Sue adores John.* A grammar using traditional phrase structure rules and no additional mechanism such as transformations or empty categories will have a rule schema like $X \longrightarrow X$ conj X, where X has to be

---

[13] Apart from the discussion of (3.47), this section is derived directly from section 3 of Barry and Pickering (1990) (bound into this thesis).

a constituent. Since *Mary loves* is not a constituent, this cannot be applied, and so the grammar will not generate this sentence.

Rather than adding an additional mechanism, we can allow this coordination directly if we have a grammar that regards *Mary loves* as a constituent. This is given independent justification because we have already claimed that it is necessary to regard *Mary loves* as a constituent in incremental interpretation. An obvious claim is that coordination is closely related to dependency formation and dependency constituency. Let us consider the syntactic description of coordinate constructions from the point of view of the Lambek Calculus and dependency constituency. It should be made clear that no attempt will be given to provide a mechanism for coordination. The purpose of this section is to describe what strings can be coordinated, and to show how to capture the regularities in terms of the present framework.

3.4.2 Coordination of Dependency Constituents

A claim often made in favour of flexible categorial grammars is their ability to deal with a large proportion of coordination phenomena by means of a single principle, usually assumed to be as follows:

Hypothesis 2 *Two (or more) strings may be coordinated to give a string of type X iff each has type X.*

For the purposes of this discussion we shall treat the two directions of implication in hypothesis 2 as two separate hypotheses. One direction may be phrased as follows:

Hypothesis 3 *If two (or more) strings can be given the same type X, then they can be coordinated to give a string of type X.*

We shall discuss the converse later (hypothesis 5).

The precise mechanism used to implement hypothesis 3 varies from formulation to formulation (e.g. polymorphic types for conjunctions, syncategorematic rule schemas), but the principle remains the same. To save space in examples, instead of giving full derivations we shall usually merely bracket each conjunct, and specify the type that must be assigned to each conjunct (and to the coordinate structure) for the derivation to proceed.

If we are working within the full Lambek calculus, hypothesis 3 holds true in a large class of cases. For instance, we can clearly coordinate strings that form standard phrase structure constituents, like (3.37) below:

(3.37) John [sang some songs] and [played the piano].
Type of each conjunct: NP\S

We can also deal with cases like (3.38) and (3.39), both of which are usually classified under the heading of 'non-constituent' coordination:

(3.38) John [will buy] and [may eat] the beans.
Type of each conjunct: (NP\S)/NP

(3.39) John loves [Mary madly] and [Sue passionately].
Type of each conjunct: (((NP\S)/PoV*)/NP)\(NP\S)

In our terms, (3.37) and (3.38) both involve coordination of dependency constituents, while (3.39) involves coordination of non-dependency constituents. Although these three examples look very different, in each case the structure of each conjunct is the same, in the sense that each consists of a sequence of words of the same lexical types. Hypothesis 3 will always allow coordination in such cases, since **L** can always give the same type to two strings with the same structure (in this sense).

More interesting are cases where the conjuncts have different structures, such as (3.40) and (3.41):

(3.40) John [plays the piano] and [sings].
Type of each conjunct: NP\S

(3.41) John [bought] and [may eat] the beans.
Type of each conjunct: (NP\S)/NP

However, the unacceptability of (3.42) shows that hypothesis 3 overgenerates:[14]

(3.42) *John loves [Mary madly] and [Sue].
Type of each conjunct: (((NP\S)/PoV*)/NP)\(NP\S)

---

[14] The intended reading of (3.42) is that John loves Mary madly and John loves Sue. This should not be confused with the orthographically similar elliptical construction, with the meaning that John loves Mary madly and John loves Sue madly, which would have a marked intonational break before *and*.

In (3.40) and (3.41) both conjuncts are again dependency constituents, but in (3.42) the first conjunct is not a dependency constituent, and the second conjunct is not a dependency constituent of this type, but is only a dependency constituent of type NP. Note that even if we took modifiers to be functors we could still give the conjuncts in (3.42) the shared type ((NP\S)/NP)\(NP\S), and *Mary madly* would still not be a dependency constituent, nor *Sue* a dependency constituent of this type.

In general, the restrictions on coordination of unlike structures seem to be stronger than those on coordination of like structures. For example, (3.43) appears to be acceptable, but (3.44) does not:

(3.43) [I believe that John] and [Harry thinks that Mary] is a genius.
Type of each conjunct: S/(NP\S)

(3.44) *[I believe that John] and [Mary] is a genius.
Type of each conjunct: S/(NP\S)

Note once more that both conjuncts in (3.43) and the first conjunct is (3.44) are not dependency constituents. Similarly, (3.45) seems acceptable, but not (3.46):

(3.45) [two small] and [three large] oranges
Type of each conjunct: NP/(PrN*\N)

(3.46) *[two small] and [three] oranges
Type of each conjunct: NP/(PrN*\N)

Here, the conjuncts in (3.45) and the first conjunct in (3.46) are not dependency constituents if modifiers are treated as arguments, although they would be if modifiers were treated as functors. This fact that the coordination in (3.46) is impossible gives linguistic support to the treatment of modifiers as arguments.

This evidence suggests that there is some correlation between coordinability and dependency constituency. In the examples where both conjuncts are dependency constituents, namely (3.37), (3.38), (3.40) and (3.41), coordination is always possible. This suggests that hypothesis 3 might be replaced by the following weaker claim:

Hypothesis 4 *If two (or more) dependency constituents have the same type X, then they can be coordinated to give a string of type X.*[15]

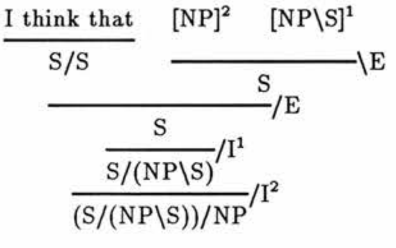
But in the examples where at least one conjunct is not a dependency constituent, coordination appears to be restricted to cases like (3.39), (3.43) and (3.45) where the conjuncts are structurally similar; it is not possible for the other examples (3.42), (3.44) and (3.46). In order to discriminate between these cases we need a more precise notion of 'like structure', which we shall discuss in the next subsection.

Note that dependency constituents are defined with respect to a particular type. This rules out examples like (3.47), where both conjuncts are dependency constituents, and can be given equivalent types, but only by non-dependency preserving derivations:

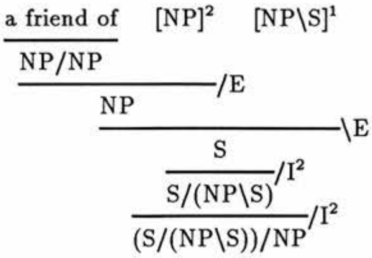
(3.47) *[I think that] and [a friend of] Mary left.

The two conjuncts appear syntactically unrelated and yet they both form a sentence when placed before *Mary left.* *I think that* is a dependency constituent of type S/S, and *a friend of* is a dependency constituent of type NP/NP. They can both be given the same raised type, S/(NP\S)/NP, but only by non-dependency-preserving operations:

(3.48)



(3.49)



Hence *I think that* is a dependency constituent of type S/S, and *a friend of* is a dependency constituent of type NP/NP, but neither is a dependency constituent of type

---

[15] However we have no explanation for (i), which would be allowed under hypothesis 4:

(i) ?*I believe [that] and [that Harry thinks that] Mary is a genius.

(S/(NP\S))/NP. Therefore the coordination is not licenced by hypothesis 4. Notice that the dependency pattern is strange, with *that* being the head of *left*, and *of* the head of *Mary*.

So far we have not examined the converse claim to hypothesis 3:

Hypothesis 5 *If two (or more) strings can be coordinated to give a string of type X, then each can be given type X.*

In order to maintain this hypothesis we must have some mechanism for dealing with well-known cases of 'unlike category coordination' (Sag, Gazdar, Wasow and Weisler 1985) such as (3.50):

(3.50) John is [lucky] and [a rogue].

Whatever the type of *lucky and a rogue*, it must be assignable to both *lucky* and *a rogue*. If we say that these two items have only the lexical types PrN and NP respectively, and that *is* is lexically ambiguous between (NP\S)/PrN and (NP\S)/NP, then we are forced to conclude that no type is assignable to *lucky and a rogue*. But if we give *lucky* and *a rogue* an additional shared lexical type, PredP say, and give *is* the single lexical type (NP\S)/PredP, then assigning the type PredP to *lucky and a rogue* is consistent with hypothesis 5. Alternatively we might extend **L** with boolean operators, as suggested in Morrill (1990), to achieve a similar effect.

From hypotheses 4 and 5, the following hypothesis follows:

Hypothesis 6 *Two (or more) dependency constituents can be coordinated to give a string of type X iff each can be given type X by dependency-preserving operations.*

3.4.3 Coordination of Non-Dependency Constituents

Hypothesis 4 puts no restrictions on the internal structure of the two dependency constituents that are coordinated, as is suggested by (3.37) and (3.40), or (3.38) and (3.41). But even when the conjuncts themselves are not dependency constituents, there appears to be no restriction on the internal structure of any substring that is a dependency constituent. For example, (3.51) and (3.52) are both as good as (3.39):

(3.51) John loves [Mary madly] and [the young woman passionately].

(3.52) John loves [Mary madly] and [Sue with great ardour].

This would appear to suggest a form of 'constituentwise' coordination, in which the conjuncts are matched maximal dependency constituent for maximal dependency constituent. The division into MDCs of *John bought* is ⟨*John bought*⟩; the division into MDCs of *the young woman passionately* is ⟨*the young woman, passionately*⟩. Let us make the following definition:

Definition 10 *A set of strings* $\alpha, \beta, \ldots$*, each with n MDCs* $\langle\alpha_1, \ldots, \alpha_n\rangle$*,* $\langle\beta_1, \ldots, \beta_n\rangle$*,* $\ldots$*, are parallel if for each i* $\alpha_i, \beta_i, \ldots$ *may be given the same type by dependency-preserving operations.*

A similar notion might be defined by including the product connective (Wood 1988; Bouma 1989).

Given this definition, we may make the following claim, which subsumes hypothesis 4:

Hypothesis 7 *Two (or more) parallel strings of type* X *may be coordinated to give a string of type* X*.*

Hypothesis 7 claims that any two (or more) strings of type X which each have n MDCs of types $\langle X_1, \ldots, X_n\rangle$ can be coordinated to give a string of type X. This covers a large class of examples of classical 'non-constituent coordination' not covered by hypothesis 4, including (3.39), (3.43), (3.45), (3.51) and (3.52) (repeated below as (3.53) to (3.57)):[16]

(3.53) John loves [Mary madly] and [Sue passionately].
Type of each conjunct: (((NP\S)/PoV*)/NP)\(NP\S)
Types of MDCs of each conjunct: ⟨NP, PoV⟩

(3.54) [I believe that John] and [Harry thinks that Mary] is a genius.
Type of each conjunct: S/(NP\S)
Types of MDCs of each conjunct: ⟨S/S, NP⟩

---

[16] It must be noted that hypothesis 7 overgenerates in some cases, e.g.:

(i) *John told [Mary Bill] and [Fred Sue] was coming.
Type of each conjunct: ((((NP\S)/S)/NP)\(NP\S))/(NP\S)
Types of MDCs of each conjunct: ⟨NP, NP⟩

This example involves two functor abstractions rather than one in the formation of each conjunct, and so the association between the MDCs is in some sense weaker than in previous examples. However, some similar constructions can be acceptable, such as the following:

(ii) John told [the mothers that their daughters] and [the fathers that their sons] were all at the party.

(3.55) [two small] and [three large] oranges
Type of each conjunct: NP/(PrN*\N)
Types of MDCs of each conjunct: ⟨NP/N, PrN⟩

(3.56) John loves [Mary madly] and [the young woman passionately].
Type of each conjunct: (((NP\S)/PoV*)/NP)\(NP\S)
Types of MDCs of each conjunct: ⟨NP, PoV⟩

(3.57) John loves [Mary madly] and [Sue with great ardour].
Type of each conjunct: (((NP\S)/PoV*)/NP)\(NP\S)
Types of MDCs of each conjunct: ⟨NP, PoV⟩

It also predicts, given some mechanism for dealing with constructions such as (3.50), that (3.58) will be grammatical:

(3.58) John is [pious in church] and [a rogue in the tavern].

The converse of hypothesis 7 (which subsumes hypothesis 5) also appears to be true:

Hypothesis 8 *If two (or more) strings can be coordinated to give a string of type X, then they are parallel strings of type X.*

For example, in none of (3.42), (3.44) and (3.46) (repeated below as (3.59) to (3.61)) are the intended conjuncts parallel, since in each case they contain different numbers of MDCs, and indeed coordination is not possible in any of them:

(3.59) *John loves [Mary madly] and [Sue].
Type of each conjunct: (((NP\S)/PoV*)/NP)\(NP\S)
Types of MDCs of first conjunct: ⟨NP, PoV⟩
Types of MDCs of second conjunct: ⟨NP⟩

(3.60) *[I believe that John] and [Mary] is a genius.
Type of each conjunct: S/(NP\S)
Types of MDCs of first conjunct: ⟨S/S, NP⟩
Types of MDCs of second conjunct: ⟨NP⟩

(3.61) *[two small] and [three] oranges
Type of each conjunct: NP/(PrN*\N)
Types of MDCs of first conjunct: ⟨NP/N, PrN⟩
Types of MDCs of second conjunct: ⟨NP/N⟩

Note that if we took modifiers to be heads both *two small* and *three* would be analysed as dependency constituents, and hence (3.61) would be incorrectly predicted to be grammatical. Hence the claim that modified elements should be taken as functors is supported by the coordination evidence, given the general account of coordination.

The present account also rules out examples such as the following two:

(3.62) ?I talked [about cricket to Edna] and [to Eric about chess].

(3.63) ?*I gave [John a pencil] and [a pen to Mary].

It might be possible to generate (3.62) by some relaxation of the definition of parallelism, but there is no obvious general way to generate (3.63).

To summarize, we may combine hypotheses 7 and 8 into the following principle:

Hypothesis 9 *Two (or more) strings may be coordinated to give a string of type X iff each has type X and they are parallel.*

It should be clear that this principle subsumes hypothesis 6, because hypothesis 6 considers the special case of hypothesis 9 when there is only one MDC in each conjunct (that is, where $n = 1$ in definition 10).

3.4.4 Conclusions

Barry and Pickering (1990) have shown that coordinable strings can be defined in a very straightforward manner in terms of dependency constituents, in a way that is not possible using standard phrase structure grammar. The Lambek Calculus in itself is too permissive, because the only principled restriction is to allow any strings that can be given equivalent types to coordinate, which overgenerates. As mentioned above, this conclusion is avoided by permitting coordination between only a subset of strings defined in terms of dependency constituents. Therefore we have a definition of constituent, wider than that used in phrase structure grammar but narrower than that implicated by simple Lambek Calculus. But we retain a means of giving types to strings which are not constituents. We now have stronger evidence for the importance of dependency constituents and dependency-preserving operations.

3.5 Incremental Interpretation, Categorial Grammar and Dependency

3.5.1 How much Incrementality?

In the last chapter we claimed that sentence processing involved more incrementality than would be permitted by the simplest application of standard phrase structure grammar, and claimed that the human sentence processor was able to interpret all dependency constituents immediately. This is equivalent to the claim that we perform at least all **L**-valid combinations (under a particular reading) that do not involve abstracting over a functor.[17]

The most straightforward hypothesis was the DIH, which states that all and only dependency constituents are interpreted. Therefore this claim holds that we perform all and only **L**-valid combinations (under a particular reading) that do not involve abstracting over a functor. The only evidence produced for the DIH was that nested constructions like (2.20), repeated below as (3.64), are hard to process, and that the simplest way to represent this is to suggest that MDCs are kept semantically distinct:

(3.64) I saw the rat that the cat that the dog chased bit.

This entails that we have to store information about the dependency constituent or constituents that we are not actively considering. For instance in *Fred said Bill laughed*, after *Bill* there are two dependency constituents, *Fred said* and *Bill*. We have to remember the syntactic type for *Fred said*, not merely the meaning, because the directionality information is crucial: *Fred said* is looking for an S to its right. It is possible that the store is organized as a stack (c.f. Ades and Steedman 1982) if more than one dependency constituent has to be held, but other arrangements are possible. But if on the

---

[17] This is not intended to imply that the human processor actually uses natural deduction. However a description in these terms may be equivalent to the processing method, whatever that method is.

other hand the restriction on abstracting over functors is not heeded by the sentence processor, we can simply parse in a structurally complete manner (as proposed by the SCIH), and therefore have no need to store 'inactive' dependency constituents.

The coordination evidence makes it appear very likely that dependency constituents do play a role in sentence processing. We have shown that it is possible to interpret, for instance, a verb and one of its arguments before the other argument is encountered, so that *Mary loves* can be interpreted before *John* is reached, and we also know that *Mary loves* can be a conjunct in 'right-node-raised' examples. However, we have seen that there are restrictions on the coordinability of non-dependency constituents. If the SCIH were correct, then we would expect to be able to perform all **L**-valid inferences, and would therefore expect to be able to perform coordinations between any strings that could be given equivalent types, which has been shown incorrect. In order to restrict coordination, we would have to 'tag' types that were constructed by using non-dependency preserving inferences, so that they were in fact not equivalent to supposedly equivalent types at a finer level of analysis. For instance, in **John loves Mary madly and Sue*, both conjuncts would be given the type (((NP\S)/PoV*)/NP)\(NP\S), but they would be differentiated by some minor feature which prevented coordination. These features would have to reproduce the division into maximal dependency constituents.[18] This seems unnecessarily complicated, and it is much more straightforward to claim that the proscription of abstracting over a functor has directly psychologically reality.

However, there is an obvious difficulty with this account. We know that non-dependency constituents may sometimes be coordinated when there is parallelism between the maximal dependency constituents of the conjuncts. We can assess the grammaticality of such sentences without constructing non-dependency constituents, by imposing the restriction of parallelism between the MDCs. But interpretation and type assignment occur together in a theory that assumes the rule-to-rule hypothesis. Therefore, if we do not give a type to non-dependency constituents, how can we ever understand a sentences which involve the coordination of these strings?

Notice that I have made no attempt to consider the semantics of coordination, or

---

[18] An equivalent claim would be that such strings are assigned two types. This would mean that combinations between words could both take place and not take place, which is bizarre. It is important to keep this distinct from the discussion in section 3.3 above about the possibility that for some semantic purposes it may be useful to consider, for instance, a quantified NP as having a higher type than the type which is used syntactically.

even the syntactic operations involved in the association of each conjunct with the conjunction. It seems likely that coordination between dependency constituents takes place by combining the two conjuncts into a single representation which can then be combined with the rest of the sentence in the same way that a non-conjoined expression would. But this is not necessary. In the coordination of non-dependency constituents, it is perhaps less likely. For instance, *Mary madly* and *Sue passionately* in *John loves Mary madly and Sue passionately* may never be given unified meanings, that can be represented as λf [(f **mary**) **madly**] and λf [(f **sue**) **passionately**]. This would mean that at least those conjuncts which are not dependency constituents are associated with the rest of the sentence on their own, without being integrated into an interpreted coordinate structure. Let us consider two possible arguments for this.

Consider (3.65) below:

(3.65) I believe that Harry and Fred thinks that Tom loves/*love someone.

The most important point is that the shared subordinate verb *loves* has to be singular, which of course contrasts with other coordinations:

(3.66) John and Bill love/*loves someone.

This suggests that there is a level of representation at which two sentences *I believe that Harry loves someone* and *Fred thinks that Tom loves someone* are constructed. This is directly reminiscent of the coordinate structure transformation of early transformational grammar, and therefore predicts a singular verb.[19] Note that a sentence like (3.67) below has a singular verb as well, presumably because parentheticals are not integrated with non-parenthetical expressions:

(3.67) John, but not Bill, loves/*love someone.

---

[19] It is interesting that there is a semantic sense in which the embedded verb could be plural, but this seems to be overriden:
(i) I believe that Harry and Fred thinks that Tom is a fool/*are fools.
However, judgements may be somewhat strained here, presumably because it is claimed (by two people) that there are in fact two fools, and each of them may believe in the existence of only one fool.

It is of course possible that we do interpret both parenthetical coordinations and non-dependency coordinations, but then for some reason do not treat the coordinate construction as plural for the purposes of agreement. But the obvious interpretation is that a representation of the coordinate structure is not constructed, and hence that there is no need to type or interpret non-dependency preserving conjuncts.

Another point is that it is at least extremely hard and perhaps impossible for *someone* in (3.65) to take wide scope, and therefore to give the sentence the meaning that there is a particular person who I believe that Harry loves and Fred thinks that Tom loves (except of course 'by chance'). This again suggests that the two conjuncts must be individually integrated with the rest of the sentence, not conjoined together first. This contrasts with (3.66) where giving *someone* wide scope is possible, and certainly preferred, and in coordinations between dependency constituents, both readings are possible:

(3.68) John likes and Mary loves someone.

This evidence may not be totally clear, but gives support to the agreement facts. So there seems good evidence that the conjuncts in (3.65) are never interpreted as a unit. This may well be signalled by the extremely marked intonation given to the sentence (which may be closely related to parenthetical intonation). Note that the process of interpretation may be very different for *John loves Mary madly and Sue passionately.* First, the intonation of the conjuncts is not marked. Second, there is no difference in readings between (3.69) and (3.70). Giving *someone* wide scope is vastly preferred and perhaps necessary in both examples:

(3.69) Someone loves Mary madly and Sue passionately.

(3.70) Someone loves Mary and Sue.

However the purpose of this discussion was not to account for the processing of coordinate constructions. Rather I have intended to show that the fact that such coordinations can occur in particular circumstances does not indicate that the processor must be able to make non-dependency preserving combinations. Therefore there is no need to give the conjuncts in non-dependency preserving coordinations non-dependency preserving types. The DIH can therefore be retained. This concludes the discussion of coordination (except for some incidental remarks in the next chapter). There will be no attempt to construct a formal rule for coordination.

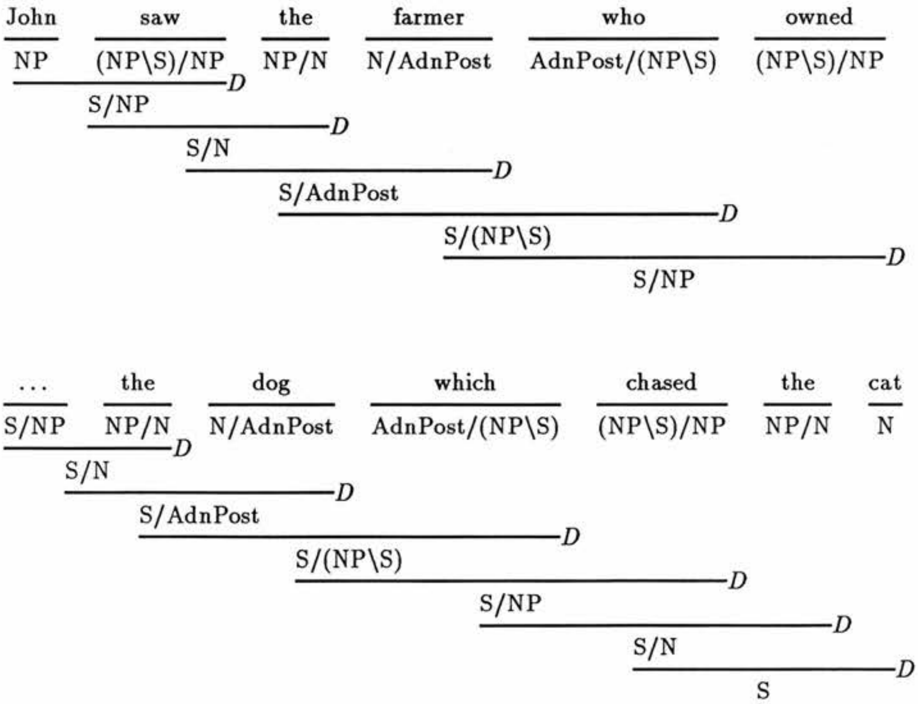
3.5.2 Explaining Processing Complexity

Let us consider a 'right-branching' sentence that we have shown presents such a problem for the simplest account of sentence processing based on phrase-structure grammar:

(3.71) John saw the farmer who owned the dog which chased the cat.

The left-associative parse for this sentence is in fact wholly dependency preserving, and is therefore in keeping with the DIH. For instance, *John*, *John saw*, *John saw the*, *John saw the farmer*, *John saw the farmer who* are all dependency constituents, and therefore the IDR consists of only single MDCs. Here we give the left-associative parse, and mark each combination with D, meaning dependency-preserving combination. We shall anticipate the analysis of relative clauses considered in the next chapter:

(3.72)



Like all 'right-branching' sentences, (3.71) retains an even intonation, as well as being easy to process. The simplicity of the operations of association seem to be reflected in the ease we have in processing it.

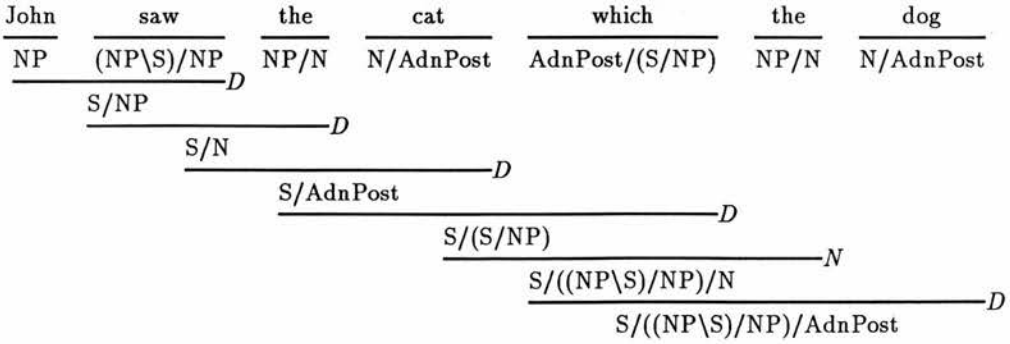
However, as we have shown, we can process sentences like *Fred said Bill laughed*, where the IDR includes *(Fred said) (Bill)* and therefore does not solely consist of single MDCs. Assuming the DIH, we have to remember the meanings of each MDC discretely. But there is a limit to the number of MDCs we can remember:

(3.73) John saw the cat which the dog which the farmer owned chased.

Such constructions are uncontroversially hard to process, as we have noted. This is in sharp contrast to (3.71), even though they are 'grammatically' of similar complexity. Unlike (3.71), (3.73) does not have a left-associative derivation that only forms dependency constituents. In (3.73), the initial substring *the cat which* forms a dependency constituent, but not *the cat which the*, so that *the dog which* must be parsed separately from *the cat which.* Similarly *the farmer* must be parsed separately from *the cat which* and *the dog which.* At the point before *owned*, then, we find that we have three unrelated dependency constituents, *the cat which*, *the dog which* and *the farmer.* Intonationally as well, we can distinguish three units, which are similar but not identical to the dependency constituents (the relative pronouns associate with the next dependency constituents). Remembering these three dependency constituents must take us beyond memory resources. On the other hand, there is no difficulty in processing *Fred said Bill laughed*, where we have to remember two dependency constituents.

Let us first consider the left-associative parse of this nested construction that is implicated by the SCIH, rather than the DIH. We notate non-dependency preserving combinations by N:

(3.74)



...	which	the	farmer	owned	chased
S/((NP\S)/NP)/AdnPost	AdnPost/(S/NP)	NP/N	N	(NP\S)/NP	(NP\S)/NP

S/((NP\S)/NP)/(S/NP) D

S/((NP\S)/NP)/((NP\S)/NP)/N N

S/((NP\S)/NP)/((NP\S)/NP) D

S/((NP\S)/NP) D

S D

We can see the complexity of the syntactic types after these combinations; there is a strong contrast with the simple types found in the derivation of (3.71) above. The lambda-terms become similarly complex. For instance, the lambda-term for *John saw the cat which* is

$$\lambda f^{\mathrm{NP}\to\mathrm{S}}[\mathbf{saw}\ (\mathbf{the}\ (\mathbf{cat}\ (\mathbf{which}\ f)))\ \mathbf{john}],$$

but the term for *John saw the cat which the* is

$$\lambda y^{\mathrm{N}}\lambda g^{\mathrm{NP}\to(\mathrm{NP}\to\mathrm{S})}[\mathbf{saw}\ (\mathbf{the}\ (\mathbf{cat}\ (\mathbf{which}\ (\lambda x^{\mathrm{NP}}[g\ x\ (\mathbf{the}\ y)]))))\ \mathbf{john}].$$

In the second lambda-term, the abstracted variable g acts as a functor, unlike f in the first lambda-term.[20]

In contrast, consider the parse supported by the DIH, which is the maximally incremental derivation that contains only dependency combinations:

(3.75)

John	saw	the	cat	which
NP	(NP\S)/NP	NP/N	N/AdnPost	AdnPost/(S/NP)

S/NP D

S/N D

S/AdnPost D

S/(S/NP) D

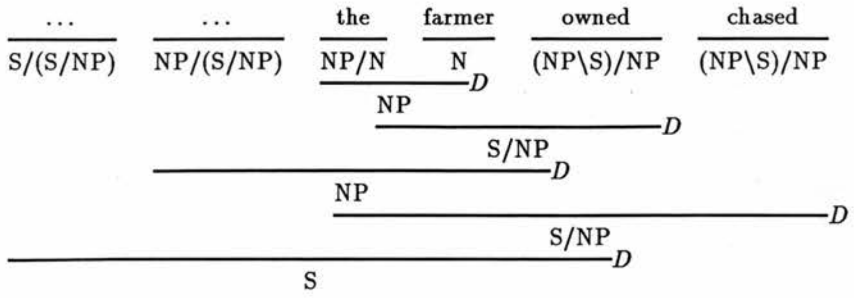
the	dog	which
NP/N	N/AdnPost	AdnPost/(S/NP)

NP/AdnPost D

NP/(S/NP) D

---

[20] Notice that f is not a functor in the meaning representation, although it has the type NP → S. It is the *role* of the variable that is relevant, not the *type*. This has been pointed out above.



As we can see, the nested construction only requires us to remember three MDCs, and it is already almost impossible to interpret. Although some sentences of this complexity may be somewhat less problematic, adding another level of embedding makes such sentences far worse:

(3.76) John saw the cat which the dog which the farmer who the child liked owned chased.

In contrast, a multiple subject relative can be extended by recursing the construction without ill effect:

(3.77) John saw the farmer who owned the dog which chased the cat which followed the mouse which nibbled the cheese.

Again we can parse this sentence without having to make use of a store or make non-dependency combinations. Indeed, we can extend this sentence as long as we like and still experience no processing difficulty, and the sentence retains its even intonation. Each additional use of the construction must not therefore add to processing complexity.[21] Indeed, this is what we see in the category assigned at various points in the incremental parse. For instance in (3.71) the category of (i) *John saw the farmer who owned* is the same as the category of (ii) *John saw the farmer who owned the dog*

---

[21] Note that it is not the number of dependency constituents that have to be remembered through the whole of the parse that is relevant to the processing difficulty, but rather how many have to be remembered at any given time. For instance, the IDR of the sentence (i) below involves many cases where two dependency constituents have to be remembered together:
(i) John said Mary believed Bill knew Sue escaped.
The IDR is *(John)*, *(John said)*, *(John said)(Mary)*, *(John said Mary believed)*, *(John said Mary believed)(Bill)*, *(John said Mary believed Bill knew)*, *(John said Mary believed Bill knew)(Sue)*, *(John said Mary believed Bill knew Sue escaped)*. Therefore, two dependency constituents repeatedly have to be remembered, but never any more than this.

which chased, namely S/NP. It is also the same as the category of (iii) *John saw the farmer who owned the dog which chased the cat which followed* in (3.77) and so on.

There is a general principle here. Any theory of parsing which forced us to regard (ii) as more complex than (i), or (iii) more complex than (ii), would be psychologically inadequate. For instance, if we could not combine a relative pronoun with its antecedent immediately, but instead had to wait until the end of the sentence, we would have to remember any number (depending on the length of the sentence) of non-maximal dependency constituents without being able to form the MDC. At some point this would extend beyond our processing abilities, *which must be finite*, however large they are. In fact the evidence from processing nested constructions suggests that they are very limited indeed. Hence we have a general test for a psychologically realistic theory of parsing. If the theory requires that the recursion of a construction will force us to store more elements every time the construction is recursed, then the sentence must eventually become impossible to process. Hence a construction, like the one repeatedly used in (3.71) and (3.77), that gives a processable sentence however many times it is used, cannot require a parse that involves the steady accretion of unintegrated material. We shall term this the *Minimal Incrementality Condition* or *MIC.* It rules out parsing systems based on many conceivable categorial grammars (such as **AB**) and all phrase structure grammars, under the condition which only allows constituents to be interpreted. It also imposes a restriction on the form of any parser using a grammar where this restriction is not obeyed.

Chapter 4

Unbounded Dependencies

4.1 Introduction

This chapter considers the linguistic treatment and psychological processing of unbounded dependency constructions. I shall assume that this term at least refers to sentences like:

(4.1) Who does Mary love?

(4.2) Bill saw the man who Mary thinks Sue loves.

(4.3) Those cakes, John thinks Fred believes Bill ate.

All these sentences contain a verb, *love*, *admires* and *ate* respectively, which does not have one of its arguments in proximity to it. This is in contrast to most other constructions using these verbs; for instance, we could say *Bill ate those cakes*, where *those cakes* directly follows *ate*. Because of this unusual fact about unbounded dependency constructions, linguistic and psycholinguistic accounts have often given them non-standard treatments.

In this chapter, I shall first contrast the approaches taken to these constructions within constituency theory with the approaches used in dependency grammar and particularly categorial grammar. I shall discuss some of the mechanisms that constituency theories have introduced, and shall make the general claim that these are unparsimonious when compared to the categorial representation of dependency grammar introduced in the last chapter. This treatment of unbounded dependencies is then contrasted with those suggested in other versions of flexible categorial grammar, specifically those advocated by Steedman and Dowty, and some problems with these accounts are indicated. The chapter then considers the relationship between grammatical treatments of unbounded dependencies and sentence processing. Again, any attempt by constituency theory to allow incremental processing appears unparsimonious, but one class of constituency theories, those based on GPSG, can be equated with flexible categorial grammar when considered incrementally. The use of empty categories in particular becomes pointless in an incrementally plausible treatment. Finally, I shall discuss some aspects of the relevant experimental literature, and shall challenge the standard assumption that there is overwhelming evidence for a processor that uses empty categories. An important conclusion is that processing is possible without a gap-filling mechanism.

4.2 Constituency Theory

Under this heading I shall consider theories that are either strongly or loosely based on phrase structure grammar. Transformational grammar and GPSG, for instance, are strongly based on phrase structure rules, whereas GB has a much more oblique relation to any such rules, because they do not have a privileged status and are a consequence of other principles of grammar. However, hierarchical phrase structural representations are still central to GB, as they are used in the formulation of several central principles, including, for instance, c-command. At the risk of some simplification I shall regard GB as essentially a phrase-structural theory.

Assuming that multidominance and crossing branches are disallowed, there is no way to represent unbounded dependencies in general in a simple phrase structure tree. For instance, in (4.2) above, we cannot indicate that *the man* is related to both *saw* and *loves*. To show the latter point, we need crossing branches, to show both together we need a multidominant structure. Such patterns cannot be generated by context free rules. Even if we are only concerned with weak generative capacity, and assuming that all the symbols used in the rules must represent traditional constituent categories, we can invent a new set of rules to deal with *who Mary loves*, but when we add a layer of embedding (*Mary thinks* in (4.2)), we have to formulate yet another set of rules to deal with this construction. Because it is possible to add an arbitrary number of layers of embedding, we would need an infinite number of rules. The grammar then ceases to be context-free (and is greatly complicated). Hence constituency theory has to add some additional mechanism. Let us consider two approaches, that used by transformational grammar and its descendents, and that used in GPSG. We shall assume that these approaches are representative of possible treatments of unbounded dependencies within constituency theory.

4.2.1 Transformations and Traces

Traditionally, transformational grammar has a rather limited view of the power of context-free phrase structure rules. Therefore the grammar is augmented with transformational rules and associated mechanisms. They operate on the phrase structure trees to produce new phrase structure trees. This means that there are (at least) two levels of syntactic structure, one before and one after the transformations are applied. They serve to show that pairs of sentences with no obvious surface relation can have properties in common that cannot be represented by phrase structure rules. An example of this is the relation between passives and actives.

But Bresnan (1978) has shown that there is no need to assume an additional level of representation to deal with passives, because it is possible to describe their relation with a lexical rule instead. This allows the constituent structure of passives to be represented in a standard phrase structure tree using only a single level of representation. However, transformational grammar and its descendants generally still make use of transformations to analyse passives, for essentially theory-internal reasons. But unbounded dependencies have a more fundamental need for transformations, because it is not possible to generate them at all using standard phrase structure rules, nor to represent them in standard phrase structure trees. Hence the descendents of transformational grammar are very strongly committed to transformational treatments of unbounded dependencies.

The traditional transformational grammar known as Standard Theory (Chomsky 1965) assumes that the topicalized sentence *John, Mary loves* has a base-generated Deep Structure form something like *Mary loves John.* A transformational rule then moves *John* to the beginning of the sentence. Therefore *John* is located at Deep Structure at the position which allows it to be treated as the direct object of the verb *loves.*

This is important because transformational theories define grammatical relations configurationally, as we have already noted. If Deep Structure did not exist, there would be no reason to regard *John* as having any relation to *loves* at all. In this sentence, the Deep Structure form is closely related to a grammatical sentence, but this is by no means necessary. For instance, there is no grammatical sentence related to the Deep Structure of (4.2) above, which has *who* located after *loves*.

In Standard Theory, the actual sentence *John, Mary loves* is represented at another level called Surface Structure. This view, however, changed with the advent of what is known as trace theory (Fiengo 1977). This assumes that all levels of syntax must have certain fundamentals in common (a criterion known as the projection principle). Thus if a word is base-generated at one location and is subsequently moved, it is necessary to leave a phonologically null element, a trace, at this location at the level of representation generated after movement. This level of representation is renamed S-Structure or SS. It does not represent what we actually hear or read, because another level called Phonological Form or PF intervenes, but we can ignore this distinction. Likewise, Deep Structure is renamed D-Structure or DS. Trace theory is assumed in Government-Binding Theory.

John, Mary loves is given the SS representation below (with irrelevant details suppressed):

(4.4) $[\text{John}]_1$, Mary loves $[\text{t}]_1$.

The trace, represented as *t*, is positioned at the DS location of *John*. The relationship between the trace and *John* is represented at SS by coindexation. A similar treatment is given to (4.2)[1]:

(4.5) Bill saw the man $[\text{who}]_1$ Mary thinks Sue loves $[\text{t}]_1$.

GB assumes that the movement of elements between nodes causes these nodes to be coindexed (and hence associated) at S-structure, even though they are not coindexed at D-structure. Therefore the construction of S-structure involves both the positioning of the trace at the vacated node and the formation of an association between the trace and the moved element.

---

[1] In fact, GB assumes that the trace is also represented at an intermediate position in this sentence, but this need not concern us here.

In summary, there is a level of syntactic representation, close to the actual form we hear or read, which contains traces which are coindexed with moved elements. In more general terms, we can say that certain constructions have non-canonical forms, and that these are related to canonical forms (which need not themselves be grammatical) by the assumption that there are extraction sites for the moved items at their canonical locations and that there are associations between the extracted elements and the extraction sites.

Let us make some general points about this. First, GB assumes that there are a set of related empty categories used in the SS representations of particular constructions, whose properties differ in specific respects. We have only considered the one used in unbounded dependencies, known as *wh*-trace. GB also assumes another trace, called NP-trace, which is assumed to occur in constructions like raising and passives. Consider a passive sentence like *Bill was loved*, which is given the representation below:

(4.6) $[\text{Bill}]_1$ was loved $[\text{t}]_1$.

As in traditional transformational grammar, the passive form is assumed to be transformed from a representation related to the active (while the active itself is regarded as canonical). In the active, *loved* has an argument directly following it, so in the passive, the subject is assumed to have moved from object position in a similar way to *John* in *John, Mary loves*, and hence there is also a process of coindexation, represented by the subscript. NP-trace and *wh*-trace are the only traces assumed by GB, but not the only phonologically null elements, or *empty categories*. There are also two others, called PRO and pro, which are base-generated in their SS locations (so that no movement is required). GB has developed a rich inventory of their respective characteristics, which are related to the characteristics of overt NPs.

This association with a trace can be regarded as an anaphoric property, because the properties of the empty category are dependent on an element that precedes (or occasionally follows) it. We cannot discover the properties of an empty element until we consider its antecedent. Thus there is a parallel with the way that overt anaphoric elements get their properties. Unfortunately, at this point there is a major terminological confusion. GB refers to only a subset of anaphoric elements as anaphors, and another subset as pronouns. For example, reflexive pronouns are anaphors and non-reflexive pronouns are pronouns. I shall use both *anaphor* and *pronoun* in their standard senses, where a pronoun is a type of anaphor. In the same way we associate traces and their antecedents, using coindexation, we can associate anaphors with their antecedents:

(4.7) Mary thinks that $[\text{Sue}]_1$ admired $[\text{herself}]_1$.

The representation shows that *Sue* and *herself* are related, and so the reflexive pronoun refers to Sue, not Mary. The association between traces and their antecedents is similar. In (4.4), the coindexation tells us that the trace is related to *John* rather than *Mary*, and so the trace refers to John, not Mary. (Note that GB assumes that the associations in these two sentences are due to separate components of the theory, which impose different restrictions. The association in (4.4) is consistent with Principle C of Binding Theory, the association in (4.7) with Principle A. See Chomsky (1981).)

Therefore GB assumes very deep parallels between empty categories and anaphoric elements and the procedures for determining their antecedents are closely related (we shall not concern ourselves with the specific comparisons). Hence unbounded dependencies are not treated in isolation from all other constructions, which would be unparsimonious.[2] But the theory does assume a fundamental division between non-canonical constructions, that make use of empty categories, and canonical constructions, that do not. It also assumes a close association between non-canonical constructions and anaphora. But GB makes very considerable assumptions about the apparatus we need. We have indicated that we do not have to to treat passives with transformations. Now let us show why they can be dispensed with in the analysis of unbounded dependencies as well.

4.2.2 Slash Categories

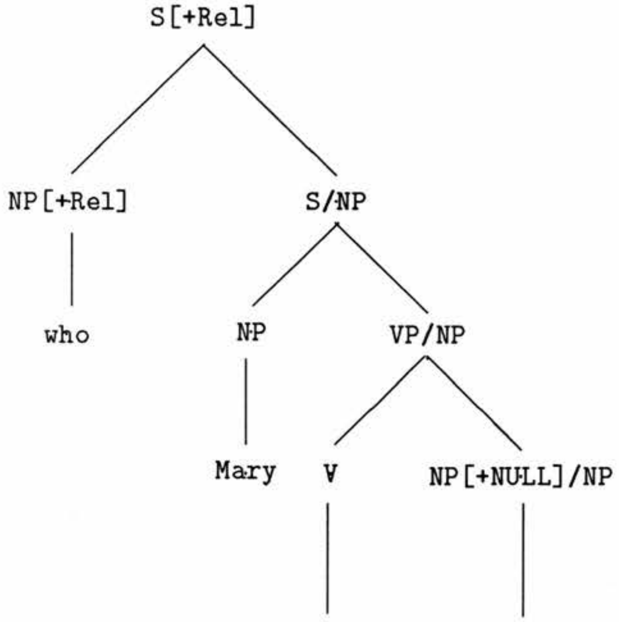
It is impossible to write a context-free phrase structure grammar with traditional constituent categories that will generate all unbounded dependencies (and not wildly overgenerate). But Gazdar (1981; 1982) claimed that it was possible to achieve the correct generative power with context-free rules if complex categories were used. He introduced a slash notation (highly influenced by (non-directional) categorial grammar but different in formulation), where X/Y is interpreted as an X missing a Y. So in addition to the

---

[2] Such a position was assumed in early LFG (Kaplan and Bresnan 1982).

rule S ⟶ NP VP, we have a derived rule S/NP ⟶ NP VP/NP, where the left hand side means an S missing an NP and the right hand side means an NP followed by a VP missing an NP.

Let us consider the later formulation in Gazdar, Klein, Pullum and Sag (GKPS) (1985), where the slash is interpreted as a feature which takes a category as its value (for instance VP/NP is shorthand for VP[Slash: NP], which is analogous to VP[Number: Singular]), and this feature is bound by the restriction that it must occur jointly on the mother and the head daughter. An unbounded dependency is introduced with the general rule S ⟶ X^2, H/X^2, which simply says that a sentence can contain an X^2 (i.e. an XP) and a sentence which contains a null X^2 constituent (H is the head, so here H is S).[3] Consider the relative clause *who Sue loves.* We need the rule S[+Relative] ⟶ NP[+Relative] S/NP. We then have the further rule S/NP ⟶ NP VP/NP, which passes the slash feature onto the head daughter, and then we make use of *Slash Termination Metarule 1*, which says that if X ⟶ W, X^2 is valid then X ⟶ W, X^2[+NULL] is valid. This allows the rule VP/NP ⟶ V NP[+NULL]/NP, and NP[+NULL]/NP can be rewritten as the empty string. This gives us the analysis below:



The same method of analysis can be used in indefinitely extended constructions. In *the man who Mary thinks Sue loves*, we use the additional rule VP/NP ⟶ NP S/NP,

---

[3] The comma indicates that the order of the two daughters has been factored out.

which is derived from VP ⟶ NP S. There is no change to the treatment at the 'top' or the 'bottom' of the tree, in particular, at the 'bottom', STM1 is again used.

GPSG treats unbounded dependencies as dependency relations, of an unusual kind, rather than as anaphoric. The GKPS treatment is context-free, because there are only a finite number of derived categories. However, Maling and Zaenen (1982) point out that in Scandinavian languages, there are cases of multiple extractions which require more than one slash category. Assuming that there is no principled limit to the number of possible extractions, a general schema is necessary for the generation of derived categories because they cannot be finitely listed. Hence there are an infinite number of rules, and so the grammar cannot be context free. But, Shieber (1985) and Culy (1985) show that there are constructions in particular languages which cannot be described in context-free terms by any grammar. Therefore we cannot criticise GPSG simply because any extension that will deal with certain constructions is not context-free. So the slash-category mechanism does allow unbounded dependencies to be treated within an (extended) phrase structure grammar, though not strictly a context-free one. Hence it is not necessary to assume that unbounded dependencies involve anaphoric processes.

GKPS does use traces, but they can be regarded as epiphenomenal. For instance, it has the rules VP/NP ⟶ V NP/NP, NP/NP ⟶ trace, but it is straightforward to replace these with the single rule VP/NP ⟶ V (see Morrill 1988). A VP lacking an NP is simply rewritten as a verb. It is also (as we shall see in chapter 7 below) roughly what is done in the treatment of extractions of embedded subjects. This reformulation makes GPSG a theory that lacks the concept of an extraction site. This fact that this simple change does away with traces makes it hard to see how it is possible to view the GPSG treatment of unbounded dependencies as involving a coindexation relation between the trace and the extracted element, as J.D.Fodor (1989) appears to do, although this may be a reasonable interpretation of, say, Gazdar (1982). We shall return to the question of traces in GPSG later, when considering incremental interpretation.

4.3 Dependency Theory and Categorial Grammar

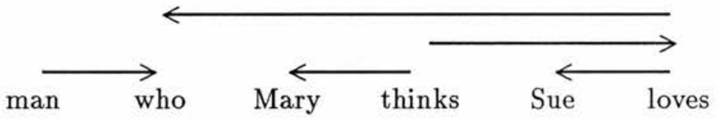
4.3.1 Dependency Grammar

The intuition behind dependency grammar is that it associates words that covary. But there has not been much attempt to make dependency grammar generative (apart from Word Grammar (Hudson 1984)), so it is difficult to assess the constraints on the construction of dependency relations. Given descriptive accounts of many constructions, we simply do not know how to treat 'unusual' constructions like unbounded dependencies. Return to (4.2), repeated below:

(4.8) Bill saw the man who Mary thinks Sue loves.

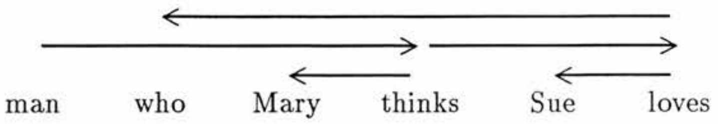
Like constituency theory, dependency grammar faces the problem that *who* appears to have two heads, *man* and *loves*, and again there are a number of solutions. We can straightforwardly give it a polycephalic structure, as below:

(4.9)



Not only does *who* now have two heads, but also *thinks* has no head, even though it is inside a relative clause. This means that the words in the sentence do not hang together as one unit. In addition, *loves* and *who* are now separated by superordinate material, which violates adjacency. This structure appears bizarre even in forms of dependency grammar that do not have a general proscription of polycephalicity. Hudson (1984), for instance, allows *happy* to be in a dependency relation with *John* in *John is happy*, with the superordinate *is* between them, only because *is* serves as the head of both *John* and *happy*. No such condition is fulfilled here, so this analysis violates even his weaker restrictions. But other possible polycephalic representations are no more plausible. For instance we could deny the association of *man* and *who* and make *man* the head of *thinks*:

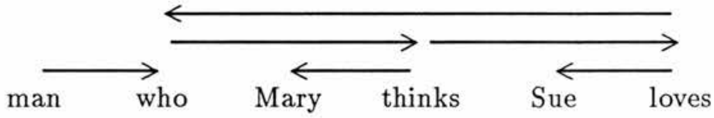
(4.10)



The dependency pattern is very odd, given that *who* modifies *man*, and again the structure violates adjacency.

It is not clear what constitutes a possible analysis of unbounded dependencies in dependency grammar. Arguments by analogy break down because of the unusual structure of unbounded dependencies. Hudson (1984) allows polycephalicity, and would analyse (4.8) by assuming that *who* serves as the head of *thinks* by virtue of a 'visitor' relationship, that *thinks* is the head of *loves*, but also that *loves* is the head of *who*:

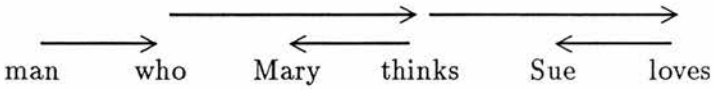
(4.11)



The problem is that it is very difficult to assess how this is motivated. We can regard polycephalicity as the dependency grammar analogue of empty categories or slash categories, in that it is used in a relatively small range of constructions and seems to capture roughly the same notion of non-canonicality. It is unclear that it is really parsimonious, since it causes an increase in generative capacity.

The dependency theory proposed in this thesis assumes that dependencies correspond to compositional-semantic operations, and require a monocephalic representation. Furthermore, every word except one must be the argument to exactly one functor, and so in dependency terms every word except the root of the sentence must have exactly one head. There is now only one reasonable representation, with *who* the head of *think* and *think* the head of *loves*. *Loves* itself lacks an object dependent:

(4.12)



It is clear that *thinks* cannot be the root to the sentence, as it is in a subordinate clause. Therefore its only possible head is *who*. Note that there is no dependency between *who* and *loves* in this representation. This is a possible analysis even though *loves* and *who* covary, because not all syntactic or semantic links are treated as dependencies. The relationship is therefore similar to that between *Bill* and *happy* in *Bill is happy*, which covary but are not in a dependency relation. More of a problem perhaps is the link between *who* and *thinks*, which seems unnecessary except to provide a chain of dependencies. We do not have any principled reason to make this analysis, except for the rather negative reason that *thinks* must have a head. It also means that *who Mary thinks* is a dependency constituent, even though *who* and *thinks* do not covary. The analysis obeys adjacency and monocephalicity, but it is clearly imperfect, and its flaws show the ultimate weakness of using this style of representation. Let us instead look at categorial grammar, considering specifically how to generalize the theory outlined in the previous chapter.

4.3.2 Flexible Categorial Grammar

Introduction

AB grammar has the same problem with unbounded dependency constructions as phrase structure grammar. Given that we can only combine adjacent elements, and that the lexical entry for a transitive verb forces us to combine with the object before the subject, there is nothing to be done when, as in *John, Mary loves*, the subject is adjacent to the verb but the object is not.[4]

In flexible categorial grammar it is generally assumed that the extracted element has to be given a special lexical entry with a type that allows it to serve as the functor over the constituent from which the extraction has taken place (Ades and Steedman 1982). For instance, the topicalized *John* in *John, Mary loves*, is given the type S/(S/NP).[5] *Mary loves* then serves as the argument, and is given type S/NP. It is of course the formation of this type which is impossible in **AB**. A relative pronoun like *who* is standardly given a type (N\N)/(S/NP) if the right-peripheral argument of a relative clause has been extracted, and (N\N)/(NP\S) if the left-peripheral argument has been extracted. Because the relative clause is a modifier to a noun, let us instead give it the type PoN/(S/NP) or PoN/(NP\S), with the modified noun being of category N/PoN (assuming no other modifiers). This allows the noun to serve as functor. Here is a derivation for *the man who Sue loves* (we do not represent the introduction and elimination rules used in each combination, but simple register the result):

---

[4] Even if the type for *loves* were taken to be NP\(S/NP), there would still be no treatment of *John, Mary thinks that Sue loves.*

[5] We may make this a more specialized type St/(S/NP), so that it generates a topicalized sentence of type St.

(4.13)
$$\dfrac{\dfrac{\text{the}}{\mathrm{NP/N}} \quad \dfrac{\dfrac{\text{man}}{\mathrm{N/PoN}} \quad \dfrac{\dfrac{\text{who}}{\mathrm{PoN/(S/NP)}} \quad \dfrac{\text{Sue loves}}{\mathrm{S/NP}}}{\mathrm{PoN}}}{\mathrm{N}}}{\mathrm{NP}}$$

L is unable to deal with non-peripheral extraction in any principled way, as found in *who I believe loved Mary* and *which I put on the shelf.* We shall return to this in chapter 7. However it has no problems with unbounded peripheral extraction, as for instance found in (4.2): *Mary thinks*, of category S/S, can combine with *Sue loves*, and then this can combine as a whole with the relative pronoun:

(4.14)
$$\dfrac{\dfrac{\text{who}}{\mathrm{PoN/(S/NP)}} \quad \dfrac{\dfrac{\text{Mary thinks}}{\mathrm{S/S}} \quad \dfrac{\text{Sue loves}}{\mathrm{S/NP}}}{\mathrm{S/NP}}}{\mathrm{PoN}}$$

L and Dependencies

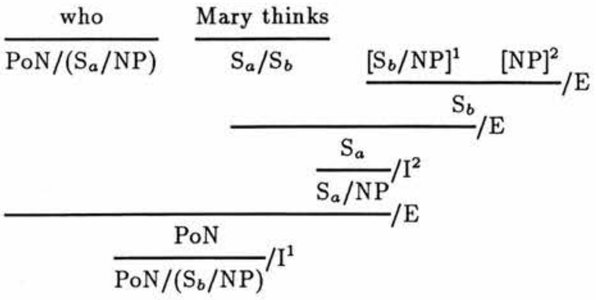
The relative pronoun serves as the functor over the whole of the rest of the relative clause. Hence, given the definitions above, the relative pronoun is both the phrasal and lexical head, and the body of the relative clause the phrasal dependent. These characterizations remain valid in terms of the present treatment of dependency in the Lambek Calculus (for instance, because lifting is forbidden). But determining the lexical dependency, and therefore the dependency structure, is a problem, as was suggested by the above discussion of dependency grammar, because *thinks* fails to qualify as the lexical head of the body of the relative clause.

First, the string *who Mary thinks* is not a dependency constituent, because we have to assume a functor. We distinguish the two Ss by alphabetic subscripting:[6]

---

[6] Note that it would make no difference if we gave *who* the type (N\N)/(S/NP).

(4.15)



On the other hand, *Mary thinks Sue loves* clearly is a dependency constituent, because we can combine S/S and S/NP by composition. But it seems that we are unable to justify the representation given in (4.12), where *thinks* is the head of the body of the relative.

Note that although *Mary thinks Sue loves* can be shown to be a dependency constituent, it has no lexical head (by definition 5). If *thinks*, of type (NP\S)/S, were the head, then there would have to be phrasal dependents of types NP and S, but in fact there is no dependent of type S, rather one of S/NP. Hence, *thinks* is not the head of the body of the relative, and so cannot be the lexical dependent of *who*, because a lexical dependent is the lexical head of a phrasal dependent (by definition 6). Of course, *loves* cannot be the head either, because it could only be made a functor by a non-dependency-preserving type transition. This dependency constituent is of course the phrasal dependent of *who*, but we cannot determine a lexical head for it.

This discussion shows that our formalization does not allow a simple dependency structure to be generated for every construction. It seems better to acknowledge the ultimate inadequacy of the simple dependency grammar representation. This is not a problem if dependency constituency, not dependency representation, is basic. But dependency relations are an easily-comprehensible approximation to the categorial analysis and to dependency constituency, and are a very useful description of most construction types. Therefore they are worth retaining for some purposes. For unbounded dependency constructions, we shall make no attempt to form a dependency representation. However, the dependency constituents tell us valuable information, for instance that there is no direct association between *who* and *thinks* in *who Mary thinks Sue loves.* It is only this whole string which is a dependency constituent. We shall indicate the point at which a dependency constituent is formed with the extracted element by numerical coindexation, giving us the representation *[who]*$_1$ *Mary thinks Sue [loves]*$_1$ in this case. We shall refer to the verb *loves* as the *governor* of the extracted element, and we can loosely describe this as an unbounded dependency.

Comparisons

Flexible categorial grammar treatments of unbounded dependencies can be distinguished from both constituency and dependency accounts because they are able to represent unbounded dependencies without having to postulate some apparatus that is not used for other constructions. Phrase structure grammar has traditionally only used standard constituent categories in its rewrite rules, and such rules are in general inadequate to deal with unbounded dependencies. Hence a relatively infrequent type of construction forces us to use such devices as transformations or slash categories. Although these devices may also be used in other constructions, we still find that the range is quite small. Likewise, a monocephalic dependency grammar has no general way to represent unbounded dependencies either, so we are forced either to allow multiheaded constructions or to make some other addition. All of these approaches can be regarded as treating unbounded dependency constructions as non-canonical and hence as fundamentally different from most other constructions.

In contrast flexible categorial grammar needs no special mechanism.[7] Rather we can assume that some lexical items have particular types, which have a complex argument that consists of a functor and an argument type. This allows the extracted element to combine with something that is not a full constituent. We shall later show that this property is precisely what we want for incremental interpretation anyway. We shall also show that if we consider how to make constituency theories incrementally plausible, then the use of traces becomes more bizarre. First, let us compare specific flexible categorial grammar treatments of unbounded dependencies.

4.4 Combinatory Categorial Grammar and Dependency[8]

---

[7] The Lambek Calculus cannot straightforwardly deal with non-peripheral extraction. But this is a limitation of **L** rather than of flexible categorial grammars in general. See chapter 7.

4.4.1 An Introduction to Combinatory Categorial Grammar

In a number of papers Steedman (eg 1985; 1987; 1988; Ades and Steedman 1982) has proposed a system of flexible categorial grammar which is more restricted than **L**, and bears considerable resemblance to the dependency-preserving subset of **L**. It should be clear that the goals of Steedman's research have been a great influence on the present theory. He allows flexible but not structurally-complete constituency, and thus gives a characterization of, for instance, coordination, extraction and incremental interpretation. He has independently described a general processing model, strongly influenced by the linguistic approach (eg 1989; Crain and Steedman 1985; Altmann and Steedman 1988). However, we have not yet considered his specific grammatical theory, Combinatory Categorial Grammar or CCG. The main theoretical difference from the present account is that CCG lacks a direct characterization of dependency, which, it can be argued, reduces its linguistic motivation and prevents generalizations over which combinations are permitted.

CCG uses a particular set of rules, known as *combinatory rules*, to assign types to all and only those strings that the theory regards as constituents. This contrasts with **L** where types can be assigned to strings that are not (dependency) constituents. The permitted rules are motivated by Curry and Feys's (1958) application of combinatory logics and correspond to the use of a set of combinators. CCG is more powerful than **AB**. It allows some, but not all, **L**-valid rules, and a number of other rules that are not **L**-valid. These **L**-invalid rules are highly restricted in their applicability and can therefore reasonably be regarded as peripheral.

CCG restricts possible type combinations to those that can be achieved by means of a particular set of binary and unary rules, including the following (ignoring the semantics):

(4.16) Forward application (>) $X/Y \;\; Y \Rightarrow X$

Backward application (<) $Y \;\; Y\backslash X \Rightarrow X$

Forward composition (>**B**) $X/Y \;\; Y/Z \Rightarrow X/Z$

Forward lifting (>**T**) $X \Rightarrow Y/(X\backslash Y)$[9]

---

[8] This section is derived from joint work with Guy Barry. See Barry and Pickering (1990) (bound into this thesis).

CCG does not include other **L**-valid inferences such as associativity ($((Y\backslash X)/Z \Leftrightarrow Y\backslash(X/Z))$, even though it is dependency-preserving. On the other hand, forward lifting is not dependency-preserving but is included in CCG. The choice of rules is made on the basis of specific linguistic phenomena, and hence there is no straightforward characterization of which rules are permitted, but there is a general assumption that the set of rules will allow the characterization of strings that can be coordinated and can be left after an extraction. CCG also puts type-specific restrictions on the use of the above rules. For instance, there is a stipulation that the use of the forward lifting rule is restricted to subjects:[10]

(4.17) $\mathrm{NP} \Rightarrow \mathrm{S}/(\mathrm{NP}\backslash \mathrm{S})$

The **L**-invalid rules are known as mixing rules, because they do not respect the simple interpretation of the directional slashes. For instance, Steedman allows the following rule:

(4.18) Backward mixed composition (<**B**x) $Y/Z \;\; Y\backslash X \Rightarrow X/Z$

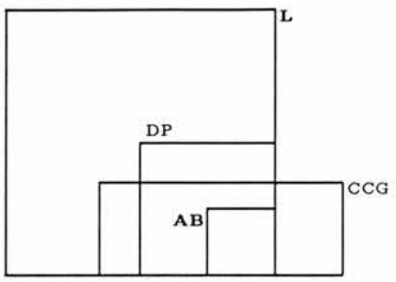
Moortgat (1988b) has shown that adding such a rule to **L** allows any word order to be accepted, and we have a system equivalent in power to the non-directional Lambek Calculus known as **LP**, which pays no respect to word order. This means that the differentiated meanings of the directional slashes is lost. For instance, if an expression combines with an immediately following Y to form an X, then it may also be combined with an immediately preceding Y to form an X, which means that if *John walks* is a sentence, then so is *Walks John.* In order to prevent this conclusion, Steedman imposes type-specific restrictions on mixing rules. Finally we should note that Steedman (1987) introduces another type of rule, functional substitution, which is in fact **LP**-invalid as well as **L**-invalid, in order to deal with parasitic gaps (Engdahl 1983).

The following diagram shows the relationship between CCG and **AB**, **L**, and the dependency-preserving subset of **L** (indexed DP):

---

[9] Steedman refers to this as *forward type-raising* (and in earlier papers gives it the index **C**$_{\bullet\bullet}$.)

[10] There are some changes between papers over the precise set of rules and restrictions.



Henceforth we shall ignore the **L**-invalid rules and so regard CCG as a subset of **L**.

4.4.2 Comparison of Dependency Constituency and CCG Constituency

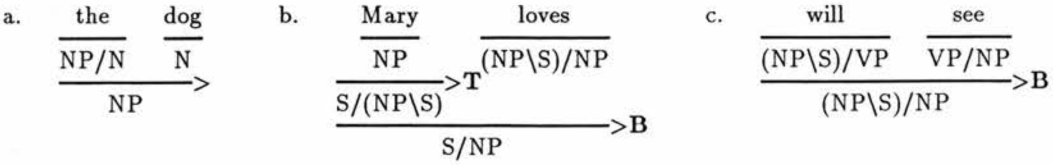
L gives types to all strings, and so not all typed elements are dependency constituents. CCG only gives types to constituents, and hence there is no way to interpret strings that are not constituents. This distinction is however not important if the Dependency Interpretation Hypothesis is assumed, because non-dependency constituents are never interpreted. In other respects the notions of constituent are roughly similar but specifically different. Because CCG includes some non-dependency-preserving rules but not all dependency-preserving ones, it is not surprising that it generates some non-dependency constituents but not all dependency constituents. Let us return to our examples from chapter 3 (repeated here as (4.19) and (4.20)):

(4.19) a. The dog runs.
b. Mary loves John.
c. John will see Mary.

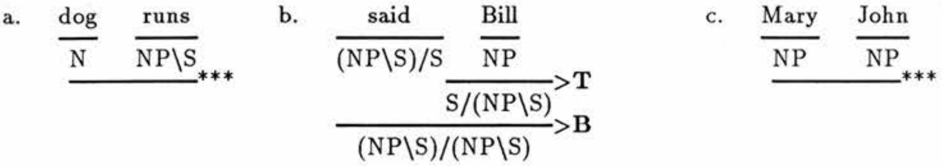
(4.20) a. The dog runs.
b. Fred said Bill laughed.
c. I showed Mary John.

Recall that the underlined fragments in (4.19), but not those in (4.20), are dependency constituents. Steedman's analysis can give a type to the fragments in (4.19a–c) and (4.20b), but not to those in (4.20a) or (4.20c):

(4.21)

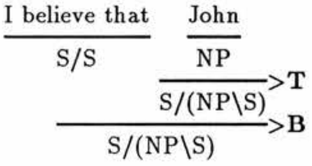


(4.22)



The only point of disagreement here is *said Bill* in (4.20b), which is a CCG constituent but not a dependency constituent. This would not happen if the non-dependency-preserving operation of lifting were absent from the grammar.[11] Similarly, the CCG account gives *I believe that John* the type S/(NP\S):

(4.23)



Mary can be lifted to this type as well, and so the ungrammatical coordination in (3.44) (repeated here as (4.24)) is allowed:

(4.24) *[I believe that John] and [Mary] is a genius.

In the last chapter, this coordination was ruled out because neither conjunct can be given the type S/(NP\S) by dependency-preserving operations, and because the conjuncts are not parallel. We can see that CCG overgenerates at those points where it extends beyond the dependency-preserving subset of **L**.

But even if the CCG rules were modified to generate all and only dependency constituents, it would still be impossible to account for phenomena that involved the coordination of non-dependency constituents, as for instance in (3.39), repeated below:

(4.25) John loves [Mary madly] and [Sue passionately].

---

[11] But removing lifting would have other implications, for instance preventing *Mary loves* in (4.19b) from being a constituent.

Without introducing the schema of parallel coordination, the only way we can generate such examples is to introduce non-dependency-preserving rules. For example, Dowty (1988) deals with such constructions by using combinatory rules of backward composition and backward lifting[12] (he assumes the latter to be lexical):

(4.26) Backward composition (<**B**) $Z\backslash Y \quad Y\backslash X \Rightarrow Z\backslash X$

Backward lifting (<**T**) $X \Rightarrow (Y/X)\backslash Y$

Although both rules are **L**-valid, only backward composition is dependency-preserving. Thus the conjuncts in (4.25) are constituents under Dowty's analysis, but not dependency constituents:[13]

(4.27)

$$\dfrac{\dfrac{\dfrac{\text{Mary}}{\text{NP}}}{((\text{NP}\backslash\text{S})/\text{NP})\backslash(\text{NP}\backslash\text{S})}{<}\mathbf{T} \qquad \dfrac{\text{madly}}{(\text{NP}\backslash\text{S})\backslash(\text{NP}\backslash\text{S})}}{((\text{NP}\backslash\text{S})/\text{NP})\backslash(\text{NP}\backslash\text{S})}{<}\mathbf{B}$$

This looks for a transitive verb to its immediate left, and a subject noun phrase beyond. But Dowty's account erroneously allows the forbidden coordination in (3.42) (repeated here as (4.28)), since both *Mary madly* and *Sue* can be analysed as constituents of type ((NP\S)/NP)\(NP\S):

(4.28) *John loves [Mary madly] and [Sue].

There is no way to prevent this within the general terms of his analysis.

4.4.3 Constraints on Extractions

CCG also forbids the construction of some dependency constituents. The primary motivation for this is to capture certain constraints on extractions. The problem is that it appears necessary to generate these constituents for other purposes.

Ross (1967) showed that it is impossible to extract from many environments. For instance, (4.29) is an instance of the 'complex NP constraint' which prohibits extractions from within noun phrases:

(4.29) *Beans, I met a man who likes.

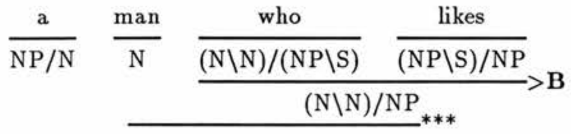
---

[12] Like Steedman, Dowty calls this backward type-raising.

[13] Note that he treats modifiers as having the lexical types of functors.

A man who likes is a dependency constituent.[14] Dependency constituency therefore does not give us a complete account of island constraints. Steedman's system on the other hand does not allow *a man who likes* to be analysed as a constituent (because there is no way in CCG to lift the type for *man* or to apply associativity to the type for *who likes*):

(4.30)



Preventing this derivation initially seems reasonable, but the problem is that strings like *a man who likes* have to be constituents for the purpose of coordination:

(4.31) Most people hate, but I know a man who likes, sonatas by Mozart.

This is allowed in the present account because the conjuncts are both dependency constituents of type S/NP (compare (4.24) above). In general, there is a clear contrast between extraction and coordination, because the restrictions on acceptable extractions are greater than those on coordination (even ignoring cases of parallel coordination). This makes a unified account of extraction and coordination impossible. Either we generate ungrammatical extractions, or we prohibit acceptable coordinations. Steedman does not recognise this distinction, and hence his approach faces a general problem. He concentrates on preventing ungrammatical extractions and makes his working hypothesis the assumption that strings which cannot be left after extraction are not constituents.

There are a number of reasons to prefer the dependency constituency account. First, there is evidence from incremental processing, which will be considered in the next section. Second, we can always add specific restrictions on extracted elements to rule out islands, whereas adding a special mechanism for certain coordinations between certain conjuncts that are not of the same type breaks with the strong assumption that only like types can conjoin.[15] Third, violations of many island constraints are sometimes acceptable. Judgements can vary between constructions which do not show

---

[14] Again, this is still true if we take modifiers to be functors.

[15] Even in non-dependency coordinations, the types of the conjuncts must be the same. This does not imply that we actually compute these types.

any difference in types. For instance *who did you see a picture of?* and *?*who did you see John's picture of?* only differ on the word *a* and *John's*, which are presumably both of type NP/N. An account that captures such differences will have to find some fine-grained featural difference between these words. But any straightforward syntactic account of island constraints will rule out acceptable examples.

Many 'standard' island constraint judgements are notoriously imprecise, but there are a number where ungrammaticality is totally clear. Perhaps because of this, these examples are rarely considered:

(4.32) a. *Loves, I know that Mary John.

b. *Loves John, I know that Mary.

a. *Mary loves, I know that John.

In all these examples, the forbidden extraction is taken to occur from the embedded clause. Notice that the string left after the extraction is not a dependency constituent. Generalizing from this, we can make the claim that if the remaining string is not a dependency constituent, then the extraction will be totally impossible (and there will not be variable judgements). This restriction rules out a number of theoretically possible extractions, for instance:

(4.33) *How many boys do you think in ten play football?

In this sentence, *how many boys* is taken to be extracted from the noun phrase *how many boys in ten*. But it is *how many boys* that contains the head, and is therefore the dependent of *play*. This means that the string *in ten play* is not a dependency constituent, and so neither is the string *do you think in ten play football*. It seems likely that no sentence with this structure will be acceptable. We can claim that many apparent counterexamples are actually cases of extraposition from an extracted item, and these will have to treated separately:[16]

(4.34) Which boy do you know that plays football?

---

[16] It is very unclear whether such a straightforward account will generalize to other languages. For instance, the facts of complex fronting in Germanic Languages (Uszkoreit 1986) may require modifying this account.

These two accounts can be distinguished, because extraposed material must come after all arguments of a verb. Hence the extrapositional account predicts the following contrast:

(4.35) Which boy did you send a letter that plays football?

(4.36) *Which boy did you send that plays football a letter?

If the head were extracted but the dependent left in place, (4.36) should be grammatical. Note that the extraposed element must be extraposed from the fronted element, not from the post-verbal location, because (4.37) is very bad:

(4.37) ?*You sent the boy a letter that plays football.

Dependency constituency may therefore help explain some island constraints. However it is clearly not an explanation for all.

CCG constituency is closely related to dependency constituency, but differs in certain specific respects. Unlike dependency constituency, it is not based on a traditional linguistic notion, and so it is very interesting that it has arrived at quite similar conclusions as to what strings are constituents. But there are some specific differences from the present account. Because it has no way to allow interpretation of strings that are not constituents, it makes some strings constituents in ways that lead to overgeneration, and because it assumes that it can explain the phenomena of island constraints syntactically, it encounters problems with coordination and, as we shall see, incremental interpretation. The conclusion is that it attempts to do too much with too small an apparatus.

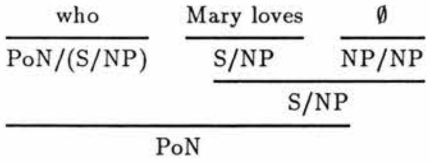
4.5 Incremental Interpretation of Unbounded Dependencies

This section first outlines the general way in which flexible categorial grammar represents the incremental processing of unbounded dependencies. In this approach it is possible to use empty categories, but they are quite superfluous. A conclusion is that incremental plausibility and the lack of traces are closely associated properties. I shall then claim that an incrementally plausible account of GPSG collapses into flexible categorial grammar (without traces), and that this is compatible with coordination data that fits badly with standard GPSG. I shall then show that the CCG account of unbounded dependencies, by attempting to account for island constraints, prevents the formation of constituents that are necessary in incremental processing. This aspect of CCG is therefore incompatible with the Minimal Incrementality Condition. This contrasts with the dependency-based account.

Extraction and coordination both involve units which are not traditional phrase structure constituents. For instance *Mary loves* in *John, Mary loves* and in *Mary loves but Sue hates Tom* does not form a phrase structure constituent. In flexible categorial grammar it is simply a constituent of type S/NP. Of course this unit is precisely what we need in an incremental interpretation of a canonical sentence like *Mary loves John.* The fact that the type S/NP is needed for coordination and extraction is strong evidence that *Mary loves* can always be given this type, whatever construction these words may form a part of.

Flexible categorial grammar does not explicitly represent coindexation, but it is possible to assign a type to an empty string. The obvious types are X/X for right-peripheral extraction, and $X\backslash X$ for left-peripheral extraction (with both Xs representing basic types with the same subscript[17].) In this way, *who Mary loves* could be given the following analysis:

(4.38)



But why assume an empty category here? It appears to serve no purpose. Another possibility is to give the empty string a simple category, eg NP, and give the relative pronoun the category PoN/S. In this case the semantics would be **who (loves $\emptyset$ Mary)**, and we would need to assume an additional (presumably anaphoric) mechanism to link *who* and $\emptyset$. This is closer to the spirit of a trace-theoretic approach like GB. But now we have assumed both traces and some coindexation process. An incremental version of

---

[17] Since modifiers can sometimes be extracted, the restriction to basic types only works if modifiers are arguments.

a trace-theoretic approach is unparsimonious. In the next chapter we shall also see why either view of traces in categorial grammar fail to satisfy the MIC in some constructions, so there is in fact a stronger argument than parsimony against assuming traces.[18]

Let us now consider GPSG in more detail. In GKPS, unbounded dependencies are analysed using a single slash feature. But there are certain constructions that cannot be analysed within these limitations and require enrichments that make it very like flexible categorial grammar. If X and Y are non-derived categories, then the form of derived categories is X/Y, with Y regarded as a feature of X. We have already noted that, in Scandanavian languages, Maling and Zaenen (1982) showed the need for the use of more than one slash. If Z is also a non-derived category, we need derived categories of the form $(X/Y)/Z$, with X having two features, Y and Z. We can show the need for such categories in English if we treat right node raising by using a rule like S ⟶ S/NP NP, and then coordinating S/NPs, as in [[[Mary met]$_{S/NP}$ but [Sue avoided]$_{S/NP}$]$_{S/NP}$ [Tom]$_{NP}$]$_{S}$. It is not clear how to perform 'non-constituent' coordination without using the slash category (GKPS do not consider these examples). If we add a string of adjuncts, we need one slash for every adjunct:

(4.39) Mary met but Sue avoided Tom in the park on Monday in the pouring rain.

We need to apply the rule S ⟶ S/NP NP, and another, generalized rule, X ⟶ X/AdvP AdvP three times, to give each conjunct the category (((S/NP)/AdvP)/AdvP)/AdvP. So even in English, we need to be able to use multiple slashes in GPSG. But it looks as though one small class of constructions need a radically more powerful apparatus than other constructions (and, unlike multiple extractions, (4.39) causes no processing difficulties). But so far we have no case of the derived category itself being complex. Dowty (1988) (see Morrill 1988 for many related examples) shows that this is also necessary for sentences like:

(4.40) Bill gave and Max sold a book to Mary and a record to Susan.

---

[18] It should be quite clear that the assumptions made in the natural deduction axiomatization do not correspond to empty categories that are regarded as a real part of sentence processing. All combinations can be made in another axiomatization of the Lambek Calculus, such as the sequent calculus, where there is nothing corresponding to empty categories.

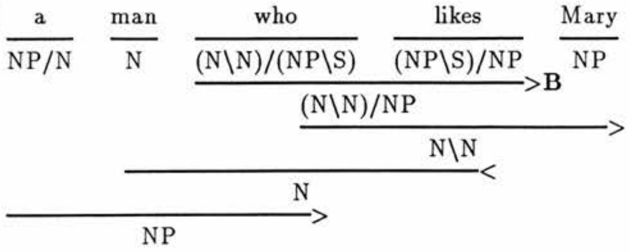
Dowty gives the analysis where *Bill gave and Max sold* has the category S/(VP/V) and *a book to Mary and a record to Susan* have the category VP/V (there is another analysis where they are given categories (S/PP)/NP and VP/((S/PP)/NP) respectively). There is a slash feature which has to bear a slash feature itself. This requires a massive extension to the GPSG machinery, for a very unusual construction, and hence suggests something is wrong.

Of course this now looks very like flexible categorial grammar, and it can be shown that the categories given by Dowty's analysis, when converted to categorial grammar, are equivalent types to those that **L** would give them, after introducing directionality. These types are needed for linguistic reasons, but we also require similar flexibility (slashes within slashes) in order to allow incremental interpretation. For instance, the incrementally constructed type of *The man who* in *The man who sings is excellent* is NP/(S/NP). We want the analysis of *Bill gave* in any sentence to be equivalent to S/(VP/V), or S/PP/NP in categorial grammar, not just in a few special constructions when some otherwise unused rule is accessed. If the rule can be used for Dowty's sentences, why not make it generally available?

Hence an incrementally plausible GPSG collapses into a form of flexible categorial grammar, and because there is no incremental value in having traces, we can make use of a form of GPSG without traces. We may keep the linguistic insights of GPSG so long as we abandon the commitment to phrase structure constituency. If we do this then there is no reason to assume empty categories. In contrast to GPSG, it is extremely unclear how we could treat GB without empty categories. Any attempt to make GB incrementally plausible will presumably retain empty categories (Pritchett 1988; Crocker 1990).

Some versions of flexible categorial grammar may be unable to deal with incremental processing, however, if we make the assumption that only constituents can be interpreted. The CCG treatment of island constraints forbids a (head) noun to combine with an incomplete relative clause. This has serious implications for processing, because it predicts that a noun cannot be combined with a following relative clause until the entire relative clause is processed. For instance, the maximally incremental derivation of *a man who likes Mary* in CCG is as follows:

(4.41)



This would incorrectly predict that multiple relative clause constructions such as (3.77) (repeated here as (4.42)) should *always* become difficult to process at some point, irrespective of the structure of the relative clauses themselves:

(4.42) John saw the farmer who owned the dog which chased the cat which followed the mouse which nibbled the cheese.

As this is not the case, the CCG analysis fails to satisfy the Minimal Incrementality Condition. The reason for this is that it is impossible to combine the relative pronouns with their antecedents until the end of the relevant relative clause, and none of the relative clauses finishes until the end of the sentence. In other words, we will have to make use of an indefinitely deep stack. This has to extend beyond memory capacities eventually. Because this does not happen in processing, we can be sure that this inability to combine elements is wrong. This problem is not encountered in the present formulation, because we have shown that the number of incremental dependency constituents in (4.42) does not increase each time we encounter a further embedded relative clause. Hence there is support for the importance of dependency constituency as well as the utility of the general framework of flexible categorial grammar in the processing of unbounded dependencies.

4.6 Experimental Accounts of Unbounded Dependencies

Psycholinguistic accounts of unbounded dependencies have almost invariably assumed some version of trace theory. The extracted element is taken to be associated with the extraction site or trace by means of a coindexation mechanism. The psycholinguistic theory then determines how this coindexation takes place. This process is known as 'gap-filling', with 'filler' referring to the moved or extracted element, and 'gap' referring to the extraction site. This association is sometimes regarded as anaphoric, but is also sometimes described as a 'filler-gap dependency', although it is actually very unclear how this form of association relates to more standard dependencies. Therefore the terms 'unbounded dependency' and 'filler-gap dependency' are very confusingly used interchangeably, even though the former term is descriptive and the latter assumes a particular treatment of the relevant constructions.

4.6.1 Historical Notes

The study of sentence processing has for a long time been associated with transformational grammar and its recent descendents. There have been many attempts to show the psychological reality of the grammars and the syntactic representations. Chomsky and Miller proposed the Derivational Theory of Complexity, which assumed that we mentally reverse all transformations in order to return to the Kernel string. This was rapidly seen to be completely computationally intractable and psychologically implausible, and was not supported by experimental evidence (see Fodor, Bever and Garrett 1974). As an alternative, Bever (1970) proposed that we process language by means of perceptual strategies uncertainly related to the grammar. This is therefore a clear violation of the Strong Competence Hypothesis. Hence Wanner and Maratsos (1978), reporting on work from the early 1970s, proposed a mechanism for processing relative clauses within the context of an augmented transition network or ATN, which was computationally tractable and relatively psychologically plausible. It mimicked the effects of transformations without actually reversing them. However, it became at least superficially related to the model of grammar with the advent of trace theory. This has formed the basis for most subsequent models of the processing of unbounded dependencies.

Wanner and Maratsos's ATN treats unbounded dependencies by putting the filler into a special memory store called HOLD and leaving it there while the rest of the sentence is parsed. When a verb (for instance) is encountered, the next word is checked, and if this is a permissible part of an NP (eg a determiner), then this is processed as an argument to the verb. If no NP is encountered, it is assumed that the NP in HOLD must be retrieved and treated as the argument. In this way the gap is filled by the NP in HOLD. This model is consistent with the trace-theoretic approach, in that it describes a method for establishing the associations that are linguistically described by coindexation, but it does have a number of problems. It appears to assume 'literal' gap-filling, with the extracted element placed directly in the extraction site. Therefore the interpretation of a topicalized sentence, for instance, would have to be the same as that of its canonical counterpart. But there is a clear difference in preferred interpretations between (4.43) and (4.44), which indicates that something more than literal gap-filling must occur:

(4.43) Some man, every woman loves.

(4.44) Every woman loves some man.

Another limitation is that it has nothing to say about rightward extraction, such as heavy-NP-shift, when the gap comes before the filler. Finally, it assumes gap-filling as a 'last resort' strategy, only if a lexical NP cannot be found (directly) after the verb. Hence for instance adjunct gaps as in (4.45) would never be filled:

(4.45) How often does Bill see Mary?

Only the last of these problems has been addressed at any length in the more recent literature on gap-filling. We shall return to these issues in chapter 6.

Wanner and Maratsos tested this model experimentally, by presenting subjects with sentences like (4.46) and (4.47) below:

(4.46) The witch who despised sorcerers frightened little children.

(4.47) The witch whom sorcerers despised frightened little children.

The sentences were interrupted at various points, and subjects were made to remember lists of words. Memory was impaired to a greater extent in the object relative sentence, but only if the interruption was made within the relative clause, when the filler has been discharged in (4.46) but not in (4.47). It is assumed that remembering the filler takes up some memory space which reduces the amount available for remembering the list of words. This result shows that the processes involved in forming long-distance dependencies are amenable to experimental investigation. Wanner and Maratsos investigated this claim by considering the memory load created by the element that is in store, but most more recent studies have concentrated on the actual formation of the unbounded dependency.

4.6.2 Reading and Reactivation

One problem with Wanner and Maratsos's experiment is that it can be explained in a number ways which are unrelated to the effort of remembering the filler. It may be that the process of extracting an object is harder than that of extracting a subject, or it may be that no unbounded relation at all is involved in simple subject relatives. These possibilities will be discussed in chapter 7. Therefore it does not give unambiguous evidence that the process of forming an unbounded dependency is psychologically real and (more importantly) investigatable. It is probably easier to examine the formation of the dependency rather than the memory for the extracted element. There are a number of possible techniques. A typical methodology is self-paced reading, which can be conducted either word-by-word or phrase-by-phrase. Subjects press a button on a computer to present each word or phrase, and are instructed to read what they see and then to press the button again when they have done this. The time between presses is measured by the computer and is assumed to reflect the time subjects have to spend in order to understand what they read.[19]

The simple reading time method was employed by Crain and Fodor (1985) and by Stowe (1986). Crain and Fodor found that subjects took longer to read the word *us* in a question like (4.48) than in a declarative sentence like (4.49), though at all other points in the sentence no such difference was found:

(4.48) Who could the little child have forced us to sing those stupid French songs for last Christmas?

(4.49) The little child could have forced us to sing those stupid French songs for Cheryl last Christmas.

Therefore we know that there is something complex about the processing of *us* in (4.48). Crain and Fodor argue that subjects locate an empty category after *expected*, because *who* could be the object of *expected*, and associate the filler with this empty category. The appearance of *us* shows that this analysis is in fact wrong, and so we are forced to undo this analysis, which increases processing time. Note that this model assumes that

---

[19] A rather less naturalistic alternative is for subjects to read each word and press one key to continue the sentence, and press another if they believe that the sentence has become ungrammatical (eg Stowe 1989).

we locate a gap immediately whenever it is possible, which contrasts with Wanner and Maratsos's assumption, that gaps are only located if there is no relevant unextracted argument present. Crain and Fodor's model is therefore an 'early' gap-filling model, Wanner and Maratsos's a 'late' one. Crain and Fodor's results support an early model, but in this experiment it is possible that this may be due to the relatively slow rate of presentation in self-paced reading time. However, we shall see in chapter 6 that this support for an early model is not artefactual.

Reading time measures show the effort involved in processing unbounded dependencies, but do not indicate anything about how the extracted element is related to the body of the sentence. In contrast, Swinney, Ford, Frauenfelder and Bresnan (1988) describe a method which is sensitive to the properties of the extracted item. Associative priming effects are well known for words presented immediately after one another, as described in chapter 2. However, the effect dies away more or less rapidly when the time between the presentation of *bread* and *butter* is increased (Swinney 1979), depending on the nature of the intervening material. Swinney et al showed that this priming effect can be found in sentences with unbounded dependencies when an associate of the extracted element is presented, not immediately after the extracted element, but at the point where the trace is assumed to be located. Subjects listened to sentences like (4.50) and at the same time watched a screen, on which an associate of *boy* was presented:

(4.50) The policeman saw the boy that the crowd at the party accused
$\emptyset$ of the crime.

They found facilitation for associates of *boy* (but not of the other noun phrases), at the gap location, but not just prior to the gap location. This indicates that there is a reactivation of the properties of the filler.

4.6.3 Interpretation of Experiments

It is sometimes assumed that these results prove the psychological reality of *wh*-trace. For instance, J.D.Fodor (1989) calls her review of processing 'non-canonical' constructions 'Empty Categories in Sentence Processing', and assumes that processing unbounded dependencies involves finding *wh*-traces and assigning antecedents. She contrasts the evidence for *wh*-trace with other empty categories, and concludes that the existence of the former is not in doubt (and therefore that the only questions are how the trace is used), but that the existence of the other empty categories proposed by GB is not certain. We can therefore assume that there is a gap-filling mechanism in some sense separate from the rest of syntactic processing. I shall return to this issue in chapter 6.

Of course the evidence that Fodor reviews show that the computation of the unbounded dependency has measurable psychological effects. This is certainly important, but because there is no doubt that some long-distance association must occur, it does not tell us anything of linguistic (as opposed to psycholinguistic) interest. This is because the evidence is just as compatible with models that do not assume *wh*-trace (or any other empty categories) as with models that do. Consider Swinney et al's example again. They assume that there is an association between the *wh*-trace and the NP (*the boy*) that serves as the head of the relative clause, but because the trace is adjacent to the embedded verb *accused*, it is just as possible that the unbounded dependency is not formed via the trace at all, but rather that the extracted element and its governor are associated directly. Similarly, Crain and Fodor's experiment was interpreted in terms of a theory with traces. We locate a gap after *expected* in (4.48), and then find that this is not the correct analysis, so we have to backtrack. It is also possible that we simply associate *who* with the verb *expected*, but when we encounter *us*, we find that *who* should not have been associated with *expected*, and therefore that the unbounded dependency should not have been formed yet, and it is this which causes us to have to backtrack. The problem is that in these experiments we cannot distinguish between an account with traces and an account without traces, because the trace is adjacent to the word which the extracted element must associate with to form the unbounded dependency. In the next chapter we shall consider ways to distinguish these two accounts.

Chapter 5

Processing with and without Gaps[1]

5.1 Distinguishing Models with and without Gaps

The previous chapter indicated that psycholinguistic evidence that has been interpreted as giving evidence for *wh*-trace can also be interpreted as showing a relationship directly between the extracted element and its governor. So for instance, in *John, Mary loves*, the association could be between the extracted element *John* and a trace after *loves*, or it could be between *John* and the verb *loves* directly. Any methodology sensitive to processing will find it very difficult to distinguish between these explanations because the proposed location for the trace and the verb are adjacent. This problem can however be avoided, by considering a construction where the related unextracted form does not have all the dependents adjacent to the head. An obvious case of this is verbs that take two post-verbal arguments, such as *put*:

(5.1) I think I put the tray in the box.

Hence, if the second post-verbal argument is topicalized, a trace-theoretic account would posit a structure like:

(5.2) [In the box]$_a$, I think I put the tray $\emptyset_a$.

---

[1] A large part of this chapter is based on joint work with Guy Barry. See Pickering and Barry (1989) (bound into this thesis) and Pickering and Barry (1990).

However, theories such as flexible categorial grammar which do not use empty categories would give this sentence a clearly different analysis, where the association takes place with the verb *before* the direct object is encountered:

(5.3) [In the box]$_1$, I think I [put]$_1$ the tray.

The coindexation describes an unbounded dependency relation in the loose sense introduced in the previous chapter. The second index is at the point where a single dependency constituent is formed. The two indexes do not correspond to a dependency relation as introduced in chapter 2. There is no need to interpret the coindexation in terms of a dependency theory; it could be an equally meaningful representation of other accounts without empty categories.

It now looks as though we have found a testable difference between the two models. In this chapter I shall differentiate between these models by recursing the extraction of second post-verbal arguments. The discussion will deliberately be kept very general, so that the distinction is simply between a model with gaps and a model without gaps. But the last section relates the model without gaps to the general account proposed in the previous chapters.

5.2 Processing Nested Constructions

5.2.1 Nested Constructions

Much discussion of the distinction between linguistic competence and performance has centred around constructions such as the following, in which one grammatically complex element appears in the middle of another:

(5.4) I believe that the boy who the students recognized is a genius.

(5.5) The man who the boy who the students recognized pointed out
is a friend of mine.

I have already referred to such constructions as nested constructions, but without giving any formal description of their properties. Let us consider Chomsky's (1965) definitions of *nested construction* and *self-embedded construction*:

> The phrases A and B form a nested construction if A falls totally within B, with some nonnull element to its left within B and some nonnull element to its right within B. ... The phrase A is self-embedded in B if A is nested in B and, furthermore, A is a phrase of the same type as B.

For example, in (5.4) *who the students recognized* is nested in *that the boy who the students recognized is a genius.* Similarly, in (5.5) *who the students recognized* is both nested and self-embedded in *who the boy who the students recognized pointed out.*[2]

In a discussion of how performance factors might affect the acceptability of various types of grammatical structure, Chomsky observes that 'repeated nesting contributes to unacceptability' and that 'self-embedding contributes still more radically to unacceptability', whereas 'there are no clear examples of unacceptability involving only left-branching or only right-branching'. He suggests that this is 'simply a consequence of finiteness of memory', giving the following reasons:

> In some measure, these phenomena are easily explained. Thus it is known ... that an optimal perceptual device, even with a bounded memory, can accept unbounded left-branching and right-branching structure, though nested (hence ultimately self-embedded) structures go beyond its memory capacity.

However, let us now consider:

(5.6) John found the box in which I put the tray on which Mary placed the cake.

By Chomsky's definitions *on which Mary placed the cake* is neither nested nor self-embedded in *in which I put the tray on which Mary placed the cake*, since no lexical material from the latter lies to the right of the former. This seems reasonable given that (5.6) is not hard to process in the way that (5.5) is.

The problem with this is that (as we shall see below) analyses of (5.6) within, for instance, the Government-Binding framework (Chomsky 1981) assume that there are *two* gaps at the end of the sentence (*the tray on which Mary placed the cake* being all one argument of *put*). The analysis that this sentence requires seems highly strange, because the derived psychological model entails that we have to hold both fillers in

---

[2] The additional term 'centre-embedding' has been used more recently to refer both widely to nesting and narrowly to self-embedding, causing some confusion of terminology. Since self-embedded constructions are merely a subset of nested constructions, to save confusion we shall usually refer to sentences like (5.4) and (5.5) simply as 'nested constructions', although nearly all the examples we give of nesting will also be examples of self-embedding.

memory right until the end of the sentence when we encounter both of their associated gaps. Chomsky's definition of *nested construction* avoids this problem by the use of the word *nonnull*, which seems reasonable if perhaps arbitrary. The problems that (5.6) raises for the theory of empty categories and filler-gap association are more severe. The associations that the empty elements make with their antecedents are nested, as we shall see. This seems wrong for a sentence that is not hard to process as nested constructions are.

The rest of this section considers this in more detail, by giving a precise description of the relevant aspects of the empty category account, and the difficulties that it leads to. This is then contrasted with an account where the extracted element and the embedded verb are associated directly.

5.2.2 Filler-Gap and Gap-Governor Associations

Let us refer to the extracted element as the filler. In a theory with gaps, we have to make two types of association between the gap and the other parts of the sentence. One type is the coindexation relation between the filler and the gap, which we shall refer to (naturally) as a *filler-gap* association. This is the standard association that is referred to as a filler-gap association.[3] For this association we shall use alphabetic subscripts, so that *John, Mary loves* is annotated:

(5.7) $[\text{John}]_a$, Mary loves $\emptyset_a$.

The other type is what would be thought of as a dependency relation between the gap and whatever would serve as the governor of the filler in the canonical sentence. In *Mary loves John*, *John* depends on the verb *loves*, so when *John* is topicalized, a dependency must be established between the gap and *loves*. In general, we shall refer to this relation as a *gap-governor* association.

We shall use superscript Greek letters to refer to gap-governor associations in examples, in order to distinguish them from filler-gap associations. *John, Mary loves* is therefore given the following annotation:

(5.8) $[\text{John}]_a$, Mary $[\text{loves}]^\alpha$ $\emptyset_a^\alpha$.

---

[3] As mentioned in chapter 4, this is sometimes confusingly called a filler-gap *dependency*. It is more properly a filler-gap association, which may be regarded as part of an unbounded dependency.

As another example, this system will give the embedded question *John wonders who Mary loves* the filler-gap and gap-governor associations in (5.9):

(5.9) John wonders [who]$_a$ Mary [loves]$^\alpha$ $\emptyset_a^\alpha$.

5.2.3 Nested and Disjoint Association Patterns

If we look at sentences in which more than one deviation from canonical form occurs, we obtain some interesting patterns of associations. Consider the contrast between the following pair of English sentences:

(5.10) I saw the farmer who owned the dog which chased the cat.

(5.11) The cat which the dog which the farmer owned chased fled.

We shall refer to sentences such as (5.10) as *multiple subject relative* constructions, and sentences such as (5.11) as *multiple object relative* constructions. Let us consider the structures of these two sentences in terms of the patterns of association described above. The annotated versions of the sentences run as follows:

(5.12) I saw the farmer [who]$_a$ $\emptyset_a^\alpha$ [owned]$^\alpha$ the dog [which]$_b$ $\emptyset_b^\beta$ [chased]$^\beta$ the cat.

(5.13) The cat [which]$_a$ the dog [which]$_b$ the farmer [owned]$^\alpha$ $\emptyset_b^\alpha$ [chased]$^\beta$ $\emptyset_a^\beta$ fled.

There is a clear contrast between the patterns of filler-gap association in the two sentences: the pattern in (5.12) is *aabb*, but the pattern in (5.13) is *abba*. We shall refer to association patterns of the form *aabb*, *aabbcc* etc. as *disjoint* association patterns, and patterns of the form *abba*, *abccba* etc. as *nested* association patterns. (There are of course various other possible patterns of association, but we shall not consider them here.) However, there is no difference between the patterns of gap-governor association in the two sentences; both have the disjoint pattern $\alpha\alpha\beta\beta$. (In (5.12) the gaps precede their verbs and in (5.13) they follow them, but this does not affect the pattern.) Thus we may characterize sentences such as (5.10) as exhibiting disjoint filler-gap and disjoint gap-governor associations, and sentences such as (5.11) as exhibiting nested filler-gap and disjoint gap-governor associations.

In further contrast, the German multiple subject relative construction in (5.14) has disjoint filler-gap and nested gap-governor associations:

(5.14) Der Bauer der das Mädchen das den Jungen küßte schlug ging.
The farmer$_{NOM}$ who$_{NOM}$ the girl$_{ACC}$ who$_{NOM}$ the boy$_{ACC}$
kissed hit went.
'The farmer who hit the girl who kissed the boy went'

The annotated version (5.15) below shows that the filler-gap association pattern is *aabb*, and the gap-governor association pattern is $\alpha\beta\beta\alpha$:

(5.15) Der Bauer [der]$_a$ $\emptyset^{\alpha}_{a}$ das Mädchen [das]$_b$ $\emptyset^{\beta}_{b}$ den Jungen [küßte]$^{\beta}$
[schlug]$^{\alpha}$ ging.

Finally, let us return to the construction in (5.6), which we shall refer to as *multiple pied-piping*. This has nested filler-gap and nested gap-governor associations. The annotated version (5.16) has filler-gap association pattern *abba* and gap-governor association pattern $\alpha\beta\beta\alpha$:[4]

(5.16) John found the box [in which]$_a$ I [put]$^{\alpha}$ the tray [on which]$_b$
Mary [placed]$^{\beta}$ the cake $\emptyset^{\beta}_{b}$ $\emptyset^{\alpha}_{a}$.

We have now categorized the four sentence types in terms of filler-gap and gap-governor associations, but of course we can also categorize them in terms of whether or not they are nested constructions in Chomsky's sense. We might presume that these two systems of categorization are related, but, as the following table shows, any relationship between them is at best obscure:

---

[4] For simplicity I shall treat phrases such as *in which* and *on which* as single lexical items.

Sentence type	Example	Filler-gap pattern	Gap-governor pattern	Construction type
English multiple subject relative	5.10	Disjoint	Disjoint	Non-nested
English multiple object relative	5.11	Nested	Disjoint	Nested
German multiple subject relative	5.14	Disjoint	Nested	Nested
English multiple pied-piping	5.6	Nested	Nested	Non-nested

5.2.4 Processing Complexity

Nested associations cannot be easier to process than disjoint associations for the following reason. In the pattern *aabb* we are able to associate the two *a*s together immediately. On the other hand, in the pattern *abba* we have to hold one *a* in memory while we associate the *b*s, and it is only after associating the *b*s that we are able to associate the two *a*s together. With the disjoint pattern, we never have to remember an unassociated element while we are associating other elements unrelated to it. Having to remember more elements is likely to increase processing complexity.

It is of course possible that we can easily remember many elements simultaneously, and it is only after we exceed some large fixed capacity that any processing difficulties arise. Because of the processing difficulties with formally not very complex sentences like (5.11) and (5.14), however, we have good grounds to suspect that the formation of associations between linguistic elements uses some component of memory that is highly restricted in its ability to hold unassociated items securely. The same point was made in terms of dependency constituents in chapter 2.

In contrast, it seems that listeners have no particular difficulty in parsing (5.10), which has similar grammatical complexity to (5.11) and (5.14). Also, processing complexity does not increase if a further relative clause is added:

(5.17) I saw the farmer who owned the dog which chased the cat which scratched the girl.

This is expected given the analysis in terms of disjoint filler-gap and disjoint gap-governor associations; since there is no nesting of either type of association, there should be no build-up of unassociated elements at any point.

On the other hand any similar extension to the already difficult (5.11) makes the sentence virtually incomprehensible:

(5.18) The girl who the cat which the dog which the farmer owned chased scratched fled.

The above account leads naturally to an explanation in terms of nesting of filler-gap associations; the more nested associations, the harder the sentence should be to process. It supports the claim that our memory resources for unassociated elements are probably very limited.

German speakers find (5.14) hard in the same way that English speakers find (5.11) hard, and the addition of another relative clause again leads to extreme processing difficulty:

(5.19) Der Bauer der das Mädchen das den Jungen der die Katze streichelte küßte schlug ging.
The farmer$_{NOM}$ who$_{NOM}$ the girl$_{ACC}$ who$_{NOM}$ the boy$_{ACC}$ who$_{NOM}$ the cat$_{ACC}$ stroked kissed hit went.
'The farmer who hit the girl who kissed the boy who stroked the cat went'

This time we should presumably account for this difficulty in terms of the nesting of gap-governor associations.

On the above assumptions, then, it follows that a sentence such as the multiple pied-piping construction (5.6), with both nested filler-gap and nested gap-governor associations, should cause considerable processing difficulty, and this should furthermore increase if the number of relative clauses is increased. Empirical evidence, however, suggests to the contrary that the sentence patterns with (5.10) in processing terms, and in particular that the addition of a further relative clause causes no global increase in processing complexity:

(5.20) John found the box in which I put the tray on which Mary placed the dish from which Bill took the cake.

Processing this sentence is clearly like processing (5.17), not (5.18). We can now conclude that the constructions that are hard to process turn out to be just those that are nested in Chomsky's sense, rather than those that have nested associations.

It is of course possible that (5.11) and (5.14) are hard to process for some reason unrelated to the fact that they have nested associations. We may have an ability to remember unassociated items which far exceeds that which is needed to process these sentences. Such an account would therefore allow (5.6) to be easy to process. There are no obvious grounds for making this assumption, we should note. But more crucially, we know that there must be *some* limit on human memory resources, however large they may be. Therefore we can be absolutely certain that *at some point*, a sufficiently long sentence with nested dependencies will become unprocessable. This can be tested, by recursing the construction used in (5.6):

(5.21) John found the box in which I put the tray on which Mary placed the dish from which Bill took the saucer on which Sue stood the cup into which Fred poured the tea.

Now we have five gaps at the end of the sentence, and the pattern for both gap-governor and filler-gap associations is nested to a depth of five. There is no concomitant increase in processing difficulty, and this remains the case however much we extend the sentence. Note that some of the prepositional phrases may be considered to be adjuncts rather than arguments, but this does not seem to affect the issue. Of course, we may have difficulty in some higher-level aspects of interpretation of such a sentence, but parsing it is not a problem. This is the same as if we further recurse the construction in (5.17). However much we extend these sentences, we find the same flat, rhythmic intonation pattern, which we lose in the multiple object relatives like (5.11) or (5.18), for instance.

At this point we are forced to conclude that there is something wrong with the standard analysis of (5.6). The processing evidence, which is ultimately based merely on the assumption that memory resources are finite, shows that it is necessary to question the linguistic assumptions that have been made.[5]

---

[5] Notice the similarity of this argument to the arguments given in chapter 2 and chapter 4 for incremental processing, and for the Minimal Incrementality Condition.

5.2.5 Associations with and without Gaps

We have so far assumed a processing account that uses empty categories, and assumes both filler-gap associations, and gap-governor associations. Let us instead consider an account without empty categories, where the extracted element associates with the verb directly. Although the displaced element *John* is no longer really a 'filler', we shall retain this terminology for present purposes and refer to this new type of association as a *filler-governor* association, which we shall distinguish from other types of association by means of common numeric subscripts:

(5.22) [John]$_1$, Mary [loves]$_1$.

Because this account does not use gaps, it is in some sense more parsimonious than the account with gaps, as was suggested in the last chapter. Since this filler-governor association simply represents the necessary connection between these two elements, we shall refer to it as a filler-governor *dependency*.

Let us now recast the analyses of our four sample sentences in terms of filler-governor dependencies. We obtain the following patterns of association:

(5.23) I saw the farmer [who]$_1$ [owned]$_1$ the dog [which]$_2$ [chased]$_2$ the cat.

(5.24) The cat [which]$_1$ the dog [which]$_2$ the farmer [owned]$_2$ [chased]$_1$ fled.

(5.25) Der Bauer [der]$_1$ das Mädchen [das]$_2$ den Jungen [küßte]$_2$ [schlug]$_1$ ging.

(5.26) John found the box [in which]$_1$ I [put]$_1$ the tray [on which]$_2$ Mary [placed]$_2$ the cake.

Because there is only one type of association in this new analysis, there is only one contrast to be made, whether the pattern is nested, 1221, or disjoint, 1122. We see that the multiple object relative (5.24) and the German multiple subject relative (5.25) both have the nested pattern, whereas the English multiple subject relative (5.23) and the multiple pied-piping sentence (5.26) have the disjoint pattern.

This now fits directly with the processing difficulties of the different sentences; as predicted, the nested dependencies are hard, the disjoint patterns not so hard. The nested pattern increases in complexity rapidly as the sentences are lengthened, the disjoint pattern remains easy. The sentences with disjoint dependencies keep their flat, rhythmic intonation, which is not found in the sentences with nested dependencies. We now have an explanation for the ease of processing the multiple pied piping sentence type. However long the sentence is made, the dependency pattern remains disjoint; there is never any need to remember unfinished dependencies while forming another association. The processing ease of this sentence, which was inexplicable in terms of a theory building up associations between fillers and gaps and gaps and verbs, falls directly out of an account not using gaps.

There is now no difference between sentences with nested constructions and those with nested dependencies, which is a considerable simplification in theory. It also means that it is possible to simplify Chomsky's definition of nested constructions by removing the word *nonnull*, as there are now no empty categories. There are no filler-gap or gap-governor associations, hence it is impossible for these to be nested. The processor therefore does not need to assume that any properties of displaced elements are associated with their canonical locations, and so needs no mechanism for gap filling.

To summarize, we can replace the table of associations in subsection 5.2.3 with the following table based on filler-governor dependencies, in which the correspondence between nested dependencies and nested constructions is made explicit:

Sentence type	Example	Filler-governor pattern	Construction type
English multiple subject relative	5.10	Disjoint	Non-nested
English multiple object relative	5.11	Nested	Nested
German multiple subject relative	5.14	Nested	Nested
English multiple pied-piping	5.6	Disjoint	Non-nested

5.3 Problems with Alternative Analyses Using Gaps

5.3.1 Gaps and Word-Order Freedom

If the parser uses a grammar with gaps, it obviously has to have some method of identifying where the gap is. There are well-known problems of mislocating 'doubtful gaps' (J.D.Fodor 1978), but these are genuine on-line ambiguities, when it is globally clear where the gap is. Much more serious is the question of word-order freedom. We can know that an element has been displaced, but not know where it has been displaced from. Consider:

(5.27) Who did you phone up?

The point is that since both *you phoned John up* and *you phoned up John* are acceptable, the gap may be sited before or after *up*. A similar example, but in this case using adjuncts, is the following:

(5.28) In which park did Bill meet Tom on Tuesday?

This phenomenon is fairly infrequent in English because of the relative rigidity of the word order, but would be more apparent in a language with clear constituents and word-order freedom within each constituent (since it would be clear from which constituent an element had been extracted, but not from which position within that constituent). We could deal with this by treating one word order as 'basic', or alternatively by assuming that there are multiple canonical forms and the parser has to reconstruct any one of them by means of some predefined strategy (e.g. site the gap as soon as possible). The point of this, however, is that any account making use of gaps has added complications that can only be resolved by making additional assumptions in the processor or the grammar (or both). None of these are needed in the alternative account.

However, it is possible that assuming a 'non-standard' location for gaps may change the pattern of associations, in such a way that a nested pattern can be avoided in sentences such as (5.6). This would give support for the 'non-standard' gap location, but, more crucially, might allow an account using gaps to survive as a competitor to the account proposed in subsection 5.2.5. We shall therefore explore this possibility next, but show that it is unable to capture all the relevant data. There are two possible non-standard gap locations that could be used in the analysis of (5.6), as we shall see.

5.3.2 An Analysis Using Heavy NP Shift

Let us return to sentence (5.6), with its standard filler-gap and gap-governor analysis (5.16) (repeated here as (5.29)):

(5.29) John found the box [in which]$_a$ I [put]$^\alpha$ the tray [on which]$_b$
Mary [placed]$^\beta$ the cake $\emptyset_b^\beta$ $\emptyset_a^\alpha$.

We can reanalyse this in filler-gap terms by arguing that *the tray [on which]$_b$ Mary [placed]$^\beta$ the cake* $\emptyset_b^\beta$ is in fact a heavy NP shifted beyond the final gap $\emptyset_a^\alpha$. This means that this gap is now adjacent to the verb *[put]*$^\alpha$ and hence we have a disjoint dependency pattern; the structure might be argued to be as in (5.30) below:

(5.30) John found the box [in which]$_a$ I [put]$^\alpha$ $\emptyset_a^\alpha$ the tray [on which]$_b$
Mary [placed]$^\beta$ the cake $\emptyset_b^\beta$.

There are at least two arguments against such an analysis. Firstly, we can construct multiple relative sentences using verbs with a ditransitive subcategorization frame like *give*. These verbs crucially do not have an optional heavy-shifted argument configuration; on the pragmatically sensible (heavy-shifted) reading, (5.31) is out:

(5.31) *I gave the book the woman in the heavy winter coat.

Now, (5.32) does not produce any processing difficulty (on the reading where the book is given to the slave, and the slave is sold to the nobleman):

(5.32) John wrote the book which Mary gave the slave who Tom sold
the nobleman.

But this sentence has the same problems as (5.6), except that the heavy-shifted reading analogous to (5.30) is now impossible, i.e. the pattern of fillers and gaps must be as in (5.34) rather than (5.33):

(5.33) *John wrote the book [which]$_a$ Mary [gave]$^\alpha$ the slave $\emptyset_a^\alpha$ [who]$_b$
Tom [sold]$^\beta$ the nobleman $\emptyset_b^\beta$.

(5.34) John wrote the book [which]$_a$ Mary [gave]$^\alpha$ the slave [who]$_b$
Tom [sold]$^\beta$ the nobleman $\emptyset_b^\beta$ $\emptyset_a^\alpha$.

Both the filler-gap and the gap-governor associations are nested, and the sentence type can be extended in exactly the same way as (5.21) (although the results are pragmatically rather strange!):

(5.35) John wrote the book which Mary gave the slave who Tom sold the senior slave who Bill sold the slavemaster who Sue sold the king.

The intended interpretation does not become hard to parse, and the intonation pattern is still that of a multiple subject relative sentence like (5.10). So a theory assuming that gaps are used in processing cannot avoid a nesting pattern of dependencies by this method. (The use of double object verbs where heavy-shifting is impossible was initially avoided simply because the existence of the second reading may add confusion.)

Second, heavy-shifting is itself usually thought of as leaving a gap (movement analyses go back to Ross (1967); see also Postal (1974)).[6] Considering just the filler-gap associations, we obtain the following heavy-NP-shifted analysis for (5.6):

(5.36) John found the box [in which]$_a$ I put $\emptyset_c$ $\emptyset_a$ [the tray [on which]$_b$ Mary placed the cake $\emptyset_b$]$_c$.

The dependency pattern for this is *acac*, with *bb* in some sense contained within the second *c*. Therefore it is clear that the association between the *c*s cannot be made until the *b*s have been processed. *b* and *c* must therefore be held in memory at the same time. Let us extend the sentence with one recursion:

(5.37) John found the box [in which]$_a$ I put $\emptyset_c$ $\emptyset_a$ [the tray [on which]$_b$ Mary placed $\emptyset_e$ $\emptyset_b$ [the dish [from which]$_d$ Bill took the cake $\emptyset_d$]$_e$]$_c$.

Here the structure gets more complicated, and crucially, the maximum number of elements that have to be remembered because the association cannot be made, is one

---

[6] We can see this as based on a particular assumption about the nature of gaps, that they are used to represent the canonical or underlying location of a displaced element. This is standard, but it is possible to argue that gaps are not implicated in bounded movement; assuming that heavy NP shift is bounded, we can simply postulate an extra phrase-structure rule, perhaps VP ⟶ V PP NP, allowed in restricted cases where the NP is 'heavy' (or whatever is the correct characterization).

greater than for (5.36) (here, at *Bill*, *d*, *e* and *b* have to be held in memory at the same time). It is clearly true that with more complicated sentences, more elements have to be remembered together, because the associations cannot be resolved. It is of course irrelevant whether the gap is to the left or the right of the filler. The heavy-shifting account therefore fails (assuming that gaps are involved in bounded movement). This is good support for the stronger argument against the analysis with heavy NP shift presented first.

5.3.3 An Analysis Using Extraposition from NP

It is conceivable that the processor avoids nested associations in sentences such as (5.6) by assuming an analysis where the relative clause is extraposed past the gap:

(5.38) John found the box $[\text{in which}]_a$ I $[\text{put}]^\alpha$ the tray $\emptyset^\alpha_a$ $[\text{on which}]_b$ Mary $[\text{placed}]^\beta$ the cake $\emptyset^\beta_b$.

Again, let us consider merely the filler-gap associations:

(5.39) John found the box $[\text{in which}]_a$ I put the tray $\emptyset_a$ $[\text{on which}]_b$ Mary placed the cake $\emptyset_b$.

This gives us a disjoint pattern of dependencies, as with the heavy-NP-shifted analysis above. It also gives rise to similar problems.

The first argument again involves ditransitive verbs like *give*. We note that extraposition from the first NP is always highly marked, and is especially bad with *that*-less relatives:

(5.40) ?*Mary gave the slave the punishment Tom had always treated well.

Therefore the extrapositional analysis is very bad in the intended interpretation of (5.41):

(5.41) John wrote the book Mary gave the slave Tom sold the nobleman.

But this sentence is not difficult to process. Since extraposition is very bad, the only way that the gap-based analysis could process this sentence would be to have nested dependencies. But again this sentence can be indefinitely extended, with even intonation. Clearly an account using gaps cannot escape the problem this way either. A similar point is that it is probably impossible to extrapose non-restrictive relatives:

(5.42) ?*Mary gave Bill the punishment, who Tom had always treated well.

Here the relative clause is to be taken as a non-restrictive modifier of *Bill.* Therefore again the intended interpretation of (5.43) should be very bad with the extrapositional analysis, but it is not:

(5.43) John wrote the book, which Mary gave Bill, who Tom sold the nobleman.

Hudson (1990; pc) makes a related argument against an extrapositional analysis. He claims that the modifier *capable of ...* cannot be extraposed (for obscure reasons). This seems clearly correct if the extraposition is from one NP past another NP:

(5.44) *We give every student a prize capable of answering all the questions.

Because a heavy-shifted analysis is not possible either, a relative clause using this construction must have a structure with the gap at the end:

(5.45) [which]$_a$ we give every student capable of answering all the questions $\emptyset_a$

A sentence like (5.46) is difficult to process, because the first post-verbal argument is so long and unwieldy:

(5.46) We give every student capable of answering every single tricky question on the details of post-Barriers theories about the interaction between functional categories and word order a prize.

Therefore extracting the second argument and replacing it with an empty category would presumably produce the same effect, but it clearly does not:

(5.47) This is the prize $[\text{which}]_a$ we give every student capable of answering every single tricky question on the details of post-Barriers theories about the interaction between functional categories and word order $\emptyset_a$.

This does not fit easily with an account with empty categories. But this lack of processing difficulty is predicted if the relative pronoun *which* is associated directly with the verb *give*. There is also an extremely strong intuition that we know that a prize is given, well before the end of the sentence is reached. This effect is apparent here because of the length of the NP following *give*.

There is another argument against this analysis, again analogous to that used for heavy NP shift. We assume that extraposition leaves a gap at the canonical position.[7] It seems clear that gaps must be left in adjunct as well as argument positions, since for instance there has to be some kind of semantic association between the fronted adjunct and the embedded clause in (5.48):

(5.48) How enthusiastically does Bill think John plays the violin?

Adding these further filler-gap associations to (5.39) gives (5.49):

(5.49) John found the box $[\text{in which}]_a$ I put the tray $\emptyset_c$ $\emptyset_a$ $[[\text{on which}]_b$
Mary placed the cake $\emptyset_b]_c$.

Just as with (5.36), the dependency pattern is *acac*, with *bb* contained within the second *c*, and so the argument proceeds as before. The analogy persists if we extend the sentence:

(5.50) John found the box $[\text{in which}]_a$ I put the tray $\emptyset_c$ $\emptyset_a$ $[[\text{on which}]_b$
Mary placed the dish $\emptyset_e$ $\emptyset_b$ $[[\text{from which}]_d$ Bill took the cake
$\emptyset_d]_e]_c$.

---

[7] As in footnote 6, it is possible to argue that there is no gap involved in extraposition, but this appears more bizarre than for heavy NP shift because this cannot be achieved by the addition of alternative phrase-structure rules for expanding the VP.

The pattern of filler-gap associations is as in (5.37). Hence, assuming gaps are implicated in extraposition, it is impossible to avoid nesting associations in this way, contrary to the processing evidence. This again supports the argument, based on ditransitives, that if there are gaps the filler-gap association patterns must be nested. Because the processing evidence militates against nested associations, there can be no psychologically real gaps.

We can now draw the conclusion that we cannot retain empty categories as psychologically real entities by postulating them in non-canonical positions; there is no pattern of gap locations in sentences such as (5.6) that is both linguistically motivated and psychologically plausible. The evidence we have presented supports the simpler theory that processing involves no gaps and hence needs no strategy for gap location.

5.4 Representation in Dependency Constituency and the Lambek Calculus

5.4.1 Introduction

I have outlined the formation of filler-governor dependencies in multiple pied piping sentences, but have not considered how the above schematic treatment corresponds to the account based on the Lambek Calculus. Let us first see how the above data does not distinguish adjuncts and arguments. The multiple pied piping examples make use both of verbs that would standardly be regarded as subcategorizing for two post-verbal arguments, and verbs that would be treated as having only one post-verbal argument plus an adjunct. In (5.21), repeated below, *put* subcategorizes for a PP, but *poured* does not:

(5.51) John found the box in which I put the tray on which Mary placed the dish from which Bill took the saucer on which Sue stood the cup into which Fred poured the tea.

The purported position of the empty category is of course the same. Interestingly, it does not seem to matter which type of verb is used. Neither causes nesting. This is obviously compatible with there being no clear dichotomy between adjuncts and arguments. It is also predicted by the analysis in terms of dependency constituents, because a fronted adjunct forms a dependency constituent at the verb in exactly the same way that a fronted argument does.

The coindexation relation establishes what we have called a filler-governor dependency, but in terms of the present theory, this is simply equivalent to an unbounded dependency in the loose sense introduced in chapter 4. The purpose of the coindexation notation was to allow a direct contrast between the analysis with empty categories and the analysis without empty categories, and what I have shown is that the analysis with empty categories leads to nested patterns of associations that cannot adequately represent actual processes involved in language comprehension. The coindexation without empty categories can be seen as defining the point at which a single dependency constituent is formed. The next section shows how the formation of dependency constituents for multiple pied piping sentences is captured by the natural deduction axiomatization of **L**, and how it respects the Minimal Incrementality Condition. This is then contrasted with a possible account mentioned briefly in chapter 4, where unbounded dependencies are analysed in flexible categorial terms but with empty categories. In chapter 4 I merely noted that this account was unparsimonious, but we can now show that it fails to respect the MIC and can therefore be discarded.

5.4.2 Modelling Multiple Pied Piping

As might be expected, any relative clause is a dependency constituent, for instance the underlined fragments of the following two constructions. There is no need to abstract over a functor in either case:[8]

(5.52) the box <u>which John knelt on</u>

(5.53) the box <u>on which John knelt</u>

---

[8] As above, phrases such as *on which* and *in which* are treated as single lexical items.

(5.54)

$$\dfrac{\dfrac{\textbf{which}}{\mathrm{PoN/(S/NP)}} \qquad \dfrac{\dfrac{\dfrac{\textbf{John}}{\mathrm{NP}} \qquad \dfrac{\dfrac{\textbf{knelt}}{\mathrm{(NP\backslash S)/PP}} \qquad \dfrac{\dfrac{\textbf{on}}{\mathrm{PP/NP}} \qquad [\mathrm{NP}]^1}{\mathrm{PP}}\,\mathrm{/E}}{\mathrm{NP\backslash S}}\,\mathrm{/E}}{\mathrm{S}}\,\mathrm{\backslash E}}{\mathrm{S/NP}}\,\mathrm{/I}^1}{\mathrm{PoN}}\,\mathrm{/E}$$

$\textbf{which } (\lambda x^{\mathrm{NP}}[\textbf{knelt (on } x\textbf{) john}])$

(5.55)

$$\dfrac{\dfrac{\textbf{on which}}{\mathrm{PoN/(S/PP)}} \qquad \dfrac{\dfrac{\dfrac{\textbf{John}}{\mathrm{NP}} \qquad \dfrac{\dfrac{\textbf{knelt}}{\mathrm{(NP\backslash S)/PP}} \qquad [\mathrm{PP}]^1}{\mathrm{NP\backslash S}}\,\mathrm{/E}}{\mathrm{S}}\,\mathrm{\backslash E}}{\mathrm{S/PP}}\,\mathrm{/I}^1}{\mathrm{PoN}}\,\mathrm{/E}$$

$\textbf{on-which } (\lambda x^{\mathrm{PP}}[\textbf{knelt } x \textbf{ john}])$

On the other hand, some fragments of relative clauses are not dependency constituents, such as the two underlined below:

(5.56) the box <u>which John</u> knelt on

(5.57) the box <u>which John knelt</u> on

(5.58)

$$\dfrac{\dfrac{\dfrac{\textbf{which}}{\mathrm{PoN/(S/NP)}} \qquad \dfrac{\dfrac{\dfrac{\textbf{John}}{\mathrm{NP}} \qquad \dfrac{[\mathrm{(NP\backslash S)/NP}]^2 \qquad [\mathrm{NP}]^1}{\mathrm{NP\backslash S}}\,\mathrm{/E}}{\mathrm{S}}\,\mathrm{\backslash E}}{\mathrm{S/NP}}\,\mathrm{/I}^1}{\mathrm{PoN}}\,\mathrm{/E}}{\mathrm{PoN/((NP\backslash S)/NP)}}\,\mathrm{/I}^2$$

$\lambda f^{\mathrm{NP}\to(\mathrm{NP}\to\mathrm{S})}[\textbf{which } (\lambda x^{\mathrm{NP}}[f\ x\ \textbf{john}])]$

(5.59)

$$\dfrac{\dfrac{\dfrac{\textbf{which}}{\mathrm{PoN/(S/NP)}} \quad \dfrac{\dfrac{\dfrac{\textbf{John}}{\mathrm{NP}} \quad \dfrac{\dfrac{\textbf{knelt}}{\mathrm{(NP\backslash S)/PP}} \quad \dfrac{[\mathrm{PP/NP}]^2 \quad [\mathrm{NP}]^1}{\mathrm{PP}}{\scriptstyle /E}}{\mathrm{NP\backslash S}}{\scriptstyle /E}}{\mathrm{S}}{\scriptstyle \backslash E}}{\mathrm{S/NP}}{\scriptstyle /I^1}}{\mathrm{PoN}}{\scriptstyle /E}}{\mathrm{PoN/(PP/NP)}}{\scriptstyle /I^2}$$

$\lambda g^{\mathrm{NP}\rightarrow\mathrm{PP}}[\textbf{which}\ (\lambda x^{\mathrm{NP}}[\textbf{knelt}\ (g\ x)\ \textbf{john}])]$

Neither derivation is dependency-preserving, since each case involves abstraction over a functor ((NP\S)/NP and PP/NP). This seems to be precisely the intuition we want to capture, because *which* associates with *on*, not with either *John* or *knelt*. Likewise, the underlined phrase in (5.60) is not a dependency constituent, because we have to abstract over the functor PP/NP:

(5.60) the tray which Mary placed the cake on

(5.61)

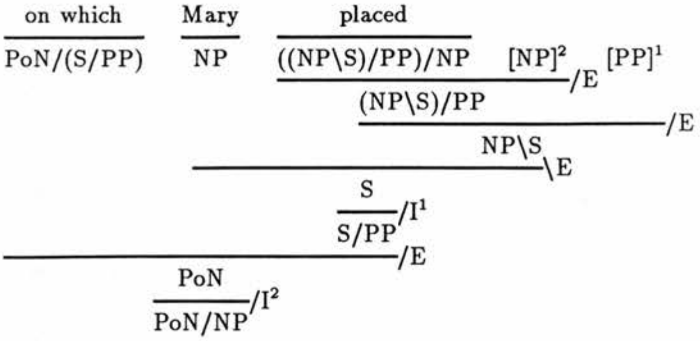
$$\dfrac{\dfrac{\dfrac{\dfrac{\textbf{which}}{\mathrm{PoN/(S/NP)}} \quad \dfrac{\dfrac{\dfrac{\textbf{Mary}}{\mathrm{NP}} \quad \dfrac{\dfrac{\dfrac{\textbf{placed}}{\mathrm{((NP\backslash S)/PP)/NP}} \quad [\mathrm{NP}]^3}{\mathrm{(NP\backslash S)/PP}}{\scriptstyle /E} \quad \dfrac{[\mathrm{PP/NP}]^2 \quad [\mathrm{NP}]^1}{\mathrm{PP}}{\scriptstyle /E}}{\mathrm{NP\backslash S}}{\scriptstyle /E}}{\mathrm{S}}{\scriptstyle \backslash E}}{\mathrm{S/NP}}{\scriptstyle /I^1}}{\mathrm{PoN}}{\scriptstyle /E}}{\mathrm{PoN/(PP/NP)}}{\scriptstyle /I^2}}{\mathrm{(PoN/(PP/NP))/NP}}{\scriptstyle /I^3}$$

$\lambda y^{\mathrm{NP}}\lambda g^{\mathrm{NP}\rightarrow\mathrm{PP}}[\textbf{which}\ (\lambda x^{\mathrm{NP}}[\textbf{placed}\ y\ (g\ x)\ \textbf{mary}])]$

In contrast, the underlined phrase in (5.62) is a dependency constituent, because there is no abstraction over a functor:

(5.62) the tray on which Mary placed the cake

(5.63)

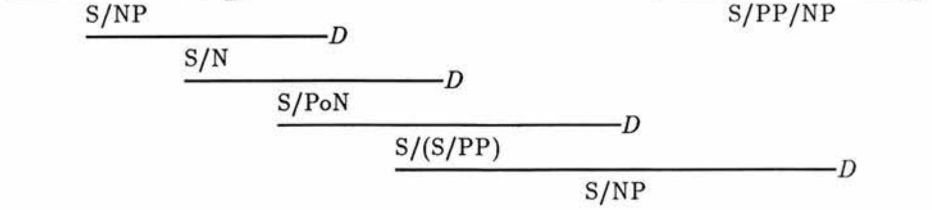


λy^{NP}[**on-which** (λx^{PP}[**placed** y x **mary**])]

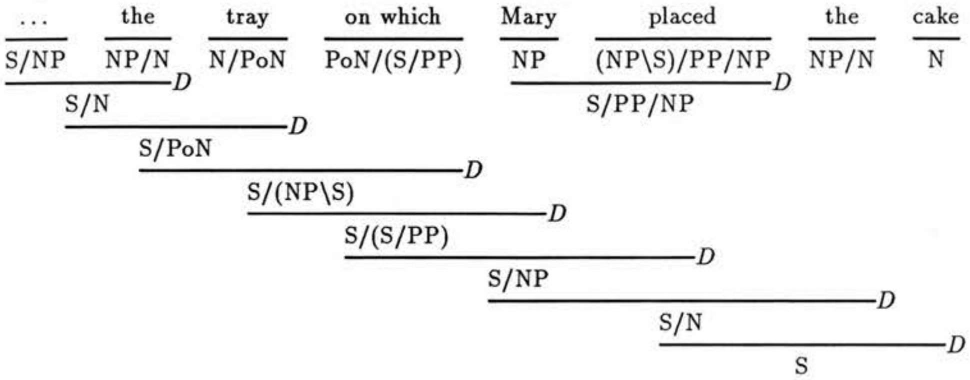
Again, *on which* fills an argument role of the verb *placed* in (5.62), whereas in (5.60) the relative pronoun *which* fills an argument role not of *placed* itself but of the prepositional argument of *placed* (which is outside the string). Therefore the definition of dependency constituent correctly captures what appears to be a rather subtle distinction.

(5.62) shows that a dependency constituent can be formed between the extracted argument and the verb even if the extracted argument would standardly be the second post-verbal argument. In other words, the definition of dependency constituent pays no attention to the 'canonical' location of arguments. This means that the notion of dependency constituent does not serve an account with empty categories, but instead can be used to model the process of filler-governor dependency formation. Let us return to the description of the incremental processing of multiple subject relatives in chapter 3. It was shown that (unlike the simplest theory derived from phrase structure grammar) an account based on dependency constituency did not cause a build-up of complexity as the construction was recursed. Therefore the account satisfied the Minimal Incrementality Condition. Similar results are found with multiple pied piping sentences, contrary to any conceivable account that uses empty categories. Consider the incremental processing of (5.6):

(5.64)



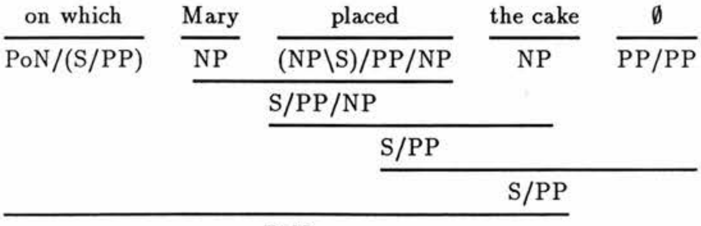
(5.65)



This shows the dependency-preserving incremental derivation. Note that the only points where we lose left-associativity are the subjects of the verbs, which do not form a dependency constituent with their left context. But a single dependency constituent is formed immediately the verb is reached, and crucially the number of dependency constituents that has to be remembered does not increase as the construction is recursed. Given that this sentence does not become harder to process as it is extended, the minimal incrementality condition is satisfied. Hence this account correctly captures the processing of multiple pied piping sentences.

Note however that an account using flexible categorial grammar and empty categories, as discussed briefly in chapter 4, would not satisfy the MIC, because it would be impossible to combine the relative pronoun and the verb until the empty category had been found, and this empty category would be at the end of the sentence. Assuming the empty category is of type PP/PP, the incremental derivation of *on which Mary placed the cake* would be:

(5.66)



The crucial point is that the relative pronoun cannot be combined with the verb to form a single dependency constituent, but instead must wait until the verb has combined with the empty category. This is because the empty category, of type PP/PP, would otherwise be left unable to combine with anything. This means that the relative pronoun can only combine with the verb plus its first post-verbal argument. Hence recursion of the construction on this post-verbal argument will lead to a steady increase in the number of dependency constituents to be remembered. Because such a construction is not hard to process, the MIC is violated, and so the use of empty categories within an otherwise incrementally valid categorial grammar will cause that validity to be lost. A similar problem would occur if the empty category were given the type PP, as was also suggested above.

We can contrast (5.62) with the preposition stranding example (5.60) above. In fact it is simply a nested construction, the only difference from, for instance, (3.73) being that the embedding comes between the verb and the preposition, rather than the subject and the verb:

(5.67) John found the box which I put the tray which Mary placed the cake on in.

In this case, we have to remember three dependency constituents, *John found the box which*, *I put the tray which* and *Mary placed the cake*, and it is only when we reach the prepositions that we can combine the dependency constituents together. This fits in with the fact that recursing this construction makes the resultant sentence very hard to process. So again, the present formulation directly captures the major processing difference between minimal pairs of sentences.[9]

5.5 Summary

In this chapter we have considered the difference between an account of processing that makes use of empty categories and one that does not, and have seen that using empty categories makes the wrong prediction about the processing of a particular class of sentences involving multiple pied piping. The conclusion is that the sentence processor does not postulate the existence of empty categories, and so a grammar obeying the Strong Competence Hypothesis must avoid empty categories. We have also seen that the

---

[9] Note that introducing empty categories would in this case have no lasting effect on the pattern of dependency constituents, because the empty categories, of type NP/NP, would be adjacent to the prepositions, of type PP/NP, and they could be combined immediately.

use of the Lambek Calculus and of the notion of dependency constituency does model the lack of difficulty we have processing these sentences compared with, for instance, nested constructions.

Chapter 6

Sentence Processing

6.1 Introduction

This chapter goes into more detail about the way in which we process sentences and considers the relation between combinatorial operations and other components of the sentence processing mechanism. One concern is how we resolve cases of ambiguity. It is claimed that there is no need to differentiate the mechanisms that we use for processing unbounded dependencies from those that we use in other parts of sentence comprehension. This follows from the assumption that there are no empty categories and no special mechanism associated with them, but it is also possible to see how experimental evidence supports this proposal. The chapter then discusses another proposed empty category, NP-trace, specifically in relation to its assumed use in passives, and shows how its psychological reality is subject to the same disconfirmation by arguments relating to nesting that were used for *wh*-trace in chapter 5. It therefore argues that experimental evidence for NP-trace must be reinterpreted. Finally, the chapter considers the power of the reactivation technique in determining the initial stages in post-lexical aspects of sentence comprehension, and suggests some possible applications and revisions.

6.2 Ambiguity in Sentence Processing

I have assumed that sentence processing involves the construction and interpretation of dependency constituents. The Dependency Interpretation Hypothesis holds that all and only dependency constituents that can be constructed are constructed and interpreted immediately. In this section I shall consider sentence processing from a less idealized perspective, and discuss how we resolve ambiguities. The fundamental division proposed is between those ambiguities that involve a choice of possible combinations, and those that involve a choice over whether to combine or not. The main proposal is that we initially perform some combination if any is in fact possible.

6.2.1 Representation of Choice of Analysis

The DIH claims that all possible dependency constituents are constructed. We have assumed an idealized context in which each string has only one reading, and therefore exactly one division into MDCs.[1] But of course it is possible for a sentence to have more than one reading. We shall term this a *global ambiguity*. It is also possible for a sentence to have only one reading, but for a substring of this sentence to have more than one reading. We shall term this a *local ambiguity*. Any word with two types, for instance, is of course locally ambiguous. But the most important type of local ambiguity for our purposes is that of a string beginning with the first word of the sentence.

The processor could use a number of strategies in cases of ambiguity. At one extreme, we could simply generate all readings of a string, and compute all possible analyses together. We can call this *naive parallelism*. But naive parallelism is of course impossible, because there is in general no limit to the number of readings for an arbitrary sentence, and so allowing all competing interpretations to coexist must at some point exceed memory resources. For instance, it is possible to compound strings of nouns such as *water meter cover adjustment screw*, in very many syntactically distinct ways (Briscoe 1987), but we cannot retain all the interpretations in parallel. The processing of locally ambiguous strings could therefore proceed in two ways. We could use a simple *serial* model, where only one reading is actively processed (though it may be possible to reconstruct other readings) and a *ranked parallel* model, where only some subset of readings are processed, some in a more prominent and some in a less prominent way.[2] A possible complication is that different readings may be remembered in part with a shared representation, using some form of underspecification. I shall assume that this is

---

[1] We strictly mean more than one *description*, because this string may not be a dependency constituent, and therefore, assuming the DIH, not have a (unified) reading as such. But we shall loosely use the term *reading*.

[2] These two characteristics are not strictly inseparable. But the 'garden path' data below strongly suggests that even when there are only two readings, one is forgotten or backgrounded.

not relevant to ambiguities of dependency constituency, though it may be used in details of interpretation, such as with scope ambiguities or polysemy. Given the difficulty that we have processing nested constructions, it is extremely unlikely that we can process multiple readings together easily, so we can assume that any ranked parallel model will be limited to a small number of readings. In what follows we shall assume a serial model of sentence processing (with an important qualification discussed below), but nothing rests on this assumption. We shall talk of *choosing a reading*, but it is possible to replace *choosing* with *foregrounding*.

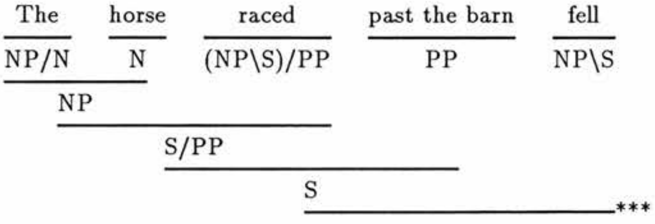
The DIH holds that the processor gives all and only dependency constituents an interpretation. This apparently requires a naive parallel model of sentence processing. In fact, this definition presupposed that any string had only one reading, and therefore that there was no local (or global) ambiguity. Let us therefore redefine the DIH:

Hypothesis 10 *If the processor gives a string a reading X, then it interprets all and only the dependency constituents of X.*

The definition now only applies to sentences that we interpret (and therefore not ones that we fail to attend to, for instance). It of course has no implications for which reading we choose in cases of ambiguity, or indeed for whether strings are given readings in parallel or in serial.

Let us first see how we can use the dependency-preserving subset of **L** to describe ambiguity. This can be illustrated with an example of a classic 'garden path' effect, adapted from Bever (1970), where a word is ambiguous between two types and the wrong type is chosen. We discover that we have chosen the wrong entry when we find that we the sentence is not globally ambiguous, and the reading that we have chosen is not of type S (and has an inappropriate semantics). For instance, in *The horse raced past the barn fell*, we pursue the following (simplified) analysis:

(6.1)



We presumably decide at this point that this analysis is wrong, but we do not know for sure unless we know that we have reached the end of the sentence (for instance, the continuation ... *and got up again* is grammatical). The question of what makes us discontinue an analysis has surprisingly been largely ignored. I shall say nothing about it here.[3]

At the point of local ambiguity, we chose the wrong lexical entry for *raced*. When we realise this, we check our assumptions. We find that we treated *raced* as a simple past tense verb and gave it the type (NP\S)/PP. We therefore instead assume that *raced* is a past participle with type PoN/PP, and also give *horse* the type N/PoN (derived from the type (PrN*\N)/PoN*). We can then interpret this sentence correctly. But this process of reanalysis is costly, and therefore we notice the comprehension difficulty and feel the need to backtrack. We perform the corrected analysis below:

$$\begin{array}{llllll}
(6.2) & \underline{\text{The}} & \underline{\text{horse}} & \underline{\text{raced}} & \underline{\text{past the barn}} & \underline{\text{fell}} \\
 & \text{NP/N} & \text{N/PoN} & \text{PoN/PP} & \text{PP} & \text{NP}\backslash\text{S} \\
\cline{2-3}
 & \multicolumn{2}{c}{\text{NP/PoN}} & & & \\
\cline{2-4}
 & \multicolumn{3}{c}{\text{NP/PP}} & & \\
\cline{3-5}
 & & \multicolumn{3}{c}{\text{NP}} & \\
\cline{4-6}
 & & & \multicolumn{3}{c}{\text{S}}
\end{array}$$

Much research into sentence processing has been concerned with the grounds on which we choose an analysis. This method of representation indicates the point at which difficulty arises directly. The fact that we cannot combine *the horse raced past the barn* and *fell*, under the analysis in (6.1), is indicated by the the fact that the sequence of types S NP\S cannot be combined by dependency-preserving operations. This is an indication of the usefulness of this method of representing the state of the parser at any stage in language comprehension.[4] Therefore it is possible to discuss the linguistic properties of garden paths directly.

There is a considerable debate about whether the parser pays attention to the possible interpretation of strings like *the horse raced* in resolving the local ambiguity. The debate

---

[3] Unless we know that we have reached the end of the sentence, it is very difficult to think of any string that has a grammatical continuation under one reading, which does not have a *possible* grammatical continuation under the other reading. But presumably many of these continuations will be unprocessable, which will make us decide to change our analysis.

[4] Of course we do not need to mentally represent the whole derivation, but rather only the output (in terms of the characterization of dependency constituents) at any stage. The combinations correspond to operations of the parser. This distinguishes it from a phrase structure tree representation, at least some part of which presumably has to be represented at any stage in parsing.

is described in Altmann (1989): see Steedman and Altmann (1989) for arguments for this position, and Clifton and Ferreira (1989) for arguments against. Both accounts assume that there is an informationally-encapsulated sentence processing module (J.A.Fodor 1983), and hence plausible interpretations cannot guide the operations of this module (contra Marslen-Wilson and Tyler 1980). We shall also make this assumption.

6.2.2 Choosing between Combinations

The account described in Steedman and Altmann was proposed in Crain and Steedman (1985) and Altmann and Steedman (1988) (see also Crain (1980), Altmann (1987; 1988), Steedman (1989)). They assume that in cases of local ambiguity, both (or all) readings are generated in parallel, and a decision is made between the readings on the basis of which interpretation is more plausible in the context of the utterance. In other words, the compatibility of each reading with other relevant beliefs is assessed in order to decide which is appropriate. This assumes a short-lived parallelism, but only in the sense that all possible outputs from one module are inputed into the next module, not that complete multiple representations are processed in parallel. The standard example, from Crain and Steedman, is based on the lexical ambiguity of *that* between being a a complement clause introducer, as in (6.3), and being (restrictive) relative clause introducer, as in (6.4):

(6.3) The psychologist told the woman that he was having trouble
with her husband.

(6.4) The psychologist told the woman that he was having trouble
with to leave her husband.

These sentences are globally disambiguated, but are locally ambiguous after the word *that*. Crain and Steedman describe an experiment where these sentences are paired with a context sentence that introduces either one or two women into the discourse. They argued that subjects assume that *that* is more likely to be a complement clause introducer if the context contains one woman, because a relative clause beginning *the woman that* requires there to be more than one woman in the context. Likewise, if more than one woman is in the context, we would not know which woman the simple NP *the* *woman* referred to, and so we assume that *that* is more likely to be a relative clause introducer.

Under either analysis, the string *the psychologist told the woman that* is a dependency constituent. If *that* introduces a relative clause, the dependency constituent has type (S/VPinf)/(S/NP) (assuming the type VPinf for an infinitive) and meaning $\lambda y^{VPinf} \lambda x^{NP \rightarrow S}$ [**told (the (woman (that** x**)))** y **(the psychologist)**]. If *that* introduces a complement clause, the dependency constituent has type S/S and meaning λx^{S} [**told (the woman) (that** x**) (the psychologist)**]. Which reading we choose will be determined by the context, and if the context is misleading, we will garden path when we read the rest of the sentence. Both garden path derivations are given below:

(6.5) The psychologist told the woman that — he was having trouble with her husband
(S/VPinf)/(S/NP) — S
***

(6.6) The psychologist told the woman that — he was having trouble with — to leave her husband
S/S — S/NP — VPinf
S/NP
***

The decision between the analyses can be made by comparing the two possible meanings to *the psychologist told the woman that* in parallel, and assessing their compatibility with the context. We then reject one of the meanings and therefore the associated syntactic type (as necessitated by the rule-to-rule hypothesis). This short-lived parallelism is obviously very different from true parallelism, where more than one analysis survives while later operations take place. Its purpose is purely to give us the opportunity to choose which analysis to pursue.

However, it is possible that the decision between the analyses is not initially made on semantic grounds at all, but is rather done, as Clifton and Ferreira argue, by choosing one reading for non-semantic reasons and then sometimes going back on that decision if the continuation is impossible or implausible. This requires a more complicated but quite consistent model. I shall however assume a model with this short-lived parallelism, but nothing important follows from this. Crain and Steedman assume that the decision between analyses is made on semantic grounds, and their experiment gives some evidence for this. It is possible that non-semantic factors, such as frequency of use of a construction or (in spoken form) intonation, also influence our choice of reading. But given that the semantic information is available at the same time as the syntactic information, it seems very likely that this semantic information is relevant to the decision, and so we would expect that all readings of a dependency constituent are rapidly compared and that plausibility with respect to context is at least one relevant factor. If the wrong reading is chosen, a garden path occurs.

6.2.3 The Immediate Combination Hypothesis

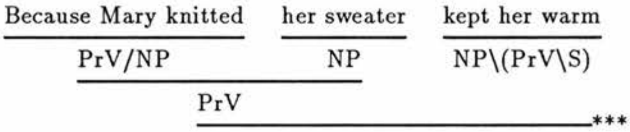
The above discussion considered sentences where both readings form dependency constituents. This changes when there is a local ambiguity over whether to form a dependency constituent or not. An example of this is with ambiguous transitive/intransitive verbs, for instance *knitted*:

(6.7) Because Maria knitted her sweater it kept her warm.

(6.8) Because Maria knitted her sweater kept her warm.

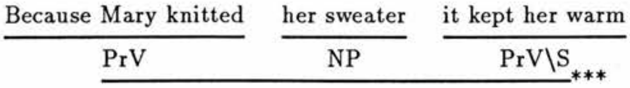
If *knitted* is assumed to be transitive, we will parse (6.7) without problem, but will garden path on (6.8):

(6.9)



If we assume *knitted* to be intransitive, we get the right reading for (6.8), but will garden path on (6.7), because we get an sequence of types PrV NP PrV\S that cannot be combined by dependency-preserving operations.

(6.10)



There is good reason to assume that the semantics of the developing representation of the sentence is not relevant in the initial choice of reading. If (6.7) and (6.8) are read, without punctuation, (6.8) causes a garden path, but (6.7) does not appear to.

Assuming that our intuitions are accurate, we appear to treat *knitted* as transitive rather than intransitive. We are not conscious of making two interpretations in parallel, and then deciding between them on the basis of plausibility with respect to the context. The reason for this, in the terms of the present theory, is that one reading forms a single dependency constituent *(Because Mary knitted her sweater)*, but the other requires two dependency constituents, *(Because Mary knitted) (her sweater)*. Therefore the transitive reading has the meaning **because** (**knitted** (**the sweater**) **mary**), but the intransitive reading requires the two dependency constituents to be kept semantically distinct, as **because** (**knitted mary**) and **the sweater**. These meanings cannot be compared directly, because there is not a unified interpretation of the intransitive reading. There can therefore not be a direct comparison between the meanings of two strings when one string is a dependency constituent and the other is not.

How do we decide which reading to assume? The above garden path effect showed that in contexts where there is no extra information available, we choose the transitive reading. We shall therefore propose a general principle, the *immediate combination hypothesis* or *ICH*:

Hypothesis 11 *Initially resolve a local ambiguity in favour of a reading that involves fewest dependency constituents.*

This includes part of Frazier's (1978) principle of 'late closure', which claims that we attach each new word into the current phrase if possible. The difference is that a dependency constituent is much wider than a phrase, so for instance, late closure predicts that *John said that Mary arrived yesterday* will be interpreted as having Mary arriving yesterday, not John speaking yesterday. Because in both cases a single dependency constituent is formed, the ICH makes no claim as to which reading is preferred. Therefore this decision needs to be made by comparing the readings in a similar way to (6.3) and (6.4) above. The principle has an obvious processing motivation in that making a combination reduces the number of dependency constituents that have to be remembered. This is reminiscent of phrase structural arguments for late closure, and the associated principle of minimal attachment (Wanner 1987).

Revisions to the ICH

There has been much experimental work that is relevant to the evaluation of the ICH as a hypothesis. There is no doubt that we do garden path on (6.8) but not on (6.7) when the sentence is read without punctuation and is not embedded in a larger discourse. But this does not constitute proof that the ICH holds in general, or that it captures all stages in processing. This section argues that prosodic information may make the listener ignore the ICH on occasion, but apart from this possibility, the ICH is always obeyed. However, the ICH refers only to the initial attempt at resolving an ambiguity, and there will be many occasions on which the initial choice of analysis is rapidly revised.

We can be certain that the plausibility of a combination cannot affect whether we make the combination or not. This is purely a logical point, given the assumption of the rule-to-rule hypothesis. The only way that we can tell that the meaning of combination is implausible is by making that combination. Of course that combination can be very short-lived and thus will constitute a minor garden path. But the combination must happen initially.

On the other hand it may be possible to influence a word's assumed type. The ICH holds that a type which allows immediate combination will be preferred over one that does not. Here, as in Frazier's example, the obvious case is a preference for transitivity over intransitivity in an ambiguous verb. One possible way to attempt to reverse this preference is to consider a verb which is far more commonly used intransitively. However, Mitchell (1987) describes an experiment suggesting that there is a general assumption of transitivity that cannot be overriden, at least in some experimental conditions:

(6.11) After the child visited the doctor prescribed a course of injections.

(6.12) After the child sneezed the doctor prescribed a course of injections.

Subjects were presented with the sentence in two parts, broken after *doctor*. They took longer to read the first presentation in (6.12) than (6.11). His interpretation of this is that they initially assume that *sneezed* is transitive, and that *the doctor* is its object, but that this decision is quickly checked and rejected. (We assume that *sneeze* has a rare transitive reading, which can only be used with cognate objects, as in *he sneezed a big* *sneeze.*) The garden path in (6.11) takes longer to correct. This experiment shows that at least in some experimental conditions, a very strong preference toward intransitivity is not enough to overrule the ICH, though the ICH analysis may be very short-lived.

So verb preferences, and by implication other preferences, may not be relevant in the initial construction of the dependency constituent. But it is also possible that type assignments may be resolved in ways that contradict the ICH, through prior context. In this case, semantics is relevant, but it is not the semantics of the actual combination. This possibility has been addressed by Stowe (1989), who considers the possible importance of thematic information. The subject of a verb may give considerable information on whether the verb is likely to be transitive or intransitive. For instance, with *stopped*, an inanimate subject strongly suggests that the verb is being used intransitively. She contrasted (6.13) and (6.14) below using a grammaticality judgement task (and the results have been replicated using self-paced reading by Stowe and Cupples 1989):

(6.13) Before the police stopped the driver was already getting nervous.

(6.14) Before the truck stopped the driver was already getting nervous.

Trucks do not normally act as agents, and so the transitive reading of (6.14) is very unlikely. Stowe found evidence that subjects garden path in (6.13), which suggests that they assume that *stopped* is transitive, but do not garden path in (6.14), which she argues to be because the thematic information tells us that *stopped* must have the intransitive reading. This suggests that prior context can make us override the ICH.

It does seem rather unlikely that intransitivity can be required by contextual plausibility but not by very strong verb preferences. For instance, the sentence *the truck stopped the flow of traffic* is not anomalous, even though an inanimate object is serving as a kind of agent. In fact, there is evidence that Stowe's result taps into a later stage in processing. The claim is that we still regard *stop* as transitive initially, but the anomaly of the combination causes us to realise that we are garden pathing quite quickly. In another experiment, Stowe found some evidence that processing was slower when the main clause subject, such as *the silence*, would be an implausible object for *the truck stopped.* Mitchell (1989) argues that this suggests that there is an early stage in processing when the thematic information does not override the assumption of transitivity. Gorrell (1989) presents similar evidence using a lexical decision paradigm in which subjects are presented with a sentence fragment such as *the happy family lives with a lot of*, or *if your bicycle is stolen, you must.* They then have to make a lexical decision immediately on an unrelated word, such as *batteries*, which is either a syntactically appropriate or inappropriate continuation. If it is syntactically appropriate, reaction times are quicker than if it is inappropriate (Wright and Garrett 1984). Gorrell contrasted the examples below, with a lexical decision on *men*:

(6.15) He said that before the police struggled/stopped/took

(6.16) He said that before the truck struggled/stopped/took

Reaction times were slower with verbs that cannot take an NP object, like *struggled*, and therefore there was a main effect of verb type (as expected). However, there was no effect of animacy. There was no evidence that *the truck stopped* had more difficulty accepting an object than *the police stopped.* This again suggests that Stowe's effect was a late effect. If we make a combination when that combination is plausible, then we will also make the combination when it is implausible. Hence prior context will not make us violate the ICH. We may of course revise our decision quite quickly, and then treat *the truck stopped* as intransitive, but this is not the original analysis.

These results suggest that plausibility information is not relevant to whether a dependency constituent is formed. Hence information such as animacy should not be represented in the syntactic type. This distinguishes it from major and minor category information, which obviously must be represented in the type, and would usually be considered the information which defines the type. The question is raised as to whether any other information is represented in the type. We can tentatively suggest that case, number and gender will be represented in the type, and are therefore syntactic features. Therefore we would not garden path on the fragment *before the police stopped he*, because *he* cannot serve as the object to a verb. Gorrell (1989) gives evidence that number and case mismatches do slow down reaction times in his paradigm, unlike animacy and other plausibility information. In any case, information represented in the type can stop the formation of a dependency constituent, but information outside the type cannot.

Although the ICH may be impervious to semantic information, it may still be amended by prosody. This is intuitively sensible, because there may be need to make implausible utterances (and to have them understood), but there is no comparable purpose in choosing an unusual intonation. Psycholinguistic studies of ambiguity resolution have paid little attention to the possibility that experimental results could be artefactual because they are usually presented in written form without punctuation, or are artificially punctuated by the presentation technique (see Briscoe (1987) for discussion). In (6.8), the addition of a comma in the written form or a pause and a change of pitch accent in the spoken form makes the sentence much easier to understand. It is possible that the ICH is initially obeyed, and then intonational information can tell us to recover from a garden path quickly, but we may in fact never test out locally acceptable readings, which are not globally acceptable, if the speaker (or writer) is trying to guard us against making the mistake. There is probably an intimate relation between intonation and flexible constituency (Steedman 1990), so we would expect different intonations to signal which constituents to assume. The suggestion is that marked intonations can tell us not to form dependency constituents, but that the default assumption is to obey the ICH. In contrast, the implausibility of a combination never causes us to disobey the ICH and to refrain from making the combination.

6.3 Ambiguity and the Processing of Unbounded Dependencies

In chapter 4 we noted cases of local ambiguity in the processing of unbounded dependencies. The string *who had the little girl expected* could be interpreted with *who* being the object of *expected*, but this need not be the case. For instance, the sentence could continue ... *us to sing to.* This local ambiguity has been called a 'doubtful gap' by J.D.Fodor (1978). Standard accounts have assumed that such ambiguities should be explained by a gap-filling mechanism, and are not directly related to the local ambiguities discussed in the previous section. I shall discuss these accounts first. But choosing a locally correct but globally incorrect analysis is precisely what we have called a garden path. Therefore I shall reinterpret doubtful gap phenomena in terms of the analysis of ambiguity introduced in the previous section. This is to be expected given that we have rejected the treatment of unbounded dependencies by a means separated from that used with other constructions. In this way we can integrate the accounts of the processing of local and unbounded dependencies.

6.3.1 A Gap-Filling Mechanism

Let us consider the structure of a gap-filling account of the processing of unbounded dependencies. The sentence processor needs to have a specialized mechanism which (i) identifies the filler, (ii) remembers the filler, (iii) locates the gap, (iv) associates the filler with the gap, and (v) interprets the 'filler-in-the-gap' in the context of the rest of the sentence. In linguistic terms this is because there is no assumption of the rule-to-rule hypothesis, and because an autonomous syntactic representation is constructed. This translates into the assumption of modularity within the sentence processor. This is essentially the structure of Wanner and Maratsos's original account, and forms the basis for all subsequent models of gap-filling. As was mentioned in chapter 4, it cannot be obviously extended to cases where the gap comes before the filler (though there is quite strong evidence that such dependencies cannot be unbounded (Ross 1967)), and it needs additional stipulations to prevent it being a model of 'literal' gap-filling, where the filler is simply interpreted as though it were located at the gap location. Processes (iii) and (iv) have no parallel in a theory without empty categories, and (v) has to be replaced by a direct association that forms the unbounded dependency. (i) and (ii) are necessary in any incremental theory. In chapter 5, I argued against the existence of empty categories at the 'canonical' locations for extracted elements. However, it is worth discussing the putative structure of the gap-filling mechanism in some detail, with the intention of showing how it can be replaced by a procedure consistent with the present account.

A gap-filling mechanism is independent. It can be regarded as an informationally encapsulated module within the language processor (Clifton and Frazier 1989, c.f. J.A.Fodor 1983), or as a subroutine in the sentence processing algorithm (Wanner and Maratsos 1978). There is no reason to assume that the processing of non-canonical constructions with empty categories need be determined in any way by the processing of canonical constructions. Instead there must be a mechanism which is accessed under particular circumstances such as when a *wh*-word is encountered. This means that there will be a separate memory store involved in the formation of filler-gap associations, like an ATN HOLD cell, independent of those memory stores used in the processing of 'canonical' constructions (though conceivably sharing memory resources with them). In linguistic terms, a theory with empty categories assumes additional descriptive devices to deal with a relatively small range of constructions, and in processing terms, the cognitive architecture must have a special place for these constructions. This is unparsimonious, and also its modular nature means that we cannot predict all aspects of its design from the rest of the language processor. However, in the same way that GB assumes close a close relationship between the linguistic description of anaphora and empty categories, there may be close connections in the mechanisms involved in processing. Thus there may be a generalized 'coreference processor' (Nicol and Swinney 1989, c.f. Cowart and Cairns 1987) which is a module that deals with both anaphora and empty categories.

But the gap-filling mechanism is independent from canonical sentence processing. Therefore a number of different gap-filling strategies are possible. We can make a general division into 'early' and 'late' gap-location strategies. In an early strategy, the processor assumes that a doubtful gap is a genuine gap without waiting for disambiguating information. This is entirely consistent with an autonomous gap-filling mechanism. The assumption is that it is activated immediately the possibility of a gap arises. However, the original models proposed by Wanner and Maratsos (1978) and J.D.Fodor (1978) assumed late gap-location strategies, where the gap-filling mechanism is only activated when there is reasonably good evidence for a gap. Although both early and late strategies are consistent with having such a mechanism (since they only differ on when it is activated), it is interesting that these original accounts proposed late strategies. The obvious reason for this is that every time a gap is postulated, the gap-filling mechanism has to be accessed, and its autonomy suggests that to access it will be costly. This is because every time a gap is possible, we have to perform a set of operations in the gap-filling module. Remembering the filler is presumably done in parallel to other parts of sentence processing, and so need not be expensive, but gap location (and gap filling) cannot be attempted until the 'main' sentence processor calls on the module. A late strategy will obviously cut down the number of mislocations dramatically, and therefore the mechanism need not be accessed so often. For instance, Fodor proposed 'trying the next constituent', so that a gap is postulated only if the next constituent after the word which licences the gap cannot itself fill the gap. This is natural if such limited lookahead is less costly than accessing a specialized mechanism. An early strategy is consistent if erroneous gap location is not costly, though this is strange if an independent module or routine has to be accessed, and it is likely to lead to a very considerable number of mistakes. The conclusions are that we cannot predict the operations of the gap-filling mechanism from the rest of sentence processing, but we have reason to assume that a relatively late strategy which imposes some conditions on the accessing of the mechanism is more attractive (even though this will slow down interpretation). There is also no reason to conclude that gap filling and interpretation must occur at the same time as gap location.

In fact, Crain and Fodor's demonstration of the filled gap effect showed that we at least sometimes locate gaps before we check the next constituent. If we simply wait until processing *us* in (4.48), repeated below, there would be no need to mislocate the gap:

(6.17) Who could the little child have forced us to sing those stupid
French songs for at Christmas?

But the increased reaction time on *us* shows that we do mislocate the gap. Furthermore, Swinney et al's result suggests that the mechanism must at least sometimes be activated very quickly, because they found priming of the filler when testing at the proposed gap location, at the offset of the embedding verb, and this experiment was conducted at normal speaking pace. This evidence suggests that the obvious, relatively late, strategy is in fact wrong. In fact there is now considerable evidence for the immediate formation of unbounded dependencies, as we shall see, which translates into a very early gap-filling strategy.

Clifton and Frazier (1989) give a clear statement of a modern gap-filling view of the processing of unbounded dependencies. They argue that the parser consists of various informationally encapsulated modules, and that gap filling is handled by certain modules in a way that differentiates it from the rest of sentence processing, with gap locating being treated separately from gap filling. Gap location is taken to be guided by the *Active Filler Hypothesis*:

> When a filler of category XP has been identified in a non-argument po-

> sition, such as COMP, rank the option of assigning its corresponding gap to the sentence over the option of identifying a lexical phrase of category XP.

The hypothesis is couched in Government-Binding/transformational grammar terms, and assumes the full range of empty categories discussed in chapter 4. Hence the restrictions on the filler essentially make the hypothesis refer solely to *wh*-fillers. In an account of unbounded dependencies, it is simply an early gap-location strategy. We shall return to some of their specific examples later.

6.3.2 Experimental Evidence and Early Association

There is now a considerable amount of experimental evidence which supports a very early account of the formation of unbounded dependencies. It also supports the view, which is not necessitated by having a gap-filling mechanism, that location, interpretation and filling are simultaneous. Most of this evidence is consistent with an early gap-filling account, but in the light of the arguments in chapters 4 and 5, I shall present some new experimental evidence against having a gap-filling mechanism at all. However, I shall begin by using the terminology of filler and gap.

The filled gap effect shows that gaps are at least sometimes postulated immediately at doubtful gap locations. It therefore gives support to an early model. However it does not prove a universally-valid early strategy, for a number of reasons. First, it only supports early location, not early filling or interpretation. For instance, we could locate a gap immediately, and then revise the decision, but never actually fill the gap or form an interpretation. Second, it may only hold when the interpretation with the filled gap is relatively plausible. Third, reading times using self-paced reading and related methodologies are slower than normal. It is conceivable that there is a gap-filling mechanism which is activated at different times under different conditions, and that in this particular case, an early strategy is induced. This is compatible with an independent mechanism.

All of these objections can be answered. Swinney et al's result shows that filling, and not merely location, at least sometimes occurs immediately. The properties of the filler are made available when the unbounded dependency is formed. It also uses a methodology that much more closely relates to normal sentence processing than self-paced reading. But their experiment concentrates on examples where the unbounded dependency is plausible. Tanenhaus and Carlson (1989) report an experiment that shows that interpretation of the 'filler-in-the-gap' does occur at the same time as the gap location and gap filling. Subjects read the sentences below word-by-word:

(6.18) The orderly wondered which bed the nurse hurried quickly towards.

(6.19) The orderly wondered which doctor the nurse hurried quickly towards.

Both sentences are globally plausible, but (6.18) is locally implausible if subjects filled and interpreted the doubtful gap after *hurried.* They found that (6.18) took longer to read than (6.19) from *hurried* onwards, which indicates that the different plausibilities of the fillers must have been noted and so interpretation must have occurred immediately. This suggests that location, filling and interpretation are simultaneous, and occur even when the interpretation is implausible. But it does use self-paced reading.

Nicol and Osterhout (reported in Nicol 1989), using the cross-modal priming technique, found reactivation of *actress* immediately after the embedded verb *planned*, even though planning an actress is anomalous:

(6.20) That's the actress that the dentist from the new medical center in town had planned to go to the party with.

Plan can of course take an NP object, as in *plan a report.* This result fits directly with the ICH, because we can only tell that the combination is anomalous once the combination is made. Note that this result is not necessary in a theory with empty categories, because it would be possible that *plan* assigned selectional restrictions to the gap, and this prevented association with the filler. Hence we would get no activation. Although this is clearly not necessary in such an account, we would be likely to use any means available to warn us of gap filling if it is a costly process. It also tells us that the immediacy of the filled-gap effect is not an artefactual result of self-paced reading, and hence lends strong support to the use of an early strategy.

Tanenhaus and Carlson found evidence that anomalies due to such mislocations had an effect on processing (see above), and hence that reactivation reflects integration, as necessitated by the ICH. However, this study did not show that effects of anomaly were felt immediately. Garnsey, Tanenhaus and Chapman (1988) gave evidence that this effect was immediate by measuring an evoked brain potential called an N400 which it is claimed is sensitive to the implausibility of a word in context (but after a constant time lag). They found this effect immediately an extracted element could be reduced into an implausible dependency constituent, at *read* in the implausible example (6.21) below:

(6.21) The mother found out which food the children read in school.

Further support for a maximally early strategy is when the doubtful gaps are overruled by the next word, as in the filled gap experiments, but in an experiment conducted at normal reading pace. This suggests that nothing will stop the formation of an unbounded dependency, and therefore that a gap-filling mechanism will be activated immediately in all cases, even though the proportion of errors is likely to be very high indeed. Further evidence for immediate activation comes from Swinney, Ford, Bresnan and Nicol (cited in Swinney, Ford, Frauenfelder and Bresnan 1988), in a study primarily concerned with the reactivation of pronouns:

(6.22) The boxer visited the doctor that the swimmer at the competition had advised him to see about the injury.

The doctor, which cannot serve as the antecedent of the pronoun, reactivates after *advised*, where there is a doubtful gap. This shows that the doubtful gap effect can be found in reactivation studies as well. After *him*, there was also reactivation of *the boxer*, which has to be the antecedent of the pronoun, but there is no activation of *the swimmer*. This suggests that reactivation of *the doctor* is due to an immediate, erroneous, formation of an unbounded dependency, and not because of the pronoun, which appears only to activate syntactically possible antecedents.

This evidence is consistent with one possible gap-filling model, where location, filling and interpretation occur virtually instantaneously, and where the earliest possible strategy is employed in all cases when a gap is even remotely possible. This means that there are a large number of (short-lived) garden paths, when it is assumed that there is an unbounded dependency but there is in fact not. It shows that activating the mechanism is not costly at all, even though it is can only be accessed when it is demanded by the 'main' language processor. This is consistent with having empty categories, but in many respects appears highly unlikely. However, chapter 5 presented evidence that empty categories cannot be used in the processing of unbounded dependencies, and are inconsistent with a grammar which has a parsimonious relation to the parser. Hence there is evidence against this whole characterization of a gap-filling mechanism. I shall therefore outline an experiment conducted jointly with Janet Nicol which gives experimental evidence that processing unbounded dependencies does not involve a gap-filling mechanism.[5]

6.3.3 Experimental Evidence against Gap-Filling

Swinney et al's experiment showed that there was reactivation of the extracted element at the point when the unbounded dependency was formed. But it is consistent with an account with or without empty categories, because the empty category and the embedded verb, which serves as the head of the empty category, are adjacent. This is similarly true of the other experiments discussed above. We can distinguish these two possible explanations of Swinney et al's results by extracting the second post-verbal argument of a verb like *give*:

(6.23) To which butcher did the woman who had just inherited a large sum of money give the very expensive gift the other day?

An account with empty categories assumes that the filler *to which butcher* is associated with a *wh*-trace after *gift*, that is, after the direct object but before the adjunct, because this is the 'canonical' location for the prepositional phrase. On the other hand, an account without empty categories assumes that the filler is associated with the verb *give* directly, so the association takes place before the direct object.

We conducted a cross-modal priming experiment with 40 subjects, using sentences like (6.23), and tested for associates of *butcher* at *give* and at *gift*.[6] An individual subject was probed at either *give* or *gift* only. Out of the probes which were words and therefore required a 'yes' response, half were related to the extracted noun and half were unrelated. These were matched on length and frequency, and when presented for

---

[5] The idea was the author's, the materials were constructed by him, and he ran some pilot studies, but the experiment reported below was overseen by Janet Nicol.

[6] The intention was to probe at the offsets of *give* and *gift*, but due to an error half the subjects were probed 100ms earlier, obviously well within the words and perhaps after recognition. All the significant results were significant simply with the subjects who were probed at the offsets.

lexical decision in isolation, showed no difference in reaction time. Hence any difference between reaction times in the experiment could only be due to priming. It was assumed that any priming would have to reflect reactivation of the extracted noun rather than residual activation, because Swinney et al and Nicol and Osterhout have shown, using examples similar in the relevant respects, that initial activation dies away quite rapidly, in any case by the point at which the unbounded dependency is about to be formed.

The results are given in the table below:

Probe type	Probe at verb	Probe at trace
Related	674	632
Unrelated	727	676
Difference	53	44

Both differences were significant by both subjects and items analyses ($p<0.005$ on all measures). However there was no significant interaction between the two positions. The results indicated that there was priming at the verb, which suggests that reactivation is not based on the association of an extracted element with an an empty category. Assuming that reactivation does directly show the point at which an integrated interpretation is made, it indicates that the location of a trace is not involved in interpretation, and therefore that its postulation is epiphenomemal at best, but most likely does not occur at all. This therefore supports an account without empty categories and is therefore consistent with the model proposed in this thesis.

However, this conclusion is possibly too strong, for two reasons. The first reason is that the effect at the verb could be due to the postulation of a gap immediately after the verb. Chapter 5 showed that the effect of nesting ditransitives cannot be simply explained by heavy NP shift, but it is conceivable that the filler is initially associated with a gap postulated directly after the verb. This would again be an extremely wasteful use of the gap-location mechanism, but it cannot be ruled out on those grounds alone. In these examples it is also possible that a heavy-shifted reading is finally arrived at, but as shown in chapter 5, this cannot in general be the case in extracting the second post-verbal argument in ditransitives. Therefore heavy shifting cannot serve as a general explanation with double object verbs, and hence it is far more parsimonious to assume that the reactivation effect is due to the formation of the unbounded dependency, not the association of filler and gap. Note however that this result is not compatible with the extrapositional analysis, because the unextracted noun phrase (here, *the very expensive gift*) in all the examples had purely preverbal adjectival modifiers.

It is possible that subjects assume that the double object verbs may in fact be used with ellipsis (for instance, *to which charity did you give, this year?*). Therefore a doubtful gap is immediately sited at the verb. There is no direct proof against this explanation, but it has to assume what is obviously not a normal subcategorization frame for the verb. It is far more straightforward to assume that the result gives evidence for the association of the verb and the extracted element without going via a trace.

However, a more serious problem is that the experiment also gave significant priming at the trace. This can be taken as some evidence for the empty category account, except that there is no good explanation for the priming effect at the verb, and the nesting results of chapter 5 cannot be accounted for. One possibility is that the priming is residual priming from the reactivation at the verb. The effect is, however, rather strong, given that there is always a gap of four words, and, though the priming is reduced from 53 to 44 ms, the difference does not approach significance. Another possibility is that reactivation is sensitive to more than just the formation of unbounded dependencies. This is very likely in the present account because unbounded dependencies are not treated with a special mechanism.

6.3.4 Processing with Dependency Constituents

Let us now assume that there is no gap-filling mechanism, and therefore that whatever principles we have for processing 'canonical' sentences will be applied directly to processing unbounded dependencies. This is a strong claim, and is therefore directly falsifiable in a way that the assumption that we have a gap-filling mechanism is not. The assumption is that a 'doubtful gap' is really a doubtful combination, caused by a string being locally ambiguous. Usually this means a choice between combining or not, but sometimes it means a choice of how to combine. This therefore allows the possibility of garden pathing.

Let us summarize the account of sentence processing proposed in section 1 above. At a point of local ambiguity, when both readings form a dependency constituent, there is some immediate way to choose which reading to continue with and which to discard. We suggest that the most likely model is that all readings are generated in parallel, and that the choice is made at least with reference to semantic information. However it is possible that the initial choice is made purely syntactically. On the other hand, if one reading forms a dependency constituent but the other does not, the ICH says that the reading that forms a dependency constituent is assumed. The only possible revision to this is if prosodic information indicates that the dependency constituent should not be formed. In all other cases, the ICH is obeyed, but there may be a revision if the reading is implausible. In some cases, such as *because the truck stopped the driver was already getting nervous*, the revision is very quick, in other cases, such as *because Mary knitted the sweater kept her warm*, the revision is slower and perceptible. This depends on when the implausibility of the analysis becomes evident.

We can apply this model to the processing of unbounded dependencies without empty categories. Because there is no special mechanism for the processing of unbounded dependencies, direct predictions can be made.

Crucially, the ICH forces a very early strategy on the formation of unbounded dependencies, because the extracted element will be combined into a dependency constituent immediately this is consistent with a reading. This therefore makes the same predictions as the Active Filler Hypothesis in all cases where the empty category is adjacent to its governor, but of course this behaviour is required by the rest of sentence processing rather than being an additional stipulation for a particular construction. For instance, it predicts the filled gap effect. Consider Stowe's (1986) example below:

(6.24) My brother wanted to know who Ruth will bring us home to
at Christmas.

As in Crain and Fodor's example, a filled gap effect is found on *us*. *My brother wanted to know who* and *Ruth will bring* can be combined into a single dependency constituent, though this is not necessarily the globally correct reading. However, this combination is required by the ICH, and since the experiments use the written form (and there is no punctuation), prosodic information cannot prevent us making a combination. When the parser finds *us* after *bring*, it realises that this analysis is wrong. Hence a minor backtracking is required, where *My brother wanted to know who* and *Ruth will bring us* are reconstructed as separate dependency constituents. It is this process of reconstruction that causes the processing load found by Stowe, and by Crain and Fodor.

Because of the assumption of the rule-to-rule hypothesis, interpretation must always occur at the same time as combination. For instance it would not be allowed that reactivation of the properties of the filler, and presumably therefore the priming of these properties, could occur later than the time at which the combination was formed. Likewise it must be possible to assess the plausibility of a dependency constituent very quickly, with perhaps a very small lag possible if inferences have to be made with respect to context. Thus this model directly predicts that reactivation will occur at the same time as the filled gap effect, and will be very early and therefore involve a high proportion of errors. This is generally supported by cross-modal priming, but most strongly by the result of Swinney, Ford, Bresnan and Nicol, which showed reactivation of the extracted element even though this was ruled out by the very next word. It is also supported by the studies of Tanenhaus and Carlson (1989) and Garnsey, Tanenhaus and Chapman (1988), which showed that doubtful combinations were not only made but also interpreted. All of this is consistent with the gap-filling model, but because there is no requirement that interpretation is simultaneous with the construction of syntactic structure, these findings cannot be predicted.

Finally, the ICH predicts that the implausibility of a combination will not stop that combination being made. Instead it is very likely that it will cause the combination to be very short-lived. This is the case either if the context suggests that an intransitive, for instance, is far more likely than a transitive, or if the combination is itself implausible. Nicol and Osterhout's experiment showed that both were true. They found reactivation with rarely transitive verbs such as *planned*, and that this was found even when the combination was very implausible, such as between *planned* and *actress*. This combination is overruled very quickly because of its implausibility. This is exactly equivalent to Mitchell's discussion of Stowe's finding that *the truck stopped* and *the silence* are combined together very briefly even though this is utterly implausible. The conclusion from these studies is that an extracted element is combined into a dependency constituent with the embedded verb immediately, independent of plausibility considerations, but subject to grammatical considerations. This conforms directly with the ICH.

We suggested that the ICH may be overruled by prosodic information that tells us not to perform a particular combination. In 'late closure' sentences, it is clear that intonation can prevent overt garden pathing, and I suggested that it was likely that the processor never in fact looked at the wrong reading at all. This is a violation of the ICH. Briscoe (1987; 1990) gives some evidence compatible with this. He suggests that there can be prosodic differences between sentences depending on what an extracted element is associated with. This could then serve as a cue to guide us whether to conform to the ICH or not. The most well-known example of this is *wanna*-contraction, where the contraction prevents us making the association.[7] Notice that *wanna*-contraction is optional, and assuming that having any phonological effect at all is optional, we would expect that the role of prosody is limited, and hence the ICH can survive as a default. We need an experiment to test whether the intonational differences actually affect comprehension. The obvious way to do this is to take clear or extreme examples of the two intonational patterns, and to show in a cross-modal priming experiment that one causes reactivation and the other does not. This would prove that at least in unbounded dependencies, intonation can guide the formation of analyses rather than choose between them.

Let us now consider some of Clifton and Frazier's observations, from the point of view of the ICH. Some interesting data is predicted both by the ICH and the Active Filler Hypothesis. For instance, (6.25) is generally taken to be questioning who was told, not who left:

(6.25) Who did Fred tell Mary left the country?

Here, the Active Filler Hypothesis assumes a gap after *tell*, and the ICH assumes that an unbounded dependency is formed at *tell*, which is presumably never revised since the reading is plausible. Sentences like (6.26) look like a problem for both models, since neither model appears to predict the garden path that does in fact occur:

(6.26) Who did you walk to the bus stop yesterday?

We appear to assume that *walk* does not take a direct object, and that we shall find a word like *with* after *yesterday*. Clifton and Frazier suggest that a gap is sited after

---

[7] This does not always hold, for instance in *who do you think's a genius?*, but there is no reason to expect that it necessarily will. This contrasts with the common trace-theoretic assumption that *wh*-trace always blocks contraction.

walk but that lexical preferences cause us to revise this assumption, in a way that in fact turns out to be erroneous. This is essentially the interpretation given in terms of the ICH. The combination is made with *walk*, as with *planned* in Nicol and Osterhout's example, even though *walk* is only rarely transitive and vastly prefers a non-human object. But this is rapidly judged to be implausible, and the alternative analysis is pursued, until it is realised that this is in fact wrong. A similar explanation may hold for the fact that (6.27) is easier to process than (6.28), which appears involve a kind of garden-path effect:

(6.27) That's the man who John wanted to give the book to.

(6.28) That's the man who John wanted to give the book to Bill.

The ICH would again explain this by assuming that *wanted* is initially given the subcategorization frame where it takes a direct object and an infinitive, but that this is overruled quickly, and *wanted* is then assumed not to take a direct object. This generates the prediction that we would be able to prime an associate of *man* at *wanted*, which is not expected if the subcategorization frame of want that generates the preferred sentence is intially selected. (It might be preferable to use verbs which do not also take a simple object, for instance *promise*.)

However, the Active Filler Hypothesis does have problems due to the fact that it assumes the existence of empty categories, and therefore has problems with the data in chapter 5. Clifton and Frazier acknowledge that the model has problems with double object verbs like *bring*:

(6.29) Which patient did the nurse bring the doctor?

There are two readings for this sentence, depending on whether the nurse brought the doctor a patient or the nurse brought the patient a doctor. The Active Filler Hypothesis predicts the latter, because the gap is before *the doctor* in this case, whereas in fact there is a generally agreed preference for the former (see J.D.Fodor 1978; Wanner and Maratsos 1978). The problem is that the Active Filler Hypothesis assumes that, in their terms, the preference is for an early gap over a late gap, whether the gaps are arguments of different heads, or whether, as in this case, they are arguments of the same head.

The Active Filler Hypothesis assumes an early gap strategy, which appears to be generally correct. But it gives the wrong prediction for the interpretation of sentences with extractions from double-object verbs. The interesting point is that this is precisely where an account with empty categories and an account where the extracted element is associated directly with the verb diverge in their predictions. The Active Filler Hypothesis wrongly predicts that the structure *filler verb ∅ NP* will be preferred over the structure *filler verb NP ∅*. The reason for this is that it assumes that preferences for gap location when the choice is between possible heads will be the same as preferences when the choice is not between heads but simply relations to that head.

On the other hand, the ICH makes no prediction at all, because in both cases the extracted element associates directly with the verb. What varies is, so to speak, the argument slot that the extracted element fills. Since this account does not associate argument slots with particular 'canonical' locations, no principle of early association is put into question by this evidence. In other words, the data is consistent with the ICH, and is not consistent with the Active Filler Hypothesis. But we clearly have to assume some further principle to explain why (6.29) is preferentially given the interpretation where the nurse brought the doctor a patient.

We have to explain why the sentence is given this interpretation, since either way round, a dependency constituent is formed at *bring*. In such cases, we would assume that semantics could be a relevant factor. We have suggested in such cases that the processor outputs both interpretations, in order that the one more consistent with the context can be accepted. But the preference remains: *the books which I gave the boys* is better than *the boys who I gave the books*, and although there is some variation in judgements on (6.29), a sentence like (6.30) below gives a very solid garden-path effect:

(6.30) Which hospital did the doctor send the patient?

Reference to context may therefore be one relevant factor in making the decision, but not the only one that is taken into account. One possibility is that accessibility is important, in terms of the Accessibility Hierarchy of Keenan and Comrie (1977), which is discussed further in chapter 7. We shall propose an *accessibility principle* or *AP*:

Hypothesis 12 *Prefer to combine with a more accessible argument than a less accessible argument.*

There is very little data to test this on, but it captures Frazier's (1987) result that subjects are more likely to interpret ambiguous subject/object relatives in Dutch as subject relatives. However, it is clear that semantics must be relevant at some point, simply because *the boys who I gave the books* is acceptable, and because some of Frazier's examples were interpreted as object relatives.

At this point, we will not consider how these two analyses differ within the categorial system proposed in this thesis. The reason is that the Lambek Calculus is unable to deal with non-peripheral extraction in any way consistent with present assumptions. This is discussed in Chapter 7, Section 3.5.

In conclusion, it is possible to construct a unified treatment of the processing of unbounded dependencies and other constructions in a theory that does not distinguish 'canonical' and 'non-canonical' constructions at all. This is in sharp contrast with models that make use of empty categories and a gap-filling mechanism. Such accounts require a great deal of additional machinery, which cannot be integrated with the rest of the sentence processing mechanism.

6.4 The Processing of NP-trace

This thesis has argued against the psychological reality of empty categories in general, but has concentrated almost solely on *wh*-trace and unbounded dependency constructions. It is not surprising that similar arguments can be constructed against other proposed empty categories. Rather than consider the whole range of empty categories, I shall concentrate on NP-trace. NP-trace and *wh*-trace share the property that they are left as the result of movement from DS to SS, and that the process of association between the extracted element and the trace must be construed in straightforwardly syntactic terms. NP-trace is used in raising constructions, where it appears to correspond to a semantic relationship. But it is also used in passives, where its use is often claimed to be very 'theory internal' to GB. If evidence can be found for the existence of NP-trace in passive constructions, at least, then we have a rather strong processing-level correspondence with the concepts of GB, and, what is more, have evidence that these concepts have direct psychological correlates. The processor would assume the existence of traces at the locations proposed in GB theory, and would then form filler-gap associations. This would suggest that it is constructing representations consistent with the constraints of GB.

6.4.1 Nested Associations

However, we can show that the assumption of NP-traces can require nested patterns of associations in a similar way to the assumption of *wh*-traces can require nested patterns of associations. It is possible to recurse these assumed nestings indefinitely without the sentences ever becoming hard to process.

In most passives, the subject of the passive serves as the immediately post-verbal argument in the active. Assuming active forms *loved Bill* and *gave Sue a book*, we get the passive representations below:

(6.31) Bill was loved ∅.

(6.32) Sue was given ∅ a book.

As in chapter 5, the filler has to be associated with the trace, and also the trace has to be associated with the word which serves as the governor of the trace, here the verbs *loved* and *given*. Therefore these sentences can be given the representations below:

(6.33) $[\text{Bill}]_a$ was $[\text{loved}]^{\alpha}$ $\emptyset^{\alpha}_{a}$.

(6.34) $[\text{Sue}]_a$ was $[\text{given}]^{\alpha}$ $\emptyset^{\alpha}_{a}$ a book.

On the other hand, an account without empty categories simply predicts that the subject is associated with the verb directly. The present account assumes that the subject is a dependent of the 'auxiliary' verb *was*, which also has the participle as a dependent (thus the analysis is the same as that for *Bill is happy* in chapter 2.) But the crucial point is that there is no trace after the verb.

As in chapter 5, the argument is based on verbs that can take two post-verbal arguments. Here we use the ditransitive *give*. Many speakers of English allow two passives, so that not only (6.32) above but also (6.35) below is acceptable under the pragmatically felicitous reading:

(6.35) A book was given Sue.

The GB representation of (6.35) has the trace after *Sue*, because *give Sue a book* is the active word order. Hence the filler, *a book*, has to be held in memory until after *Sue*. As with the pied-piping sentences, let us extend the example in the following way:

(6.36) A bone was given a dog which was given a man.

The representation with NP-traces for this sentence is as follows:

(6.37) [A bone]$_a$ was [given]$^{\alpha}$ [a dog]$_b$ which was [given]$^{\beta}$ a man $\emptyset_b^{\beta}$ $\emptyset_a^{\alpha}$.

As with multiple pied pipings, we have both nested filler-gap and gap-verb associations. Hence the sentence should be hard to process, or at least become so when the construction is recursed enough. But it is in fact possible to recurse the construction indefinitely without problem. Note that in each case the passivization must involve the second post-verbal object:

(6.38) A bone was given a dog which was given a slave who was given
a senior slave who was given a master.

The representation with NP-traces has a row of four at the end of the sentence. Processing would therefore have to involve remembering large parts of the sentence in literal form until the very end, when suddenly a number of associations have to be carried out together. This is clearly not in keeping with the lack of processing difficulty for this sentence, and again its intonational pattern is that of a sentence with disjoint dependencies.

It is also apparent that these sentences cannot be derived from a representation involving heavy NP shift or extraposition. Heavy NP shift is impossible because the order of the arguments cannot be reversed:

(6.39) *I gave a bone [a dog which was given a man].

Extraposition is impossible because (6.40) is very bad:

(6.40) ?*I gave a dog a bone [which was given a man].

(The judgements are of course with respect to the relevant readings.) Again, either alternative analysis would probably require empty categories in the act of shifting or extraposing.

It is also possible to construct an argument based on that employed by Hudson (1990), making use of the fact that *capable of* cannot be extraposed, as (5.44) shows (repeated below):

(6.41) *We give every student a prize capable of answering all the questions.

Therefore, the passive below must involve an empty category at the end of the sentence:

(6.42) A prize was given every student capable of answering every single tricky question on the details of post-Barriers theories about the interaction between functional categories and word order.

This ought to make the sentence long and unwieldy, just as (5.46) did in chapter 5, but clearly it does not. This is further evidence against the psychological reality of NP-trace. The conclusion is that NP-trace is not compatible with the demands of a grammatical theory obeying the Strong Competence Hypothesis.

6.4.2 Experimental Studies of NP-trace

There have recently been a few attempts to provide experimental evidence for NP-trace. It is a consequence of the above argument that such results will in fact have alternative explanations. Bever and McElree (1988) and McElree and Bever (1989) consider NP-trace used in raising constructions (as opposed to passives). They presented subjects with sentences, in chunks indicated by the slashes, like (6.43) below:

(6.43) The stern judge / who met with the defence / is sure to / argue about the appeal.

GB assumes that there is an NP-trace after *sure*, with antecedent *the stern judge who met with the defence*. They presented subjects with the probe word *stern* after the sentence, and asked subjects to indicate whether the word was in the sentence. They contrasted this sentence with a number of other sentences containing GB empty categories and (overt) anaphors, but crucially with a control sentence:

(6.44) The stern judge / who met with the defence / flatly rejected
the / arguments for an appeal.

They found that subjects responded to *stern* more quickly after (6.43) than (6.44), and argue that this is because it is reactivated by the trace. First, however, the activation did not occur immediately, but only at the end of the sentence. This suggests that the activation may be due to processes far removed from the presumably immediate association of the filler with the gap. Perhaps *stern* is more salient in some complete semantic representation of (6.43) than (6.44). But a more basic problem is that the sentences are not properly controlled (see J.D.Fodor 1989). For instance, *flatly rejected the* is a strange set of words, and includes a determiner but not the rest of the noun phrase. It is very likely that this causes interference with memory for *stern*.

J.D.Fodor (1989) reports a cross-modal priming experiment by Osterhout and Swinney on NP-trace created by passivization, which used sentences like:

(6.45) The baker who had just moved into the neighbourhood was
asked by the woman to help out at the party.

They found no significant reactivation after *asked*, or 500 ms later, but did find an effect when the target was presented 1000 ms later. This contrasts with the immediacy of the finding for *wh*-trace. It is conceivable that the later priming indicated the formation of a filler-gap link, which occurred much after the gap location, but this suggests that processing NP-trace is different from processing *wh*-trace.[8] But it is far more likely that the cross-modal priming task is sensitive to more than simply the immediate formation of the dependency constituent. This is consistent with the priming at the purported trace location in the outlined experiment; both may be caused by some later integrative effect. Even if there had been reactivation at the verb *asked*, this could be due to the formation of a single dependency constituent and the reactivation of the semantic properties of *baker*. We know too little about whether reactivation is solely caused by the formation of dependency constituents, or whether higher-level processes are also relevant. A final experiment that claims to give evidence for NP-trace is in MacDonald

---

[8] Notice also that GB assumes a PRO, associated with *the baker*, after *woman*, which may contribute to this effect.

(1989). She contrasted the sentences below using an end-of-sentence probe task (asking whether the word had occurred or not):

(6.46) The new mayor at the center podium was shot.

(6.47) The new mayor at the center podium was surprised.

(6.48) The new mayor at the center podium was furious.

In GB-theory, only (6.46) uses a verbal passive, and therefore has an NP-trace. (6.47) has an adjectival passive, whereas (6.48) simply has an adjective. She found quicker responses to (6.46) than to (6.47), and to (6.47) than to (6.48). This can be interpreted as giving evidence for NP-trace, but it is just as straightforward to assume that the difference is due to a higher-level semantic effect. This could give an explanation for the difference between (6.47) and (6.48), which has no explanation in terms of empty categories.

None of these experiments show conclusive evidence for the syntactic representation of NP-trace. The argument from recursion is not consistent with a syntactic representation of NP-trace, so we have to assume that the experiments tap into some other process. There is an additional problem with NP-trace which is avoided if this empty category is not assumed, which is that it is usually or always impossible to tell that an NP will be an antecedent to a trace. For instance, in (6.46)-(6.48) above, the fact that *the new mayor at the center podium* is an antecedent in (6.46) only can only be known at the disambiguating word (in the spoken form, the intonation for all the sentences is the same). So we would either find ourselves with no filler when we needed one, or a filler when we did not need it. This is very unparsimonious, since we can parse without assuming the empty category at all.

6.5 Implications of Reactivation

This section includes a considerable amount of speculation about the implications of the cross-modal priming technique. It is apparent that we know very little about it, but we have strong evidence that it is sensitive to at least some of the low-level processes involved in sentence processing. Its appeal stems from its extremely unobtrusive nature.

Self-paced reading, for instance, has often been criticised because it is unclear that it necessarily taps into the processes involved in the earliest stages of sentence processing. For instance, reading rates are far slower than those in normal reading. This is also a problem with studies using evoked potentials, which require slow presentation of materials to get interpretable results. Other techniques such as making on- or off-line grammaticality judgements face more severe problems, because it is unclear whether these involve processes unrelated to normal comprehension. In contrast, Swinney (1979) found little evidence that subjects realised that the lexical decision task was related to the sentence listened to, so it seems likely that the task is not influenced by conscious strategies. This is less clear in the probe task used by McElree and Bever, for instance, because subjects can apply an obvious strategy of remembering words rather than trying to understand sentences, and because subjects could be comparing the probe word with the sentence at different linguistic levels.

The cross-modal lexical decision task (and the associated naming task, which has been used more rarely) also has the obvious advantage that it produces effects immediately the unbounded dependency is formed. This suggests that it is at least sensitive to the lowest level of representation, when the extracted element is associated with the rest of the sentence. We would surely expect higher-level effects to take longer, because they would require inferencing based on general knowledge or beliefs, and in J.A.Fodor's (1983) account, would require the accessing of other modules or central processes. However, although we have good evidence that this task is sensitive to the formation of the unbounded dependency, we do not know whether it is also sensitive to higher-level representations which make use of inferential mechanisms. Although it seems to be sensitive to integrative processes, reactivation may in fact be contributed to by other factors, based on higher-level semantic representations. For example, parts of a sentence which come into focus for discourse reasons may be reactivated in this way.

6.5.1 Other Types of Priming

Rather stronger evidence that the technique reflects only the early integrative processes internal to the sentence processing module would come if we could prime the extracted element in a way not based on the meaning that the word eventually contributes to the sentence. In that case, any priming would have to tap directly into integrative processes.

Tanenhaus, Carlson and Seidenberg (1985) found evidence that rhymes of the extracted element could be primed when the unbounded dependency was formed. They presented subjects with sentences like (6.49) below:

(6.49) The man was surprised at which beer/wine the judges awarded
the first prize to.

Subjects were then presented with *fear* for lexical decision. Responses were faster when the sentence contained the rhyme *beer* than when it contained the non-rhyme *wine*, but the results were not particularly strong. It is even unclear whether rhyme relationships are in general facilitatory (eg Colombo 1986). It would be worth trying to find phonological priming using other relationships, such as alliteration (Levelt p.c.). In principle, any kind of phonological priming at the assumed point of reactivation gives the strongest evidence that the reactivation taps into the lowest level integrative processes. Therefore one possibility is to use priming of contextually inappropriate senses, in a way inspired by Swinney's (1979) experiment outlined in chapter 2. Swinney showed that a sense of a word is activated, and can therefore serve as a prime, even if that sense is not supported by the context. It is quite possible that this 'wrong sense' of the extracted element is momentarily activated in the formation of an unbounded dependency. If such an effect were found, it would have to be caused by immediate integration processes alone. Therefore we could use sentences like:

(6.50) Which bank did the man raid yesterday?

Around the offset of *raid*, we would expect activation of both senses of *bank*, a side of a river and a place where money is kept. Both senses would be activated at the actual location of *bank* (according to Swinney's original finding), and given that there is no way to determine which reading is required before *raid*, presumably both senses are remembered and it is only when we reach *raid* that we can tell which is the correct sense. Because reactivating the wrong sense must be irrelevant to the interpretation of the sentence, it is impossible that a result could be confounded by higher level priming factors. In other words, we could be sure that this effect monitored basic integrative processes, and so we would strongly predict that we would only get the effect in the immediate vicinity of the point of reactivation. This could be used with ditransitive verbs to show whether reactivation, and therefore intergration, occurred at the trace or at the verb.[9]

6.5.2 Reactivation and Dependency Formation

If we assume empty categories and a gap-filling mechanism, it is conceivable that reactivation is sensitive to gap-filling or interpretation but to no other process. But in the account I have proposed, or in another model without empty categories, it is unclear what reactivation is actually sensitive to. Because of the immediacy of the effect, it does seem very likely that it indicates on-line integrative processes, though it may also be effected by other processes. It is possible that reactivation only occurs with unbounded dependencies, but this is unlikely given that we have rejected the assumption of a special processing module. More likely it is sensitive to dependency formation in general. Before the unbounded dependency is formed, the extracted element is not in the current dependency constituent. The formation of the single dependency constituent changes this, and foregrounds the extracted element. It is this foregrounding that seems likely to be the cause of reactivation. However, it is not the fact of being extracted that is important under this explanation. For instance, in an SOV language having NP NP V word order but no extraction, we would expect the first NP to be reactivated at the verb, because it has been backgrounded whilst the second NP is being processed.

This suggests an explanation for the priming effect found at the purported trace location in the experiment conducted with Janet Nicol. The extracted element is integrated with the verb, therefore recalling the dependency constituent containing the extracted element, and this causes the first reactivation. Whilst the post-verbal argument is processed. the dependency constituent containing the extracted element and the verb itself are backgrounded. However, when this NP is finished, the backgrounded dependency constituent is brought to the fore, causing reactivation. Because this is a secondary effect, it might be presumed to be less strong, but it is also possible that the effects of the first reactivation would not have died away totally, so the 44ms activation at the trace might be due to the summation of two effects.

---

[9] Note that this experiment could probably be conducted without large amounts of material separating the filler from the probe point, because 'wrong sense' priming is so short-lived.

Chapter 7

Processing Subject Extractions

7.1 Introduction

This chapter considers a type of unbounded dependencies where the subject is extracted, from the perspective of what kind of linguistic account could serve as the basis for a processing model. It includes discussion of an experiment, which it is claimed gives evidence about what form of grammar should be adopted. The chapter first considers types of subject extractions in relation to other extractions, and discusses the possibility that embedding may have a different effect on extraction from subject position than it has on extractions from other positions. It then discusses the assumptions made by two constituency theories, GB and GPSG, and concludes that the standard accounts within both theories treat subject extractions as special. This is contrasted with an account in terms of dependency constituency, which suggest that processing of subject extractions is not special. It then discusses approaches that have been taken within flexible categorial grammar. The pure Lambek Calculus cannot provide a principled treatment of embedded subject extraction, and Steedman's account has a number of problems. On the other hand, Moortgat's system does manage to capture most of the relevant data and model dependency constituency, but its inelegance shows the ultimate limitations of the Lambek Calculus for our purposes. The chapter then describes an experiment designed to test whether subject extractions involve special processing. This concludes that they do not. Hence subject extractions do not appear to have special properties, and so the most likely approach to their analysis seems to be to use a set of primitives which does not require a special mechanism to deal with them. Again, this can be adequately if not elegantly modelled in Moortgat's extension of **L**.

7.2 The Parallelism Hypothesis

7.2.1 Comparing Relative Clauses

Consider the following set of four sentences with different types of relative clause. We shall refer to these sentence types by the terms in brackets after each sentence:

(7.1) The man who loved Mary arrived yesterday. (simple subject relative)
(7.2) The man who Mary loved arrived yesterday. (simple object relative)
(7.3) The man who I thought loved Mary arrived yesterday. (embedded subject relative)
(7.4) The man who I thought Mary loved arrived yesterday. (embedded object relative)

These four sentences have surface forms that differ in only very minimal ways. The subject relatives have the word order *loved Mary*, the object relatives have this reversed. The embedded relatives have the additional words *I thought* that the simple relatives do not have. Therefore in surface form embedded object relatives differ from embedded subject relatives in exactly the same way that simple object relatives differ from simple subject relatives. In other words, adding an embedding clause has a constant surface effect.

But the similarity may only exist on the surface, and an accurate linguistic treatment of relative clauses (or of extractions in general) may require more than a reflection of this simple relationship. If extraction from subject position is special, unrelated to extraction from other positions, then it is quite possible that embedding will not have a constant effect. If so, assuming some degree of isomorphism between linguistic analysis and sentence processing, we would further expect that embedding on subject position would have a different effect on processing from embedding on object position. Hence the processing relationship between embedded subject and object relatives would be different from that between simple subject and object relatives.

We shall call the claim that embedding does have a constant effect on subject and object extraction, both on linguistic analysis and processing, the *Parallelism Hypothesis.* We shall almost exclusively concentrate on sentences with relative clauses, as in the quartet above. In the next section we shall review some linguistic evidence for and against the hypothesis, and shall conclude that there is no overwhelming evidence either way.

7.2.2 Linguistic Evidence

Simple Subject Relatives

Transformational approaches to grammar assume that (7.2) involves movement of the object of *loved.* This is because 'canonical' sentences with a transitive verb, for instance *Mary loved the man*, have the object directly after the verb, and it is assumed that the only way that a constituent can serve as the object of a verb is to be located in this position. Hence, trace-theoretic approaches give (7.2) an analysis like (7.5):

(7.5) The man [who]$_a$ Mary loved $\emptyset_a$ arrived yesterday.

A trace is positioned immediately after *loved*, and is coindexed with the relative pronoun, in order to allow the relative pronoun to serve as the object of *loved.* We shall only consider this 'standard' filler-gap association, and ignore the gap-governor relation introduced in chapter 5. This analysis therefore involves movement from an underlying or deep structure where *who* is represented after *loved.* Similar analyses have to be given for embedded subject and embedded object relatives, because in both cases an argument of the verb *loved* is not in its canonical location.

It is possible to regard simple subject relatives as involving no traces or movement at all. But the standard assumption is that they are analysed in a similar way to other relatives:

(7.6) The man [who]$_1$ $\emptyset_1$ loved Mary arrived yesterday.

We note that the movement in this case is string-vacuous, because there is no lexical material between *who* and the trace. We therefore get a unified treatment of relative clauses, where simple subject relatives involve movement. A unified treatment is also expected in theories that have no level of syntactic representation with traces or empty categories, and do not assume that a constituent needs to be associated with a canonical location in order for it to be given the correct interpretation. This corresponds to the direct filler-governor dependency also discussed in chapter 5, and therefore is the position assumed by the account discussed in this thesis. A dependency constituent is formed including the extracted element at the point when the governor is reached. But it is also the position taken by such non-transformational generative linguistic theories as Word Grammar (Hudson 1984) and recent Lexical-Functional Grammar (Kaplan and Zaenen 1988). If this view is correct, there is no obvious reason to regard simple subject relatives as fundamentally different from other relative clauses.

Coordination evidence suggests that simple subject relatives are like other relatives. Let us make two minimal assumptions about coordination, that the conjuncts must be of like category (under some loose enough definition of 'category'), and that a conjunct with an element extracted from it cannot be of the same category as a conjunct with no element extracted. Now, (7.7) is acceptable:

(7.7) This is the candidate who is extremely objectionable but we still hope will become president.

The second conjunct must involve an extraction from embedded subject position. Hence, in order for the first conjunct to be of the same category as the second conjunct, it must involve an extraction as well. For instance, if we analyse the second conjunct with a trace, we must do likewise for the first conjunct. This therefore suggests that simple subject relatives must have essentially the same analysis as other relatives.[1]

Embedded Subject Relatives

The default assumption for embedded subject relatives is that they do not fundamentally differ from other relatives (except perhaps simple subject relatives). However, there is at least some evidence that they are not like other relatives, but are marked in some way. Below we review the evidence for this, which is certainly suggestive but by no means conclusive.

---

[1] Such coordinations have been questioned, for instance by Woolford (1987), but the acceptability of (7.7) shows that at least some coordinations between embedded and simple subject relatives are possible.

As we have seen, embedded subject extractions can be coordinated with simple subject extractions. However, they can also be coordinated with simple object extractions:

(7.8) This is the candidate who everyone admired and hoped would become president.

However, it is not possible to coordinate simple subject and simple object extractions:

(7.9) *This is the candidate who became president and we admired.

Presumably these two conjuncts do not have the same category, and so the fact that embedded subject extractions can coordinate with both suggests that their category is underspecified in some way. However, this is not a property of embedded extractions generally, because we cannot coordinate embedded object extractions and simple subject extractions:

(7.10) *This is the candidate who became president and we hoped everyone admired.

Therefore this free ability to coordinate appears to be a special property of embedded subject extractions. This suggests that the construction should be regarded as marked, so long as the coordination possibilities cannot be explained in some other way.

Two other arguments for markedness are based on analogy, from other constructions and other languages. First, consider the addition of an overt complementizer *that* to the embedded relatives:

(7.11) *The man who I thought that loved Mary arrived yesterday.

(7.12) The man who I thought that Mary loved arrived yesterday.

The ungrammaticality of (7.11) is known as the Fixed Subject Constraint (Bresnan 1972) or *That*-Trace Filter (Chomsky and Lasnik 1977). Here we have a clear subject-object asymmetry. Indeed it is specifically extraction from subject position that is proscribed, because we can extract from other positions as well as direct object. This constraint holds in English (for exceptions see Sobin (1987)), and is common but not invariably found in other Germanic languages (Maling and Zaenen 1978; Engdahl 1985). There is no doubt that there is a restriction on embedded subject extraction with an overt complementizer, in at least some cases, that is not found with extractions from other embedded positions. This fact suggests that embedded subject relatives, without the complementizer, may be a marked construction as well; there is something unusual about embedded subject extractions in general. But the analogy may not hold, and the proper analysis of the Fixed Subject Constraint may be unrelated to that of embedded subject extractions without the complementizer.

There is also evidence that embedded subject extractions are cross-linguistically rarer than might be expected. Keenan and Comrie (1977) propose that relativization is guided by an Accessibility Hierarchy, where certain grammatical positions are cross-linguistically more commonly relativized on than others. Comrie (1981) argues that there is a very simple hierarchy Subject > Object > Non-Direct Object[2] > Possessor, and that this hierarchy is governed by a strong and very nearly universal principle that if a language can relativize on a particular position in the hierarchy, then it can relativize on all positions higher in the hierarchy, but no reverse implication holds. Hence for instance if a language can relativize objects then it will also be able to relativize subjects. English can relativize on all four positions, as Comrie shows:

(7.13) The man who bought the book for the girl

(7.14) The book which the man bought for the girl

(7.15) The girl for whom the man bought the book

(7.16) The boy whose book the man bought for the girl

There are languages which are more limited than English. The Fering dialect of North Fresian does not allow relativising possessors, Kinyarwanda only allows relativising subjects and objects, and Malagasy only subjects. There is evidence for a more general notion of obliqueness within English which is closely related to the hierarchy, for instance from the possibilities of passivization and perhaps reflexivization (see e.g. Pollard and Sag 1987). Some linguistic theories encode obliqueness as a primitive, e.g. Relational Grammar (Perlmutter 1983; Perlmutter and Rosen 1984), HPSG. There is therefore considerable evidence for the general significance of the ordering of the accessibility hi-

---

[2] This includes prepositional objects, and seems to refer to any 'third' argument.

erarchy. We have already seen that the hierarchy may be relevant to the processing of extractions from double object verbs.

The simplest extension of the Accessibility Hierarchy would predict that embedding would not change the ordering, so embedded subject extractions should be more frequent than embedded object extractions (and these should be more frequent than more oblique positions). As Comrie points out, this does not appear to be the case. In fact, the evidence suggests the opposite, that embedded subject extractions are cross-linguistically rarer than embedded object extractions, even when there is no overt complementizer. For instance embedded subject extractions but not embedded object extractions are ungrammatical in Hungarian and Imbabura Quechua. Comrie has no explanation of why this should be so. Thus we have evidence that the relationship between embedded subject extractions and simple subject extractions may be different from the relationship between embedded object extractions and simple object extractions (and similar more oblique pairs), but it is by no means conclusive. The argument is again by analogy, and depends crucially on the assumption that cross-linguistically rare constructions will be in some sense hard or special in a language that does permit those constructions. In conclusion, there is some evidence to suggest that embedded subject relatives are marked in a way that we would not predict from their surface form.

7.3 Grammatical Treatments of Subject Extractions

This section considers grammatical theories of subject extraction and suggests how they relate to the Parallelism Hypothesis. I shall first consider two constituency approaches, those of GB and GPSG, and claim that neither is obviously compatible with the hypothesis. I then briefly consider embedded subject extractions from a descriptive dependency perspective, and follow this with a discussion of treatments within flexible categorial grammar.

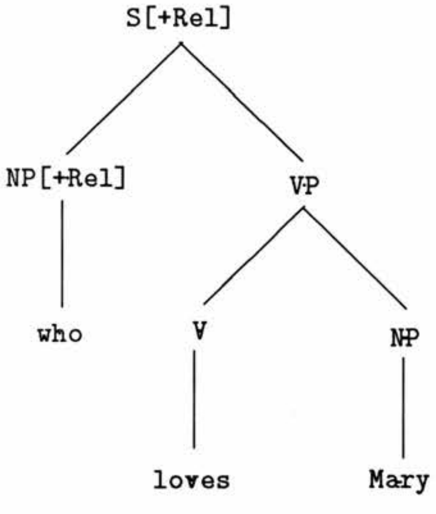
7.3.1 Constituency Theory

Government-Binding Theory

Government-Binding Theory, like all versions of transformational grammar, treats simple object relatives and embedded subject relatives using movement. Simple subject relatives are usually treated with string-vacuous movement (see above). Embedded subject relatives clearly involve movement. With respect to such constructions, Chomsky (1981) and others have taken *that*-trace violations to be 'typical', and have assumed some 'relaxation' involving the Empty Category Principle in order to allow acceptable *that*-less embedded subject extractions. Standard GB accounts therefore treat embedded subject extractions in a very construction-specific manner.

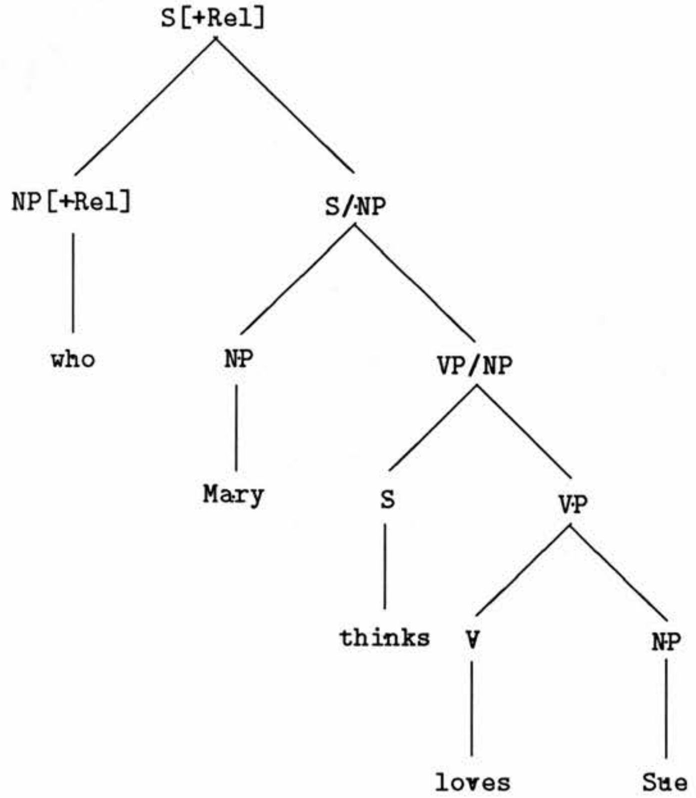
7.3.2 Generalized Phrase Structure Grammar

Generalized Phrase Structure Grammar (Gazdar, Klein, Pullum and Sag 1985) treats both embedded and simple subject relatives in ways that are fundamentally different from more oblique relatives, in a manner that clearly does not respect the Parallelism Hypothesis. Extractions in general are analysed using a slash-category, as we have seen. But this is not the case for simple subject relatives. It treats the relative clause as simply a S (marked with the +R feature), which has a NP [+R] subject and a normal VP. Hence the analysis for *the man who loves Mary* will simply be:



If we regard a relative as involving an extraction if and only if it uses the slash mechanism, as seems reasonable, then GPSG regards simple subject relatives as canonical.

Embedded subject extractions do use the slash mechanism, but in a very non-standard way. All other constructions using the slash-category are resolved by applying Slash Termination Metarule 1, which allows XP[+NULL]/XP to be rewritten as the empty string. But since this metarule only applies to categories that are sisters of lexical heads, it cannot be used for embedded subject extractions, and hence a special rule called Slash Termination Metarule 2 is introduced specifically in order to allow this construction. If we have a rule of the form X → W, V^2[+SUBJ, FIN], then we also have a rule X/NP → W, V^2[-SUBJ]. No trace is in fact postulated at all, and this analysis is unrelated to the treatment of any other relative clause type. This gives the analysis below for *who Mary thinks loves Sue*:



7.3.3 Conclusion

We can conclude that both of the theories reject the Parallelism Hypothesis. The treatments of embedded subject relatives are interestingly similar and in both cases require devices that are introduced for very limited purposes. If subject relatives are very marked constructions, then these analyses can be supported, because a very unusual construction may require a very unusual analysis. But if parallelism is supported, then both theories lose the justification for their analyses, and the fact that both theories go wrong at the same point would suggest that these theories have something in common that is seriously adrift.[3]

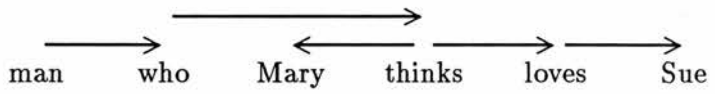
---

[3] The obvious possibility that fits with the arguments of this thesis is the assumption that they have a VP constituent, which entails that the subject has a privileged status.

7.3.4 Dependency Grammar

We have seen in chapter 4 that there is no straightforward analysis of unbounded dependencies within dependency grammar. We suggested that the most reasonable representation of an embedded object relative, such as *the man who Mary thinks Sue loves*, has the relative pronoun *who* as the head of *thinks*, *thinks* as the head of *loves* and *loves* lacking an object dependent. In this way, it is straightforward to represent *the man who Mary thinks loves Sue* as:

(7.17)



But we criticised the assumption of a relationship between *who* and *thinks* in chapter 4, and replacing *Sue loves* with *loves Sue* clearly makes no difference here.

However, we can talk about the dependency relations, as introduced in chapter 2, within the body of the relative clause. We have assumed that verbs serve as heads and have arguments like subjects and objects as their dependents. Therefore in *Mary thinks John loves Sue*, *loves* is the head of *John* and *Mary*, but is the dependent of the main verb *thinks*. Hence *Mary thinks John loves* is a dependency constituent, and so forms a dependency constituent when *Sue* is extracted. On the other hand, *Mary thinks* and *loves Sue* do not form a dependency constituent to the exclusion of *John*, but only because it is not a string. Hence when *John* is extracted, *Mary thinks loves Sue* is a dependency constituent, as is *Mary thinks loves*, or *thinks loves.*

However, we cannot assign a dependency constituent structure to the full relative clause in this way because there is no adequate representation of dependency relations for unbounded dependencies. We assume that a complete relative clause, such as *who Mary thinks Sue loves* is a dependency constituent simply because it serves as a modifier to a noun, and *who Mary thinks Sue* is not, because *Mary thinks Sue* is not a dependency constituent, and so the coindexation relation should be expressed between *who* and *loves. Who Mary thinks loves Sue* is also a dependency constituent, because it is a modifier, but it is not immediately clear that *who Mary thinks loves* is a dependency constituent, and hence that *who* and *loves* should be marked by coindexation.

There are two reasons to suggest this relationship. First, *loves* governs *who* whether it is a subject extraction or an object extraction. Second, we can appeal to processing evidence using the Minimal Incrementality Condition. We have already shown that simple subject relatives must allow the association of the relative pronoun with the embedded verb immediately the verb is encountered, and therefore that a string like *who loves* or *the man who loves* is a dependency constituent in *the man who loves Mary.* This was used to argue against the CCG treatment of island constraints. We can construct a similar argument to show that *the man who Mary thinks loves* is a dependency constituent and must be interpretable as a unit, without having to wait for *Sue.* Notice that there is no processing difficulty with (7.18) below:

(7.18) I saw the farmer who Mary said owned the dog which Fred noticed chased the cat which Bill believed followed the mouse which Sue assumed nibbled the cheese.

As before, we can extend this sentence as long as we like. Any account which assumed that the dependency constituent was not formed between the relative pronoun and its governor, but which required the next object to be found first, would make sentences of this form harder to process every time another level of embedding was added.

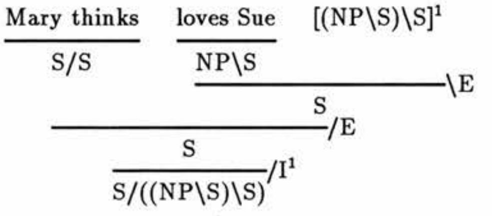
7.3.5 Flexible Categorial Grammar

We have seen how to encode dependency constituency within the Lambek Calculus. Therefore we would expect that this characterization to work with embedded subject extractions, and we would suspect that we could show that *who Mary thinks loves* is a dependency constituent, and require no abstracting over a functor. But there is a very fundamental problem with this, which is that the Lambek Calculus cannot provide a principled account of embedded subject extractions at all.

As we have seen in chapter 4, **L** can give us appropriate treatments of many unbounded dependencies. Extractions from a right peripheral position (as in many object relatives) can be dealt with by giving the relative pronoun the type PoV/(S/NP), and extractions from left-peripheral position (as in simple subject relatives) require the type PoV/(NP\S). But this is not possible when the extraction is not peripheral to a given constituent. We can see this in terms of the semantics of the directional slashes. *Mary loves* is of type S/NP because it can combine with an immediately following NP to form an S, and likewise *loves John* is of type NP\S because it can combine with an immediately preceding NP to form an S. But there is no type that we can assign to *Mary thinks loves Sue* which will allow us to represent the fact that it can combine with an NP internal to it to form an S. This means that we have no way to deal with embedded subject extractions. Not surprisingly, there are also other constructions that involve medial extraction. An important example is the extraction of the first post-verbal argument of a ditransitive, as in the relative clause *which I put under the mat.* Again, *I put under the mat* can combine with a medial NP to give an S, and so this construction cannot be represented in **L** either. We can provide similar descriptive evidence to that used for embedded subject extractions that *which I put* is a dependency constituent here, but, again, the Lambek Calculus cannot be used to make any comment about this. This explains why we did not give categorial grammar representations of (6.29) in Chapter 6, Section 3.4 above.

However it is too strong to claim that it is impossible to analyse medial extractions at all within **L**, because we can form a more complex type for the body of the extraction, and give the extracted element a type which serves as a functor over the complex type. For instance, we can analyse *Mary thinks loves Sue* as below:

(7.19)



If the extracted element is given a type that looks for $S/((NP\backslash S)\backslash S)$ to its right, then the derivation can be performed. This is clearly not valid within the dependency-preserving subset of **L**. It is easy to show that there is no means of performing a dependency-preserving derivation of a sentence that contains a string like *Mary thinks loves Sue* assuming these type-assignments for *Mary thinks* and *loves Sue.* This is obviously inadequate. More generally, it seems very unlikely that any account based on **L** will be satisfied with this derivation, because we have to assume an otherwise unmotivated type for the extracted element. Similar points can be made about other cases of medial extraction.

Any flexible categorial grammar treatment based on **L** will therefore require an extended apparatus for this comparatively small set of constructions. As in the constituency theories that we have considered, this only seems reasonable at all if the construction is highly marked. But if the construction is not so marked, it suggests that **L** may be an inappropriate formalism for the description of natural language, at least so long as the slashes are given the standard interpretations. We shall next consider two accounts within flexible categorial grammar of medial extractions, and discuss how they relate to dependency constituency.

Combinatory Categorial Grammar

Chapter 4 mentioned that CCG includes **L**-invalid rules called mixing rules, and gave the example of backward mixed composition. It noted that the addition of **L**-invalid rules leads to word order collapse, but that CCG prevents this by the use of type-specific restrictions on mixing rules. The discussion was then limited to the **L**-valid subset of CCG. We shall now return to a discussion of the mixing rules, which Steedman (1987) uses in his treatment of embedded subject extractions.

The mixing rules suffer from the general problem that they fail to respect the directionality restrictions of the slash connectives, as we have mentioned, and if unconstrained, lead to word order collapse. Therefore Steedman imposes type-specific restrictions. He attempts to characterize embedded subject extractions using the directional slashes. There are two possible methods, one giving the body of the relative the type S/NP, which means that it is analysed as though it lacks a NP to its right, the other giving it the type NP\S, which means that it is analysed as though it lacks an NP to its left. Neither seems right, because it 'really' lacks an NP in its middle.[4]

Steedman first suggests introducing another mixing rule, forward mixed composition, to combine *Mary thinks*, of type S/S, and *loves Sue*, of type NP\S, to give the resultant type NP\S. But this then allows us to overgenerate, permitting a sentence with an untopicalized NP before *Mary*, and forbids coordination between an embedded subject relative and an object relative, like (7.8).

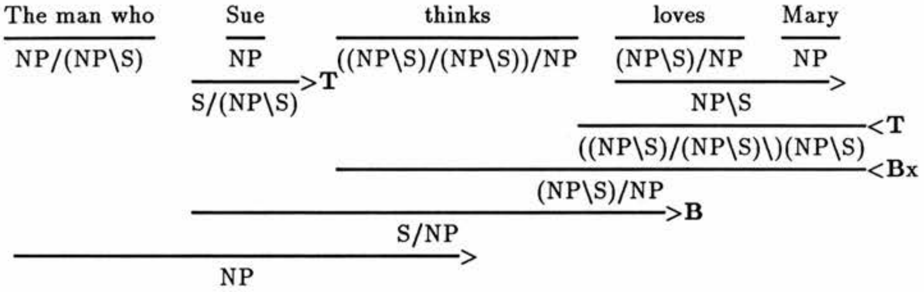
Steedman prefers making the body of the relative have the type S/NP, and instead gives verbs like *think* or *believes* the special type ((NP\S)/(NP\S))/NP, in the place of

---

[4] Perhaps it does not 'really' lack an NP *anywhere*, because this seems to assume the existence of extraction sites, but instead it lacks an embedded subject NP, which does not have a specific location. The point is that some residual notion of an extraction site seems tied up with the interpretation of the slashes in terms of *where* the missing argument has to be found, not *what* the missing argument is.

their normal type (NP\S)/S, when they are used as bridge verbs in embedded subject extractions. This means that they first combine with an NP and then with a VP rather than with an S directly. The revised derivation still involves a mixing rule, this time backward mixed composition:

(7.20)



This account still has a number of problems. It assumes that an embedded subject relative has the type S/NP, which permits the generation of heavy-shifted constructions such as Steedman's (58b):

(7.21) ?Brutus implied was no good each Caesar whom he ostensibly praised.

This analysis is highly questionable, and seems fundamentally just as inadequate as giving it the type NP\S. Using the type S/NP also prohibits coordinations between simple and embedded subject relatives, as in (7.7). The fact that embedded subject relatives can coordinate with both simple subject and object relatives shows that neither S/NP nor NP\S can be the correct type.

But we also lose the derivational flexibility that Steedman regards as a major advantage of CCG. In the present terms, we have seen that CCG approximates to the dependency-preserving subset of **L** and therefore allows the formation of a good proportion of dependency constituents. But the correspondence between his analysis of embedded subject extractions and the strings we have argued are dependency constituents is weak. We have seen that *Mary thinks*, *thinks loves* and *Mary thinks loves*, for instance, are dependency constituents. In fact, Steedman does not allow the generation of any of these. He only allows the derivation given above, with the constituents *loves Mary*, *thinks loves Mary* and *Sue thinks loves Mary*. This creates many problems. For example, because we cannot combine the subject NP *Sue* with the verb *thinks* directly, we cannot treat *Sue thinks* as a conjunct in a coordination like *who Sue thinks and Bill hopes loves Mary* (in comparison his account does allow *who Sue thinks and Bill hopes Mary loves*).[5]

Also it is necessary to combine *loves* with *Mary* before any combination can be made with *thinks*, again because of the restriction on possible mixing rules. This creates the problem that there is no incremental analysis of sentences like (7.18), and therefore Steedman's analysis fails to satisfy the MIC. The general conclusion is that mixing rules cannot provide a satisfactory treatment of medial extractions in general and embedded subject extractions in particular, because we cannot retain the flexibility we need and at the same time prevent overgeneration. This is related to the deeper fact that we cannot represent an element missing from the middle of a string by a type that otherwise means there is an element missing from one periphery of the string.

Extending L with Additional Connectives

Steedman includes mixing rules in his system and therefore makes the directional slashes have complex meanings. An alternative way to deal with phenomena like embedded subject extraction is to include an additional connective. Moortgat (1988a) introduces the extraction operator $\uparrow$, where an expression of the form $X\uparrow Y$ designates an incomplete expression that has an argument Y missing somewhere (not necessarily on the periphery), and will yield an expression of type X in combination with such an argument. Hence $\uparrow$ has a more general meaning than $\backslash$ or $/$.

In the natural deduction framework, it is possible to give the connective an introduction rule, but not an elimination rule. The introduction rule $\uparrow$I takes the form:

(7.22)
$$\dfrac{\begin{array}{c}[Y]^n\\ \vdots\\ X\end{array}}{X\uparrow Y}\uparrow\mathrm{I}^n$$

Here Y can be any undischarged assumption in the proof of X. We can straightforwardly prove that X/Y or $Y\backslash X$ can be rewritten as $X\uparrow Y$:

---

[5] I have skipped many details of Steedman's apparatus. This structure would be permitted if Steedman allowed lifting the NP\S *loves Mary* to S/(NP\S)\S rather than (NP\S)/(NP\S)\(NP\S).

(7.23) $$\dfrac{\dfrac{X/Y \quad [Y]^1}{X}\,/\mathrm{E}}{X\uparrow Y}\,\uparrow\mathrm{I}^1 \qquad\qquad \dfrac{\dfrac{[Y]^1 \quad Y\backslash X}{X}\,\backslash\mathrm{E}}{X\uparrow Y}\,\uparrow\mathrm{I}^1$$

These transitions are valid in terms of the semantics of the connectives, because an expression lacking a Y on its immediate left or right is an expression lacking a Y somewhere. However, the lack of an elimination rule means that not all valid transitions are permitted. For instance, $X\uparrow Y$ and $Y\uparrow Z$ could validly combine to give $X\uparrow Z$, but this is not possible in the present system.

Let us call this extended system $\mathbf{L}^{\uparrow}$. We can now capture the fact that certain strings should be analysed as dependency constituents even though they cannot be generated within **L**. In *the man who Mary thinks loves Sue*, we can give the string *thinks loves Sue* the type (S↑NP)/NP by dependency-preserving operations, because there is no need to abstract over a functor in the derivation:

(7.24) $$\dfrac{\dfrac{\dfrac{\text{thinks}}{(\mathrm{NP}\backslash\mathrm{S})/\mathrm{S}} \quad \dfrac{[\mathrm{NP}]^1 \quad \dfrac{\text{loves Sue}}{\mathrm{NP}\backslash\mathrm{S}}}{\mathrm{S}}\,\backslash\mathrm{E}}{\mathrm{NP}\backslash\mathrm{S}}\,/\mathrm{E}}{(\mathrm{NP}\backslash\mathrm{S})\uparrow\mathrm{NP}}\,\uparrow\mathrm{I}^1$$

Similarly, it is possible to show that other strings which are descriptively dependency constituents, such as *thinks loves*, *Mary thinks* and *Mary thinks loves*, can be given types by dependency-preserving operations. Hence it appears possible to characterize the dependency-preserving subset of $\mathbf{L}^{\uparrow}$ just as was done for **L**.

We can give relative pronouns the lexical type R/(S↑NP), and hence analyse *who Mary thinks loves Sue* as below. We do not include the stages in the natural deduction proof:

(7.25) $$\dfrac{\dfrac{\text{who}}{\mathrm{PoN}/(\mathrm{S}\uparrow\mathrm{NP})} \quad \dfrac{\dfrac{\text{Mary thinks}}{\mathrm{S}/\mathrm{S}} \quad \dfrac{\text{loves Sue}}{\mathrm{NP}\backslash\mathrm{S}}}{\mathrm{S}\uparrow\mathrm{NP}}}{\mathrm{PoN}}$$

But we can also perform the incrementally-valid derivation, where *who Mary thinks* *loves* is a dependency constituent (but where *who Mary thinks* is not):

```
(7.26)     who        Mary thinks       loves        Sue
        ----------    -----------   ------------    ----
        PoN/(S↑NP)        S/S        (NP\S)/NP       NP
                          ---------------------
                                (S↑NP)/NP
        -----------------------------------
                    PoN/NP
                    ---------------------------------------
                                     PoN
```

This system therefore satisfies the MIC, because strings like *who Mary thinks loves* can be given a type and an interpretation by dependency preserving operations. This contrasts with the CCG system discussed above.

$\mathbf{L}^{\uparrow}$ allows the representation of dependency constituents in the same way that they can be represented in **L**, and so from this point of view the system is adequate. This contrasts with CCG, which is unable to form all the constituents that we have indicated we need for coordination and for incremental interpretation. However, $\mathbf{L}^{\uparrow}$ is a rather inelegant formulation which assumes that non-peripheral extractions, such as from embedded subject position, have to make use of special machinery. It is crucially based on the additional connective not having an elimination rule. It also appears unable to prevent coordinations between simple subject and simple object relatives, such as (7.9), because both conjuncts can be given the type S↑NP. More generally, the distinction between peripheral and non-peripheral extraction is irrelevant from the point of view of dependency and dependency constituency. This is a problem which cannot be avoided within a framework like the Lambek Calculus, which is based on linear ordering. It suggests that $\mathbf{L}^{\uparrow}$ can represent dependency, but not in a transparent way. Another, related, limitation is that this account cannot explain the Fixed Subject Constraint or *that*-trace filter, as discussed in Section 2.2 above.

We can now give derivations for the two readings of (7.27), repeated as (7.27) below:

(7.27) Which patient did the nurse bring the doctor?

Let us instead consider the NP *the patient who the nurse brought the doctor*, because it is similar to the derivations above, and avoids irrelevant complications regarding the representation of questions. The dispreferred reading, where the nurse brought the patient a doctor, is given below under the incrementally-valid derivation. This is the problematic case for **L**. In this derivation, the two object NPs are distinguished by alphabetic subscripting:

$$\dfrac{\dfrac{\dfrac{\text{the patient who}}{\mathrm{PoN}/(\mathrm{S}\uparrow\mathrm{NP}_\beta)} \quad \dfrac{\dfrac{\text{the nurse}}{\mathrm{NP}} \quad \dfrac{\text{brought}}{((\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}_\alpha)/\mathrm{NP}_\beta}}{(\mathrm{S}/\mathrm{NP}_\alpha)/\mathrm{NP}_\beta}}{\mathrm{PoN}\uparrow\mathrm{NP}_\alpha} \quad \dfrac{\text{the doctor}}{\mathrm{NP}_\alpha}}{\mathrm{PoN}} \tag{7.28}$$

The extracted element combines with NP_β in the type given to *brought*, which is also what *the patient* combines with in the sentence *the nurse brought the patient the doctor*:

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{The nurse}}{\mathrm{NP}} \quad \dfrac{\text{brought}}{((\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}_\alpha)/\mathrm{NP}_\beta}}{(\mathrm{S}/\mathrm{NP}_\alpha)/\mathrm{NP}_\beta} \quad \dfrac{\text{the patient}}{\mathrm{NP}_\beta}}{\mathrm{S}/\mathrm{NP}_\alpha} \quad \dfrac{\text{the doctor}}{\mathrm{NP}_\alpha}}{\mathrm{S}} \tag{7.29}$$

Hence (7.28) corresponds to the dispreferred reading.

The preferred reading, where the nurse brought the doctor a patient, has the incrementally-valid derivation below:

$$\dfrac{\dfrac{\dfrac{\text{who}}{\mathrm{PoN}/(\mathrm{S}\uparrow\mathrm{NP}_\alpha)} \quad \dfrac{\dfrac{\text{the nurse}}{\mathrm{NP}} \quad \dfrac{\text{brought}}{((\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}_\alpha)/\mathrm{NP}_\beta}}{(\mathrm{S}/\mathrm{NP}_\alpha)/\mathrm{NP}_\beta}}{\mathrm{PoN}\uparrow\mathrm{NP}_\beta} \quad \dfrac{\text{the doctor}}{\mathrm{NP}_\beta}}{\mathrm{PoN}} \tag{7.30}$$

(Note that this derivation is also possible in **L**.) Here, the extracted element combines with $\mathrm{NP}\alpha$ in the type given to *brought*, which is also what *the patient* combines with in the sentence *the nurse brough the doctor the patient*:

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{The nurse}}{\mathrm{NP}} \quad \dfrac{\text{brought}}{((\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}_\alpha)/\mathrm{NP}_\beta}}{(\mathrm{S}/\mathrm{NP}_\alpha)/\mathrm{NP}_\beta} \quad \dfrac{\text{the doctor}}{\mathrm{NP}_\beta}}{\mathrm{S}/\mathrm{NP}_\alpha} \quad \dfrac{\text{the patient}}{\mathrm{NP}_\alpha}}{\mathrm{S}} \tag{7.31}$$

Hence (7.31) corresponds to the preferred reading. We can see that the difference between the two methods of combination is, so to speak, which argument slot is combined with which argument, but that either combination occurs at the same time. We therefore cannot explain the preference by appealing to any principle of early association, and hence need to invoke an additional mechanism such as the *accessibility principle* discussed in Chapter 6, section 3.4. However, the basic account does not make the wrong prediction, as does the Active Filler Hypothesis of Clifton and Frazier (1989).

7.3.6 Conclusion

The categorial analyses bear a direct relation to the sentence comprehension, because they allow incremental processing. Hence we can make some direct predictions about how we process relative clauses. This contrasts with constituency theories, where, as always, incremental interpretation is impossible without some complicated additional parsing mechanism. We can be reasonably sure, however, that any parser which is based on a constituency theory like GB or GPSG that denies the parallelism hypothesis will process subject relatives in a way that ignores parallelism as well. The CCG analysis does not allow the same flexibility in the formation of constituents in embedded subject extractions that it allows in other extractions, and specifically it restricts the incremental constituents possible, and so it predicts that parallelism will be rejected. On the other hand, the dependency-preserving subset of $\mathbf{L}^{\uparrow}$ does allow the same flexibility for embedded subject extractions, so it predicts that parallelism will be supported.

7.4 Processing Subject Relatives

Unless we assume that linguistic theories have no direct psychological reality, there must be some relation between complexity of linguistic analysis and markedness on the one hand, and processing difficulty on the other. A theory that assumes empty categories in all relatives except simple subject relatives predicts that simple subject relatives will be easy to process because the gap-filling mechanism will not have to be accessed. A theory where simple subject relatives involve extraction, or a theory that avoids empty categories completely, will predict that simple subject relatives are processed similarly to other relatives. Any theory that treats embedded subject relatives with a marked analysis will predict that they require unusually complex processing. For instance, in a theory with empty categories, some part of the gap-filling procedure must be more difficult than with other relatives.

How can we test whether embedded subject relatives are unusually hard to process, or whether simple subject relatives are unusually easy? There has been very little direct research into the processing of subject relatives (though Stowe (1986) is an exception). Consider Wanner and Maratsos's uncontroversial finding that simple subject relatives are easier to process than simple object relatives. It is obviously possible that this signifies that a gap-filling mechanism is used in simple object relatives but not in simple subject relatives. But they argue instead that there is a gap in both kinds of relatives, with the gap being earlier in simple subject relatives than in simple object relatives. Hence the filler has to be remembered for less long before it can be associated with the gap. Other explanations are possible, for instance, that it is easier to extract from a position higher up the Accessibility Hierarchy than a position lower down. Some evidence for this may come from Frauenfelder, Segui and Mehler (1980), who contrasted processing of simple subject relatives and stylistically-inverted simple object relatives in French. The latter have the same word order as subject relatives; the only difference is the form of the relative pronoun (*que* rather than *qui*). Hence the filler would have to be remembered for the same length of time. Simple subject relatives were still easier than simple object relatives. This might suggest that simple subject relatives therefore do not use gaps, but that simple object relatives do use gaps. However there are alternatives, such as the difference in obliqueness, or because inverted simple object relatives are themselves marked. From this point of view, it is also not clear how we should interpret Frazier's (1987) study in Dutch, which showed a preference for regarding ambiguous simple subject/object relatives as subject relatives.

Likewise, it would not be possible to compare embedded subject relatives and embedded object relatives directly. For instance, embedded subject relatives might prove easier to process than embedded object relatives because the gap comes earlier. However, we can compare the sentences as a quartet. Because there are no grounds for assuming either that embedded subject relatives are especially easy to process or that simple subject relatives are especially hard, there are only two possible patterns of results. These are (i) that the difference between embedded subject relatives and embedded object relatives is the same as the difference between simple subject relatives and simple object relatives, or (ii) that it is less. (i) supports the Parallelism Hypothesis, (ii) is evidence against it.

This second possibility includes three sub-cases, which we can see in Figure 1 below. (a) Embedded subject relatives may actually be harder to process than embedded object relatives, (b) They may be the same in difficulty, or (c) Embedded subject relatives may be the easier to process, but the difference between them and embedded object relatives may be less than the difference between simple subject and simple object relatives.

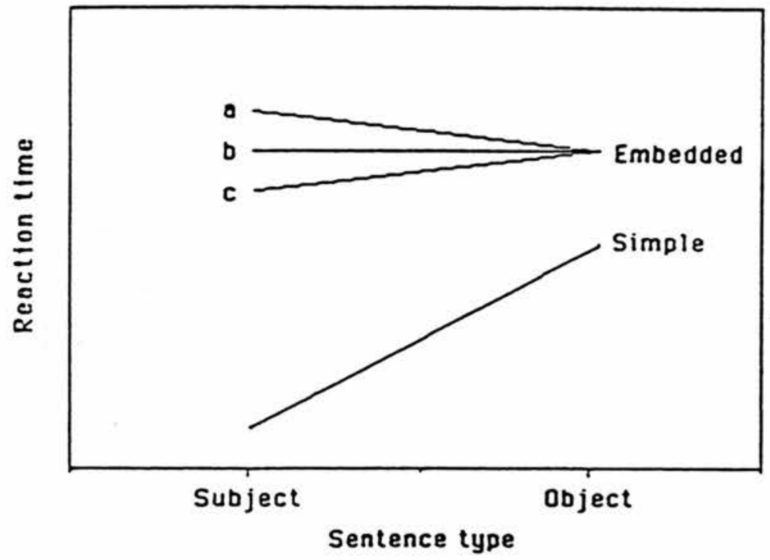


Figure 1. Possible results which would contradict the Parallelism Hypothesis (see text for details)

We shall speak of these sub-cases together, using the 'less difference' terminology of sub-case (c). If the difference is less (case (ii)) this may be because simple subject relatives are especially easy to process, or because embedded subject relatives are especially hard, or because of a combination of the two. If the difference is the same (case (i)), then we can conclude both that simple subject relatives are not especially easy and that embedded subject relatives are not especially hard to process, because, as mentioned above, there appears to be no reason to assume that two effects are cancelling each other out. Hence case (i) constitutes evidence for the Parallelism Hypothesis.

Looking back at sentences (7.1), (7.2), (7.3) and (7.4), we can see that this comparison is directly controlled. The surface form of (7.3) is simply an extension of that of (7.1) in exactly the same way that (7.4) is an extension of (7.2). Therefore by comparing the processing of these sentences, we can argue for quite specific explanations. This contrasts with, say, Wanner and Maratsos's results, which supported a number of different possible explanations.

Shillcock (1984) compared the global reading times for embedded subject and object relatives (but not their simple counterparts). There was a non-significant tendency ($p=0.12$) for embedded subject relatives to be easier to read than the embedded object relatives. This therefore gives very tentative support for the Parallelism Hypothesis. However, in order to test the hypothesis, we clearly must test simple relatives as well.

A reading time experiment considering these four sentence types was conducted. A self-paced reading time method was used, where the reading time for each presentation was recorded. Local parallelism, considering the corresponding phrases in the contrasting sentences, would constitute much better evidence for the Parallelism Hypothesis than a global result would. Conversely, if the hypothesis were refuted, the local measures might help us find an explanation.

7.5 Experiment

7.5.1 Materials and Method

We conducted a self-paced reading time experiment, with the sentence materials divided into words and phrases, in order to test the Parallelism Hypothesis. There were 48 subjects in the experiment, which was conducted on a BBC microcomputer and terminal. The four sentence-types were contrasted, varying as to whether the subject or the object was relativized on and whether there was an additional embedding clause or not. All sentences contained the relative clause introducer *that* rather than *who*, because *that* serves both as a subject and as an object relative marker.[6]

An example of an embedded subject extraction is given below:

(7.32) The sportsman that you thought hated Karen was having a very bad season.

In the simple sentences, the words *you thought* were omitted. In the object sentences, *hated* and *Karen* were reversed. Thirty-two quartets were constructed. These are included in the appendix. The embedding clause used one of the four verbs *think*, *swear*,

---

[6] We did not use *who*, or *who* and *whom*, because of the unclear status of *whom*.

believe and *hope*, in either the past or present tense form, together with a subject pronoun. All four verbs commonly take sentential complements, and are bridge verbs (that is, it is possible to extract from the complement). The most embedded clause consisted of a name and a regular past tense verb, and in every example both words had the same number of letters and syllables as each other.

The sentences were divided into seven presentation frames, and pressing a key removed one presentation and immediately replaced it with another directly after it on the same line of the screen. The divisions of the sentence were as follows (with the subscripts indicating the number of the previous frame):

(7.33) The sportsman $_1$/ that you thought $_2$/ hated $_3$/ Karen $_4$/ was
$_5$/ having a very $_6$/ bad season.$_7$

For the simple relatives, the second frame merely consisted of the word *that*, and for the object relatives, the third and fourth frames were reversed. The crucial third, fourth and fifth presentations were always one word only. The sixth varied in length, and the final frame was always the last two words only.

Thirty-two filler sentences were included from a variety of syntactic types. All the sentences, including the fillers, had an associated yes/no question, involving some paraphrasing, which subjects answered when they had finished reading the sentence. These were included purely to encourage the subjects to process the sentences fully for meaning. The experiment made use of four groups of twelve subjects in a Latin Square design. All subjects were presented with the sentences in the same random order.

7.5.2 Results

The reaction times were measured from the second presentation onwards. The first presentation is irrelevant, since it occurs before disambiguation. An initial analysis of the results replaced outliers two or more standard deviations from the mean for each subject with the value for two standard deviations from the mean. (Almost all outliers were more than two SDs *above* the mean.) The global means are shown in the table below:

	Embedded	Simple
Subject	606.9	526.5
Object	626.8	550.0

These means are represented in the figure below:

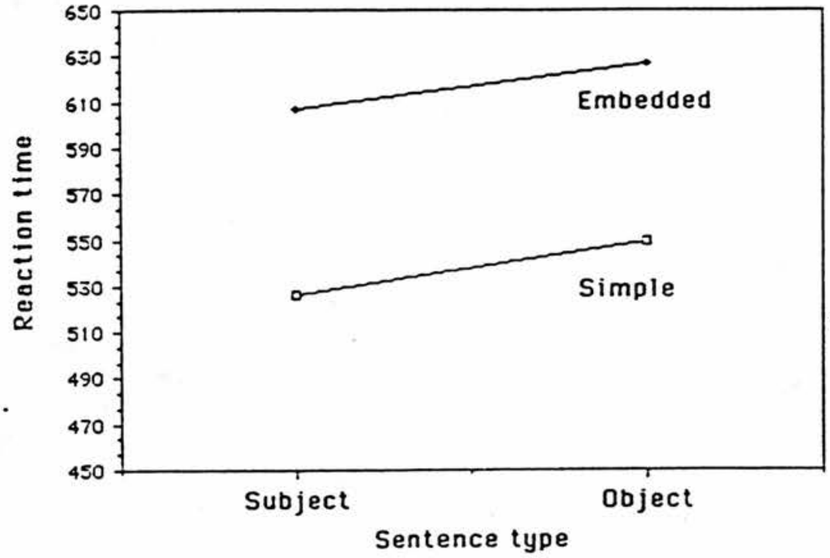


Figure 2. Mean reaction times, averaged over 6 presentations, for the four sentence types (N = 48)

Subject relatives were much easier to process than object relatives (F1=17.3, df=1,44, p<.0001; F2=7.9, df=1,31, p<.01). This also held for both embedded and simple relatives. Simple subject extractions were easier than simple object extractions (F1=5.9, df=1,44, p<.02; F2=3.3, df=1,31, p<.05), and embedded subject extractions were easier than embedded object extractions (F1=8.2, df=1,44, p<.01; F2=1.8, df=1,31, n.s.). There was no interaction between the embedded-simple and subject-object effects. As figure 2 shows, the data support the Parallelism Hypothesis: the same relative difference seems to obtain whether there is embedding or not. Embedded extractions took longer to read than simple extractions, when we compared from the third presentation onwards (F1=35.5, df=1,44, p<.0001: F2=27.7, df=1,31, p<.0001) (Presentation 2 is not comparable between the embedded and the simple materials).

We also considered the reading time on presentations 3 to 5, the two points within and the one immediately after the most deeply embedded clause. The means are given in the table below, for each presentation, with embedded and simple sentences considered both together, and separately:

Sentence type	Presentation 3	Presentation 4	Presentation 5
Simple subject	478.7	527.2	515.9
Simple object	473.1	543.8	578.0
Embedded subject	601.7	598.4	571.2
Embedded object	597.5	624.7	616.3
Total subject	540.2	562.8	543.6
Total object	535.3	584.2	597.1

The following figure shows the points considered together:

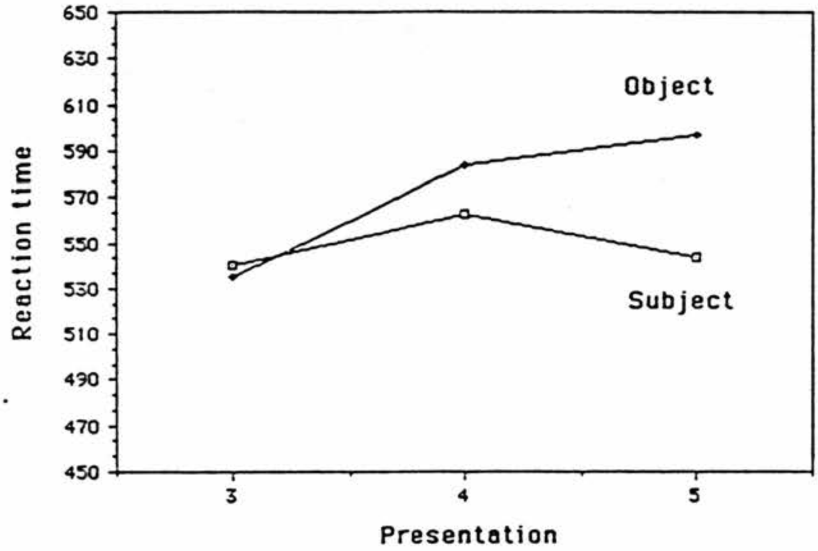


Figure 3. Mean reaction times for presentations 3, 4 and 5 for the simple materials and the embedded materials combined (N = 48)

This graph shows that object relatives are harder to process than subject relatives at the end of the most embedded clause but not at the beginning. This is predicted by the fact that the association between the extracted element and the embedded verb occurs later in the object relative than in the subject relative. In an analysis which included only presentation 5 and the mean of presentations 3 and 4, object extractions were found to be harder relative to subject extractions at presentation 5 but not at presentations 3 and 4 (F1=9.3, df=1,44, p<.005; F2=6.3, df=1,31, p<.02). (It would not be a valid comparison to look at either the third or the fourth presentation alone, because they differed between the subject and the object relatives.)

Figure 4 shows the points considered separately, depending on whether they are embedded or simple relatives:

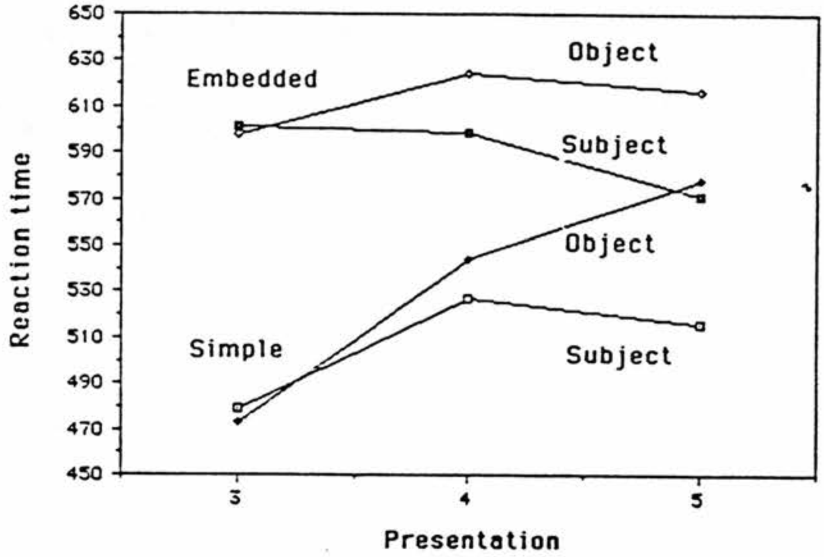


Figure 4. Mean reaction times for presentations 3, 4 and 5 for the simple materials and the embedded materials (N = 48)

Very similar graphs are found for embedded and non-embedded relatives considered separately. What is most striking, however, is the parallelism between the two graphs, given that we have found the method sensitive to local differences in processing difficulty. In other words we have both global parallelism and meaningful local parallelism.

7.6 Discussion of Results

7.6.1 Summary

The experiment supported the Parallelism Hypothesis. The methodology is clearly sensitive to quite small differences in processing complexity. Subject relatives are easier to process than object relatives, in both embedded and simple cases, and no interaction is found. This parallelism is found both at the global and the local level; at the local level it is found at all the relevant points. Hence the Parallelism Hypothesis is sustained. We conclude (i) that embedded subject extractions are not especially hard to process and (ii) that simple subject extractions are not especially easy to process, with respect to other extractions.

7.6.2 Implications for Processing and Grammar

In a model of processing based on a linguistic theory with empty categories, which makes use of a gap-filling mechanism, we can assume that the use of this mechanism increases processing complexity and therefore time. Hence (ii) above indicates that simple subject relatives do make use of this mechanism, and therefore do involve an extraction or string-vacuous movement. If the use of the slash-category is to be associated with the gap-filling mechanism, the GPSG treatment of simple subject relatives is not supported. However, this makes the assumption that it is syntactic feature manipulation, rather than, for instance, the lambda-bindings in unbounded dependencies, that correlates with processing complexity. A model based on a theory without empty categories or a gap-filling mechanism has no reason to assume that simple subject relatives would be processed differently from other relatives, and so is directly compatible with (ii).

The implication of (i) is that embedded subject relatives are not marked, and so their analysis should not involve particularly unusual or complex operations. The strength of this conclusion depends on how close or 'transparent' (Berwick and Weinberg 1984) a relationship we assume between linguistic theory and language processing. Even if we do assume a transparent relationship, it is not at all clear how we can measure the complexity of a linguistic operation. For example Slash Termination Metarule 2 in GPSG has to be applied only rarely, but the mechanism is not obviously more complex than Slash Termination Metarule 1 used for other extractions. On the other hand, syntactic rarity may well increase processing time (compare frequency effects in word recognition). It is reasonable to conclude that, because embedded subject extractions behave like other extractions on both local and global measures, they are not fundamentally distinct. Thus any unparsimonious treatment, as provided by GPSG or probably GB, is less attractive. Instead the linguistic analysis ought to fall out from general principles of unbounded dependencies.

Note that the results may be influenced by whether we can be sure what extraction has occurred at a given point in a sentence. For instance, after *the man that Mary loved*, it is possible that the sentence will continue *the man that Mary loved the book about arrived yesterday*. This is not possible after *the man that loved*, and so this might be a reason for the object relative to be harder to process than the subject relative. However, exactly the same pattern occurs with embedded extractions, and so we can conclude that both embedded and simple object relatives involve local ambiguity, but both embedded and simple subject relatives do not. Hence the parallelism is retained and so there is no possibility of a confound here.

Finally, we should consider the evidence from other languages and other constructions. The relative pervasiveness of the *that*-trace filter, and in particular its existence in almost all dialects of English, suggested that embedded subject extractions might be hard to process, because embedded subject extractions with or without *that* would be marked in some sense. But the results of the experiment suggest that such extractions are not marked, and are a completely normal part of English syntax. The conclusion is that there is no direct relationship between the embedded subject extractions considered in this paper and embedded subject extractions with an overt complementizer, and that the Fixed Subject Constraint is directly due to some peculiarity regarding embedded clauses with overt complementizers. The results are also not what might be expected from the cross-linguistic rarity of embedded subject extractions (assuming Comrie has made a valid generalization). We can conclude that the cross-linguistic rarity of a construction does not necessarily imply that this construction will be marked in a language that permits such a construction. In addition the result suggests that, whatever the explanation for the ease of processing simple subject extractions over simple object extractions, it is not solely that subject position is more accessible.

7.7 Subject Extraction and Dependency Constituency

7.7.1 Introduction

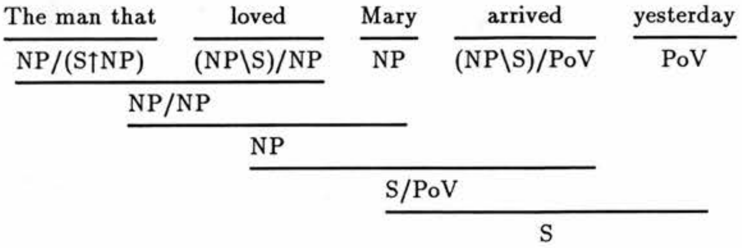
Because constituency theories do not directly allow incremental interpretation, the relation between grammar and parser must be indirect. Therefore we have to be careful assessing the effect that any processing result should have on their formulation. However, this difficulty indicates that such theories are ultimately of limited interest in the attempt to construct a grammar which assumes the Strong Competence Hypothesis. The results are important to constituency theories, however, if we make the attempt to convert them into incrementally plausible ones, as we found was possible with GPSG. The fact that the processing evidence supports parallelism is important as to what form the constituency grammar should take and how this conversion process should be framed.

Flexible categorial grammar on the other hand has a method of incremental processing built into it, so there is no difficulty relating it to processing evidence. We can therefore compare different theories directly, making some minimal assumptions, such as that extra processing load will in general increase processing time. In this section, we shall see how the Incremental Dependency Representations of the dependency preserving subset of $\mathbf{L}^{\uparrow}$ can be taken to model sentence processing. We shall also see how this contrasts with the comparable CCG account.

7.7.2 Modelling Subject Extraction

Let us consider the incremental processing of the four kinds of relative clause tested in the experiment within the dependency-preserving subset of $\mathbf{L}^{\uparrow}$. Consider a simple subject relative. We assume the type $\mathrm{PoV}/(\mathrm{S}{\uparrow}\mathrm{NP})$ for the relative clause introducer *that*, and treat *the man that* as a lexical item of type $\mathrm{NP}/(\mathrm{S}{\uparrow}\mathrm{NP})$ for simplicity:

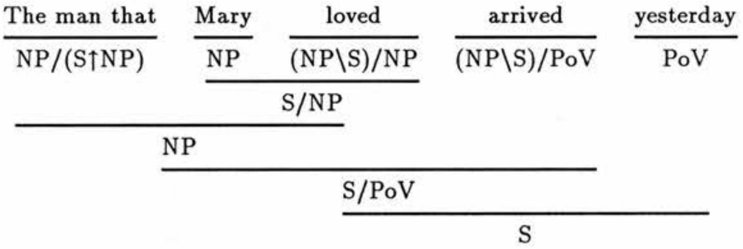
(7.34)



Every time we process another word of (7.34), we can construct a single dependency constituent. Hence the IDR is *(the man that)*, *(the man that loved)*, *(the man that loved Mary)*, *(the man that loved Mary arrived)*, *(the man that loved Mary arrived yesterday)*.

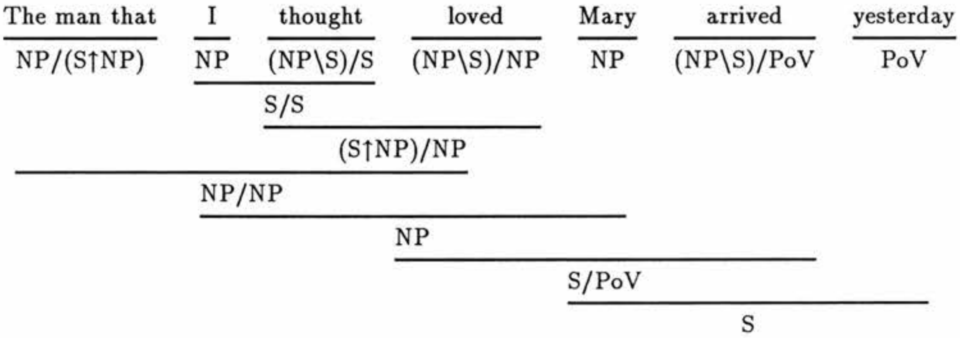
This contrasts with a simple object relative:

(7.35)

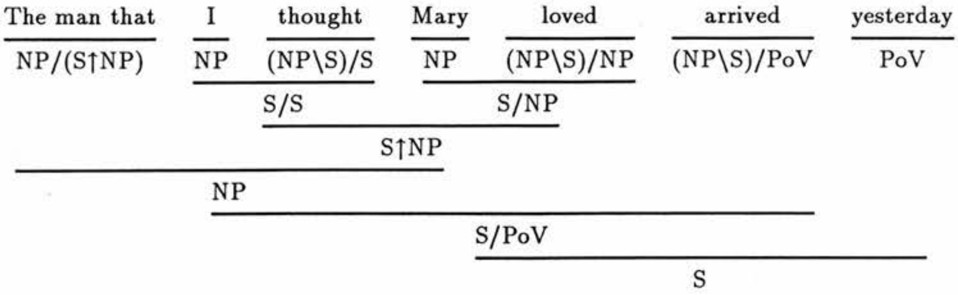


Here the string *The man that Mary* is not a dependency constituent. Therefore there is a point at which two dependency constituents have to be held in memory together. This might suggest a reason why (7.35) is harder to process than (7.34). But the important point however is found when contrasting these derivations with those of embedded subject and object relatives:

(7.36)



(7.37)



In both cases, the words *I thought* will not associate with *the man that.* Hence *the man that I thought* consists of two dependency constituents *(the man that)(I thought).* When we encounter *loved* in (7.36), we do form a single dependency constituent, just as in (7.34). But in (7.37), as in (7.35), we cannot form a single constituent on reading *Mary*, but again have to wait for the verb *loved.* After *Mary* we have three dependency constituents, *(the man that)(I thought)(Mary).* This again reduces to one dependency constituent at *loved.* Hence exactly the same processing pattern is expected for the embedded sentences as for the simple ones, as the Parallelism Hypothesis predicts.

We can contrast this with Steedman's analysis, which allows many fewer constituents to be formed incrementally. Steedman does not allow the formation of a constituent at *the man that loved* in (7.34). The reason for this, as we have seen, is that this allows many island constraint violations, such as **who did you see the man that loved* to be ruled out. Therefore in both simple subject and simple object relatives a single constituent can only be formed at the end of the relative clause. This rules out the explanation of why subject relatives are easier to process than object relatives, and of course has the general problem that it does not allow an incremental analysis of simple subject relatives. However, a more serious problem is that the embedded object relative allows the formation of a constituent *I thought*, whereas the embedded subject relative does not, because Steedman's analysis requires *loved Mary* to combine with *thought*, not *I thought.* Hence more constituents need to be held in memory in embedded subject relatives than embedded object relatives. This is likely to predict that embedded subject relatives should be harder to process than embedded object relatives, contrary to the parallelism hypothesis. This was refuted by the experimental data.

The important point is that a psychologically adequate system must represent dependency constituents directly. Such a system will therefore predict the parallelism hypothesis. This suggests that it is possible to make direct psychological predictions from linguistic theory, and that psychological evidence can constrain adequate linguistic analyses.

7.8 Conclusion

The processing evidence supports the Parallelism Hypothesis. There is no reason to regard simple subject extractions as different from other extractions or to regard embedded subject extractions as marked constructions. This has a number of implications, in particular for the grammatical analysis that these constructions should be given. A number of current linguistic treatments regard these constructions as very different from other extractions. The fact that their processing does not appear to be unusual suggests that that these highly specific treatments are not warranted. On the other hand, a system that directly represents dependency constituency does predict the parallelism hypothesis, and so the fact that it has been experimentally supported gives further evidence for the importance of dependency constituency. More generally, this chapter has suggested how psycholinguistic results can and should have an influence on the general form (if not the implementational details) of grammatical theory.

Chapter 8

Conclusion

This thesis has attempted to show that it is possible to construct a grammar which is at the same time linguistically motivated and psychologically plausible, and where all the units that can be interpreted are constituents of the grammar. We have seen that this is possible if the formation of dependency relations not the construction of traditional phrase structure trees is taken as fundamental. This allows us to use flexible rather than rigid constituents in linguistic analyses. We have seen how dependency constituents can be encoded in flexible categorial grammar in a very straightforward manner, and how this formulation allows the description of strings that can be coordinated in a way that contrasts with the inadequacy of phrase structure accounts.

The specific emphasis of much of the thesis has been on unbounded dependencies. Standard linguistic and psycholinguistic accounts have assumed that special mechanisms are needed for what is a rather restricted range of constructions. We have seen that transformational treatments, using empty categories, are both unparsimonious and indeed do not correspond well to the operations of the sentence processor. Instead, we have seen that unbounded dependencies can be processed in the same way as other constructions, and have described how this allows much apparently diverse experimental evidence to be brought together. The last chapter applied this approach to a particular kind of unbounded dependency.

This thesis has attempted to show how grammatical descriptions and language comprehension can be brought closer together, and can make use of the same set of primitives. Therefore the appropriate research methodology is to assume that particular theories of language can be discarded because of their incompatibility with either linguistic or processing evidence, of whatever kind. Whereas some experimental techniques may need considerable justification before any results should affect accounts of the structure of language, the fact that human sentence processing is incremental, faster than is possible in a model where all we do is interpret phrase structure constituents, cannot be seriously questioned. We have seen that this suggests that standard constituency assumptions will not serve in a parsimonious way as a component of a more general cognitive theory. Instead, the unit of analysis can be taken to be the dependency constituent. We can use this concept to develop a general and unified science of language and language use.

References

Ades, A.E. and Steedman, M.J. (1982). On the order of words. *Linguistics and Philosophy*, **4**, 517–558.

Ajdukiewicz, K. (1935). Uber die syntakische Konnexitat. *Studia Philosophia*, **1**, 1–27.

Altmann, G.T.M. (1987). Modularity and interaction in sentence processing. In Garfield, J. (ed.) *Modularity in Knowledge Representation and Natural Language Processing*. Cambridge, Mass.: MIT Press.

Altmann, G.T.M. (1988). Ambiguity, parsing strategies, and computational models. *Language and Cognitive Processes*, **3**, 73–98.

Altmann, G.T.M. (ed.) (1989). *Language and Cognitive Processes: Special Edition on Parsing and Interpretation*. London: Lawrence Erlbaum Associates Ltd.

Altmann, G.T.M. and Steedman, M.J. (1988). Interaction with context during human sentence processing. *Cognition*, **30**, 191–238.

Bach, E. (1977). Comments on the paper by Chomsky. In Cullicover, P., Wasow, T. and Akmajian, A. (1977). *Formal Syntax*. New York: Academic Press.

Bard, E.G., Shillcock, R.C. and Altmann, G.T.M. (1988) The recognition of words after their acoustic offsets in spontaneous speech: effects of subsequent context, *Perception and Psychophysics*, **44**, 395–408.

Bar-Hillel, Y. (1953). A quasi-arithmetical notation for syntactic description. *Language*, **29**, 47–58.

Barry, G. and Pickering, M. (1990). Dependency and constituency in categorial grammar. In G. Barry and G. Morrill (Eds.) *Edinburgh Working Papers in Cognitive Science, 5: Studies in Categorial Grammar.*

Berwick, R.C. and Weinberg, A. S. (1984). *The Grammatical Basis of Linguistic Performance: Language Use and Acquisition*. Cambridge, MA: MIT Press.

Bever, T.G. (1970). The cognitive basis for linguistic structures. In Hayes, J.R. (ed.) *Cognition and the Development of Language*. New York: Wiley.

Bever, T.G., Lackner, J.R. and Stolz, W. (1969). Transitional probability is not a general mechanism for the segmentation of speech. *Journal of Experimental Psychology*, **79**, 387–394.

Bever, T.G. and McElree, B. (1988). Empty categories access their antecedents during comprehension. *Linguistic Inquiry*, **19**, 35–43.

Bloomfield, L. (1933). **Language**. New York: Holt, Rinehart and Winston.

Bouma, G. (1989). Efficient processing of flexible categorial grammar. In *Proceedings of the Fourth Conference of the European Chapter of the ACL*, UMIST, Manchester.

Bresnan, J. (1972). *Theory of Complementation in English Syntax.* Doctoral dissertation, MIT. Published (1979), New York: Garland Press.

Bresnan, J. (1978). A realistic transformational grammar. In Halle, M., Bresnan, J. and Miller, G. *Linguistic Theory and Psychological Reality.* Cambridge, Mass.: MIT Press.

Bresnan, J. (ed.) (1982). *The Mental Representation of Grammatical Relations.* Cambridge, Mass.: MIT Press.

Briscoe, E.J. (1987). *Modelling Human Speech Comprehension: A Computational Approach.* Chichester: Ellis Horwood.

Briscoe, E.J. (1990). Prosodic resolution of syntactic ambiguity in unbounded dependencies. Paper presented at the LAGB meeting, Cambridge.

Burton-Roberts, N. (1986). *Analysing Sentences: An Introduction to English Syntax.* New York: Longman inc.

Chomsky, N. (1957). *Syntactic Structures.* The Hague: Mouton.

Chomsky, N. (1965). *Aspects of the Theory of Syntax.* Cambridge, MA: MIT Press.

Chomsky, N. (1970). Remarks on nominalization. In Jacobs, R. and Rosenbaum, P. (eds.) *Readings in English Transformational Grammar.* Waltham, Mass: Ginn and Co.

Chomsky, N. (1981). *Lectures on Government and Binding.* Dordrecht: Foris Publications.

Chomsky, N. (1986). *Knowledge of Language.* New York: Prieger.

Chomsky, N. and Lasnik, H. (1977). Filters and control. *Linguistic Inquiry, 8*, 425-504.

Clark, H. and Clark, E. (1977). *Psychology and Language: An Introduction to Psycholinguistics.* New York: Harcourt Brace Jovanovich.

Clifton, C. and Ferreira, F. (1989). Ambiguity in context. *Language and Cognitive Processes: Special Edition on Parsing and Interpretation.* London: Lawrence Erlbaum Associates Ltd.

Clifton, C. and Frazier, L. (1989). Comprehending sentences with long distance dependencies. In Carlson, G.N. and Tanenhaus, M.K. (eds.) *Linguistic Structure in Language Processing.* Dordrecht: Kluwer Academic Publishers.

Colombo, L. (1986). Activation and inhibition with orthographically similar words. *Journal of Experimental Psychology: Human Perception and Performance,* **12**, 226–234.

Comrie, B. (1981) *Language Universals and Linguistic Typology.* Oxford: Basil Blackwell.

Cowart, W. and Cairns, H. (1987). Evidence for an anaphoric mechanism within syntatic processing: some reference relations defy semantic and pragmatic constraints. *Memory and Cognition,* **15**, 318–331.

Crain, S. (1980). Pragmatic constraints on sentence comprehension. Unpublished Ph.D. dissertation, University of California at Irvine.

Crain, S. and Fodor, J. D. (1985). How can grammars help parsers? In Dowty, D. R., Karttunen, L. and Zwicky, A. M. (eds.) *Natural Language Parsing: Psychological, Computational and Theoretical Perspectives.* Cambridge: Cambridge University Press.

Crain, S. and Steedman, M.J. (1985). On not being led up the garden path: the use of context by the psychological parser. In Dowty, D. R., Karttunen, L. and

Zwicky, A. M. (eds.) *Natural Language Parsing: Psychological, Computational and Theoretical Perspectives.* Cambridge: Cambridge University Press.

Crocker, M.W. (1990). Principle-based sentence processing: a cross-linguistic account. Research Paper HCRC/RP-1, Human Communication Research Centre, University of Edinburgh.

Culy, C. (1985). The complexity of the vocabulary of Bambera. *Linguistics and Philosophy*, **8**, 345–351.

Curry, H. B. and Feys, R. (1958). *Combinatory Logic*, Volume 1. Amsterdam: North Holland.

Dowty, D. (1988). Type raising, functional composition, and non-constituent conjunction. In R. Oehrle, E. Bach and D. Wheeler (eds.), *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht.

Engdahl, E. (1983). Parasitic gaps. *Linguistics and Philosophy*, **6**, 5–34.

Engdahl, E. (1985). Parasitic gaps, resumptive pronouns and subject extractions. *Linguistics*, **23**, 3–44.

Fiengo, R.W. (1977). On trace theory. *Linguistic Inquiry*, **8**, 35–61.

Fodor, J.A. (1983). *The Modularity of Mind.* Cambridge, Mass.: MIT Press.

Fodor, J.A. and Bever, T.G. (1965). The psychological reality of linguistic segments. *Journal of Verbal Learning and Verbal Behavior*, **4**, 414–420.

Fodor, J.A., Bever, T.G. and Garrett, M.F. (1974). *The Psychology of Language: An Introduction to Psycholinguistics and Generative Grammar.* New York: McGraw-Hill.

Fodor, J. D. (1978). Parsing strategies and constraints on transformations. *Linguistic Inquiry*, **9**, 427-473.

Fodor, J. D. (1989). Empty categories in sentence processing. In G. T. M. Altmann (Ed.) *Language and Cognitive Processes: Special Edition on Parsing and Interpretation.* London: Lawrence Erlbaum Associates Ltd.

Fraser, N.M. (1989). Parsing and dependency grammar. In Carston, R. (ed.) UCL Working Papers in Linguistics Volume 1. University College London.

Frauenfelder, U., Segui, J. and Mehler, J. (1980). Monitoring around the relative clause. *Journal of Verbal Learning and Verbal Behaviour*, **19**, 328–337.

Frazier, L. (1978). On comprehending sentences: syntactic parsing strategies. Ph.D. Thesis, University of Connecticut. Indiana University Linguistics Club.

Frazier, L. (1987). Syntactic processing: evidence from Dutch. *Natural Language and Linguistic Theory*, **5**, 515–559.

Frazier, L., Clifton, C. and Randall, J. (1983). Filling Gaps: Decision Principles and structure in sentence comprehension. *Cognition*, **13**, 187-222.

Gaifman, H. (1965). Dependency systems and phrase-structure systems. *Information and Control*, **8**, 304–337.

Garnsey, S.M., Tanenhaus, M.K. and Chapman, R.M. (1989). Evoked potentials and the study of sentence comprehension. *Journal of Psycholinguistic Research*, **18**, 51–60.

Garrett, M.F., Bever, T.G. and Fodor, J.A. (1966). The active use of grammar in speech perception. *Perception and Psychophysics*, **1**, 30–32.

Gazdar, G. (1981). Unbounded dependencies and coordinate structure. *Linguistic Inquiry*, **12**, 155–84.

Gazdar, G. (1982). Phrase structure grammar. In Jacobson, P. and Pullum, G.K. (eds.) *The Nature of Syntactic Representation*. Dordrecht: Reidel.

Gazdar, G., Klein, E., Pullum, G. and Sag, I. (1985). *Generalized Phrase Structure Grammar*. Cambridge, MA: Harvard University Press.

Gorrell, P. (1989). Syntactic features and syntactic processing. Paper presented to the Department of Psychology, University of Arizona.

Greenberg, J.H. (1966). Some universals of grammar with particular reference to the order of meaningful elements. In Greenberg, J.H. (ed.) *Universals of Language*, 2nd Edition. Cambridge, Mass.: MIT Press.

Hawkins, J.A. (1983). *Word order universals.* New York: Academic Press.

Hudson, R.A. (1984). *Word Grammar.* Oxford: Basil Blackwell.

Hudson, R.A. (1987). Zwicky on heads. *Journal of Linguistics 23*, 109–132.

Hudson, R.A. (1990). The grammarian's world. Inaugural Lecture as Professor of Linguistics, University of London.

Jackendoff, R. (1977). $\overline{X}$ *Syntax: a Study of Phrase Structure.* Cambridge, Mass.: MIT Press.

Kaplan, R.M. and Bresnan, J. (1982). Lexical-functional grammar: a formal system for grammatical representation. In Bresnan, J. (ed.) *The Mental Representation of Grammatical Relations.* Cambridge, Mass.: MIT Press.

Kaplan, R.M., and Zaenen, A. (1988). Long-distance dependencies as a case of functional uncertainty. In Baltin, M. and Kroch, A. (eds.) *Alternative Conceptions of Phrase Structure*, University of Chicago Press, Chicago, Illinois.

Karttunen, L. (1986). Radical lexicalism. In Baltin, M. and Kroch, A. (eds.) *Alternative Conceptions of Phrase Structure*, University of Chicago Press, Chicago, Illinois.

Keenan, E. and Comrie, B. (1977) Noun phrase accessibility and universal grammar. *Linguistic Inquiry*, **8**, 63–100.

Kimball, J. (1973). Seven principles of surface structure parsing in natural language. *Cognition*, **2**, 15–47.

Lambek, J. (1958). The Mathematics of Sentence Structure. *American Mathematical Monthly*, **65**, 154–170.

Ladefoged, P. and Broadbent, D.E. (1960). Perception of sequence in auditory events. *Quarterly Journal of Experimental Psychology*, **13**, 162–170.

Levelt, W.J.M. (1970). Hierarchical clustering algorithm in the psychology of grammar. In Flores D'Arcais, G.B. and Levelt, W.J.M. (eds.) *Advances in Psycholinguistics.* Amsterdam: North-Holland Publishing.

MacDonald, M.C. (1989). Priming effects from gaps to antecedents. *Language and Cognitive Processes*, **4**, 35–56.

McElree, B. and Bever, T. G. (1989). The psychological reality of linguistically defined gaps. *Journal of Psycholinguistic Research*, **18**, 21–35.

Maling, J. and Zaenen, A. (1978) The non-universality of a surface filter. *Linguistic Inquiry*, **9**, 475–497.

Maling, J. and Zaenen, A. (1982). A phrase structure account of Scandinavian extraction phenomena. In Jacobson, P. and Pullum, G.K. (eds.) *The Nature of Syntactic Representation*. Dordrecht: Reidel.

Matthews, P.H. (1981). *Syntax*. Cambridge: Cambridge University Press.

Marslen-Wilson, W.D. (1973). Linguistic structure and speech shadowing at very short latencies. *Nature*, **244**, 522–3.

Marslen-Wilson, W.D. and Tyler, L. (1980). The temporal structure of spoken language understanding. *Cognition*, **8**, 1–74.

Meyer, D.E. and Schvaneveldt, R.W. (1971). Facilitation in recognizing pairs of words: evidence of a dependence between retrieval operations. *Journal of Experimental Psychology*, **90**, 227–35.

Miller, J. (1985). *Semantics and Syntax: Parallels and Connections*. Cambridge: Cambridge University Press.

Mitchell, D.C. (1987). Lexical guidance in human parsing: locus and processing characteristics. In Coltheart, M. (ed.) *Attention and Performance XII*. Hillsdale, New Jersey: Lawrence Erlbaum Associates Inc.

Mitchell, D.C. (1989). Verb-guidance and other lexical effects in parsing. In G.T.M. Altmann (ed.) *Language and Cognitive Processes: Special Edition on Parsing and Interpretation*. London: Lawrence Erlbaum Associates Ltd.

Montague, R. (1973). The proper treatment of quantification in ordinary English. In Hintikka, J., Moravcsik, J.M.E. and Suppes, P. (eds.) *Approaches to Natural Language*. Dordrecht: D. Reidel.

Moortgat, M. (1988a). *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus.* Dordrecht: Foris Publications.

Moortgat, M. (1988b). Mixed composition and discontinuous dependencies. In R. Oehrle, E. Bach and D. Wheeler (Eds), *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht.

Morrill, G. (1988). Extraction and coordination in phrase structure grammar and categorial grammar. Unpublished Ph.D. Thesis, Centre for Cognitive Science, University of Edinburgh.

Morrill, G. (1990). Grammar and logical types. In Barry, G. and Morrill, G. (eds.) *Edinburgh Working Papers in Cognitive Science, 5: Studies in Categorial Grammar.*

Morrill, G., Leslie, N., Hepple, M. and Barry, G. (1990). Categorial deductions and structural operations. In Barry, G. and Morrill, G. (eds.) *Edinburgh Working Papers in Cognitive Science, 5: Studies in Categorial Grammar.*

Nicol, J. (1989). What the parser knows about the grammar: psycholinguistic evidence. Proceedings of WCCFL VIII.

Nicol, J. and Swinney, D. (1989). The role of structure in coreference assignment during sentence comprehension. *Journal of Psycholinguistic Research*, **18**, 5–19.

Perlmutter, D. M. (ed.) (1983) *Studies in Relational Grammar 1.* Chicago: University of Chicago Press.

Perlmutter, D. M. and Rosen, C. (eds.) *Studies in Relational Grammar 2.* Chicago: University of Chicago Press.

Pickering, M. and Barry, G. (1989). Processing extractions without gaps. Research Paper No. EUCCS/RP-36, Centre for Cognitive Science, University of Edinburgh.

Pickering, M. and Barry, G. (1990). Sentence processing without empty categories. Unpublished ms., Centre for Cognitive Science, University of Edinburgh.

Pollard, C. and Sag, I. (1987). *An Information-Based Approach to Syntax and Semantics: Volume 1 Fundamentals.* Stanford, Ca.: Center for the Study of Language and Information.

Postal, P. M. (1974). *On Raising.* Cambridge, Mass.: MIT Press.

Pullum, G. and Gazdar, G. (1982). Natural language and context-free languages. *Linguistics and Philosophy*, **4**, 471–504.

Pritchett, B.L. (1988). Garden path phenomena and the grammatical basis of language processing. *Language*, **64**, 539–76.

Pulman, S.G. (1986). Grammars, parsers and memory limitations. *Language and Cognitive Processes*, **1**, 197–225.

Pulman, S.G. (1987). Computational models of parsing. In Ellis, A. (ed) *Progress in the Psychology of Language*, Volume 3. Hillsdale, NJ.: Lawrence Erlbaum.

Radford, A. (1988). *Transformational Grammar: A First Course.* Cambridge: Cambridge University Press.

Reinhart, T. (1976). The syntactic domain of anaphora. Unpublished Ph.D. Dissertation, MIT.

Robinson, J.J. (1970). Dependency structures and transformational rules. *Language*, **46**, 259–285.

Ross, J. R. (1967). Constraints on variables in syntax. PhD Thesis, MIT. Indiana University Linguistics Club.

Sag, I., Gazdar, G., Wasow, T. and Weisler, S. (1985). Coordination and how to distinguish categories. *Natural Language and Linguistic Theory*, **3**, 117–171.

Sampson, G. (1975) The single mother condition. *Journal of Linguistics*, **11**, 1–12.

Shieber, S. (1985). Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, **8**, 333–343.

Shieber, S. (1986). *An Introduction to Unification Approaches to Grammar.* CSLI Lecture Notes 4. Stanford: CSLI.

Shillcock, R.C. (1984). The processing of syntactic dependencies. Unpublished doctoral dissertation, University of Sussex.

Sobin, N. (1987). The variable status of COMP-trace phenomena. *Natural Language and Linguistic Theory*, **5**, 33–60.

Steedman, M.J. (1985). Dependency and coordination in the grammar of Dutch and English. *Language*, **61**, 523–68.

Steedman, M.J. (1987). Combinatory grammars and parasitic gaps. *Natural Language and Linguistic Theory* **5**, 403–439.

Steedman, M. (1988) Combinators and grammars. In Oehrle, R., Bach, E. and Wheeler, D. (eds.) *Categorial Grammars and Natural Language Structures.* Dordrecht: D. Reidel.

Steedman, M.J. (1989). Grammar, interpretation and processing from the lexicon. In W. Marslen-Wilson (Ed.), *Lexical Representation and Process*, MIT Press, Cambridge, Massachusetts.

Steedman, M.J. (1990). Syntax and intonational structure in a combinatory grammar. To appear in G.T.M. Altmann (ed.), *Cognitive Models of Speech Processing: Psycholinguistic and Computational Perspectives*, MIT Press, Cambridge, Massachusetts.

Steedman, M.J. and Altmann, G.T.M. (1989). Ambiguity in context: a reply. *Language and Cognitive Processes: Special Edition on Parsing and Interpretation.* London: Lawrence Erlbaum Associates Ltd.

Stowe, L.A. (1986). Parsing WH-constructions: Evidence for on-line gap location. *Language and Cognitive Processes*, **1**, 227–245.

Stowe, L.A. (1989). Thematic structures and sentence comprehension. In Carlson, G.N. and Tanenhaus, M.K. (ed.) *Linguistic Structure in Language Comprehension.* Dordrecht: Kluwer Academic.

Stowe, L.A. and Cupples, L. (1989). Animacy and reduced relative clauses. Poster at the conference on sentence processing, City University of New York.

Swinney, D. (1979). Lexical access during sentence comprehension: (re)consideration of context effects. *Journal of Verbal Learning and Verbal Behavior*, **15**, 545–69.

Swinney, D., Ford, M., Frauenfelder, U. and Bresnan, J. (1988). On the Temporal Course of Gap-Filling and Antecedent Assignment during Sentence Comprehension. In Grosz, B., Kaplan, R., Macken, M. and Sag, I. (eds.) *Language Structure and Processing*. Stanford: CSLI.

Tanenhaus, M.K. and Carlson, G.N. (1989). Lexical structure and language comprehension. In Marslen-Wilson, W.D. (ed.) *Lexical Representation and Process*. Cambridge, Mass.: MIT Press.

Tanenhaus, M.K., Carlson, G.N. and Seidenberg, M. (1985). Do listeners compute linguistic representations? In Dowty, D. R., Karttunen, L. and Zwicky, A. M. (eds.) *Natural Language Parsing: Psychological, Computational and Theoretical Perspectives*. Cambridge: Cambridge University Press.

Tanenhaus, M. K., Garnsey, S. M. and Boland, J. (1989). Combinatory Lexical Information and Language Comprehension 1. To appear in Altmann, G. (ed.) *Cognitive Models of Speech Processing: Psycholinguistic and Computational Perspectives*. Cambridge, Mass.: MIT Press.

Uszkoreit, H. (1986). Linear precedence in discontinuous constituents: complex fronting in German. CSLI–86–47. Stanford: CSLI.

Wanner, E. (1987). The parser's architecture. In Kessel, F. (ed.) *The development of Language and Language Researchers: Essays in Honor of Roger Brown*. Hillsdale, New Jersey: Lawrence Erlbaum Associates Inc.

Wanner, E. and Maratsos, M. (1978). An ATN Approach to Comprehension. In Halle, M., Bresnan, J. and Miller, G. A. (eds.) *Linguistic Theory and Psychological Reality*. Cambridge, MA: MIT Press.

Wexler, K. and Cullicover, P. (1980). *Formal Principles of Language Acquisition*. Cambridge, Mass.: MIT Press.

Whitelock, P.J. (1988). A feature-based categorial morpho-syntax for Japanese. In Reyle, U. and Rohrer, C. (eds.) *Natural Language Parsing and Linguistic Theories*. Dordrecht: Reidel.

Wood, M. (1988). *A Categorial Syntax for Coordinate Constructions.* Unpublished Ph.D. thesis, University College London. Available as Technical Report UMCS–89–2–1, Department of Computer Science, University of Manchester.

Woolford, E. (1987). An ECP Account of Constraints on Across-the-board extraction. *Linguistic Inquiry, 18,* 166-171.

Wright, B. and Garrett, M. (1984). Lexical decision in sentences: effects of syntactic structure. *Memory and Cognition*, **12**, 31–45.

Zwicky, A.M. (1985). Heads. *Journal of Linguistics*, **21**, 1–29.

Appendix

The manager that I swear appointed Frederick is in charge of many people.

The celebrity that they think directed Philippa is appearing on the show.

The relative that I think escorted Roderick lives in Bristol in a flat.

The teenager that you swear resented Jonathon wants to catch a bus now.

The woman that we thought attended Samantha was wearing a new dress.

The child that they swore detested Rosalind was on holiday by the sea.

The aunt that you thought suspected Geraldine boarded the Liner for France.

The policeman that they swore Elizabeth indicated walked to the angry crowd.

The researcher that I swear fled John is a lecturer at the university.

The footballer that we think prodded Jeffrey is well known at the club.

The neighbour that we think noticed Michael gives a lot to charity.

The colleague that they swore envied Jackie owns a rather large dog.

The teacher that I swore disliked Michelle was at the play at the school.

The sportsman that you thought hated Karen was having a very bad season.

The actress that we thought loved Clive arrived in town this morning.

The lady that you swore adored Philip stayed at home all weekend.

The gentleman that you hope demoted Claudia is eating a large meal.

The uncle that we realise deserted Nicholas is standing by the window.

The girl that you realise attacked Margaret spends the summers in the country.

The writer that we hope respected Gabrielle runs a few miles every day.

The man that they realised disciplined Christopher was popular in the firm.

The villager that we hoped rewarded Benjamin was waiting for some guests.

The chef that I realised telephoned Jacqueline married a widow last month.

The friend that they hoped Timothy avoided entered suddenly by the door.

The person that they realise annoyed Matthew is a newcomer in the department.

The fellow that I hope aided Carol is over there by the table.

The individual that I hope spotted Raymond hurries away on most evenings.

The youth that you realise accused Stewart works most nights in a shop.

The assistant that we realised cited Brian is a student at the college.

The stranger that they hoped bribed George was in court this afternoon.

The pupil that I realised teased Claire joined the group last term.

The farmer that you hoped praised Charles wandered along to the pub.

Processing Extractions without Gaps

Martin Pickering and Guy Barry
Centre for Cognitive Science
University of Edinburgh
2 Buccleuch Place
Edinburgh EH8 9LW
United Kingdom

Research Paper EUCCS/RP-36
September 1989

Centre for Cognitive Science
University of Edinburgh
2 Buccleuch Place
Edinburgh EH8 9LW
United Kingdom

Abstract

Most psycholinguistic theories have traditionally made use of the transformational grammar notions of filler and gap in explaining the processing of constructions involving extraction. On the assumption that recursive nesting of dependencies must ultimately cause processing difficulty, we show that standard analyses using gaps make incorrect predictions for a class of sentences involving pied-piping (and that alternative analyses using gaps are deficient in other respects). From this we argue that the processor establishes dependency relations between filler and subcategorizer rather than between filler and gap, and demonstrate that the use of a theory of syntax without gaps, such as a flexible categorial grammar, yields a more appropriate processing model. We show how this approach can be used to describe the processing of centre-embedded constructions, and integrated with experimental results on long-distance dependencies.

1 Processing Extractions with Gaps

1.1 Extractions and Gaps

The traditional view of constructions such as relative clauses, constituent questions and topicalizations is that they all involve deviation from some canonical form. Thus consider (1) below:

(1) [John]$_1$, Mary loves $\emptyset_1$.

Here the extracted element, the NP *John*, is found after *loves* in the canonical sentence *Mary loves John.* Thus, the grammar specifies that there is a gap, represented here by $\emptyset$, at this location, and that the extracted element is linked with this gap by coindexing. In a sense, there are two distinct syntactic components in the analysis of such constructions; the coindexing and the usual relationship between the verb and its object NP argument. This coindexing is similar to the relationship between a pronoun or other anaphoric element and its antecedent. Notice that the canonical form in this case is a grammatical sentence, but this is not always the case, as with embedded questions:

(2) I wonder [which man]$_1$ Mary loves $\emptyset_1$.

Here, *I wonder Mary loves which man* is ungrammatical. Hence the notion of canonicality must be taken to refer to argument positions, rather than word order in a canonical sentence.

In psycholinguistic terms, the extracted element (or some of the attributes of the extracted element) is usually treated as a 'filler', which is held in memory until the gap is located, at which point it is recalled and associated with the gap location. This association can alternatively be viewed as the establishment of an anaphoric link. Note immediately that this can only serve as an explanation for leftward extraction cases; it is unclear how to develop a treatment of, for instance, heavy NP shift in this manner. The processing of left-extracted elements is analogous to their syntactic analysis; there have to be both filler-retention and gap-location mechanisms. Gap location may be complicated by on-line ambiguity, but once the gap is found, the filler can be linked with the gap by a procedural mechanism analogous to the coindexing process, and then the parse can proceed in effect as though the filler (or its relevant attributes) had always been located at the site of the gap. This is essentially the way that unbounded elements are treated in ATN models of processing; they are placed in a special memory store known as a HOLD cell (Wanner and Maratsos 1978).

Both linguistically and psycholinguistically this type of treatment seems unparsimonious. We have to posit a special mechanism for dealing with one relatively unusual kind of construction, which may be, in any case, not found in all languages. In a manner dissimilar to that used in the processing of other constructions, we have to establish a link between the filler and the gap before we are able to begin an interpretation based on the construction of a dependency relation between the subcategorizer and a now meaningful gap. What is usually called a 'filler-gap dependency' is therefore seen as an anaphoric association between an extracted element and a phonologically unrealised entity.

In this paper we shall put forward the alternative view that a dependency is established between the extracted element and its subcategorizer directly, and therefore that no gaps are needed. We shall argue that such an account is not only parsimonious but is also an empirically more adequate basis for an account of processing.

1.2 Remembering Several Fillers

Sentences with multiple extractions involve more than one gap in traditional analyses, and coindexing disambiguates which extracted element is associated with which gap. One such sentence-type is formed by making a constituent question from a 'tough-construction' (Fodor 1978):

(3) [Which pot]$_1$ is [this rice]$_2$ easy to cook $\emptyset_2$ in $\emptyset_1$?

It is generally found that this sentence is quite hard to process. One possible cause of its difficulty, in filler-gap terms, is that both fillers, *which pot* and *the rice*, have to be held in memory together. Thus our memory system must either have multiple special memory stores or HOLD cells, or have one structured store. We shall assume that the store is structured, and, following Fodor, that it is structured as a stack (i.e. organized on a last-in first-out principle). This allows the processor to deal with nested dependencies.[1] Another case involving nested multiple extractions is illustrated by centre-embedded constructions such as (4), which we shall refer to as *multiple object relative* constructions:

(4) The cat which the dog which the farmer owned chased fled.

Such sentences are notoriously difficult. We shall assume that this is a processing problem, but not a grammatical one. A closely related case of centre-embedding is the *multiple preposition stranding* construction, exemplified in (5):

(5) John found the box [which]$_1$ I put the tray [which]$_2$ Mary placed the tin on $\emptyset_2$ in $\emptyset_1$.

Although there are of course other explanations for the difficulty of processing centre-embedded constructions, the fact that we have to hold two fillers in memory should, as with (3), contribute to their difficulty. The dependency pattern is again nested: using F for filler and G for gap, we can depict the pattern as F_1 F_2 G_2 G_1. Let us make the assumption that, *at some point*, memory limitations must make multiple nesting dependencies difficult to process because of the number of fillers that need to be held together before any gap is found. This follows from the simple assumption that resources are finite. In fact, the evidence suggests that any more than one extraction is problematic. A sentence-type which can

---

[1] We shall ignore the issue of crossing dependencies throughout this discussion, though the structure of the store could be modified to allow for these.

have indefinitely many nesting dependencies without creating processing difficulties would consequently be an impossibility.

Disjoint dependency patterns, where each filler is associated with a gap before the next filler is encountered, are not surprisingly in general easy to process. Consider:

(6) The man [who]$_1$ John knows $\emptyset_1$ met the woman [who]$_2$ Mary is afraid of $\emptyset_2$ in the park [where]$_3$ Bill goes $\emptyset_3$.

The filler-gap pattern is F_1 G_1 F_2 G_2 F_3 G_3, hence no overlapping of dependencies. Another example is the *multiple subject relative* construction (7):

(7) I saw the farmer [who]$_1$ $\emptyset_1$ owned the dog [which]$_2$ $\emptyset_2$ chased the cat.

These sentences can be extended in the same manner without ever causing processing problems, for instance:

(8) I saw the farmer [who]$_1$ $\emptyset_1$ owned the dog [which]$_2$ $\emptyset_2$ chased the cat [which]$_3$ $\emptyset_3$ ate the mouse [which]$_4$ $\emptyset_4$ nibbled the cheese.

This is very unlike the nesting dependency examples (3) and (5) above. The intonation patterns are also quite different, with (6), (7) and (8) displaying an evenness which contrasts sharply with the intonation for nesting dependency constructions. The conclusion that we can draw is that these sentences are easy to process at least partly because the stack of fillers never needs to get beyond a depth of one; in (3) and (5), the stack grows beyond this depth.

This account presupposes the existence of gaps, and that fillers are associated with these gaps. The alternative suggested above is that the extracted element is associated with the subcategorizer directly; we shall notate this in the following examples by coindexing the filler and the subcategorizer, rather than the filler and the gap. We can also regard this as representing a component of the dependency relations of the sentence. Non-local dependencies are therefore treated in a similar way to local dependencies. Thus, for instance, (3) becomes (9), and (5) becomes (10):

(9) [Which pot]$_1$ is [this rice]$_2$ easy [to cook]$_2$ [in]$_1$?

(10) John found the box [which]$_1$ I put the jar [which]$_2$ I put the tin [on]$_2$ [in]$_1$.

This makes no change to the predictions. Both of these sentences use nested dependency patterns in exactly the same way as before. When gaps are posited, they are adjacent to the subcategorizers, so there is little difference between the structures of the sentences. However, there is no longer a need for a special mechanism for dealing with filler-gap dependencies. As we shall demonstrate in section 2, it is possible to construct a processing model which treats all dependencies with a single stack mechanism. Let us rewrite (6), say, as (11):

(11) The man [who]$_1$ John [knows]$_1$ met the woman [who]$_2$ Mary is afraid [of]$_2$ in the park [where]$_3$ Bill [goes]$_3$.

The dependency pattern remains disjoint. And now the difficulty of nested unbounded dependencies can be explained in the same way as other nested dependencies. Contrast the English and German sentences below, which have the same meaning, but where only the German sentence is hard to process, as pointed out by Bach, Brown and Marslen-Wilson (1986):

(12) Johanna has [helped]$_1$ [the men]$_1$ [teach]$_2$ [Hans]$_2$ [to feed]$_3$ [the horses]$_3$.

(13) Johanna hat [den Männern]$_1$ [Hans]$_2$ [die Pferde]$_3$ [füttern]$_3$ [lehren]$_2$ [geholfen]$_1$.
Johanna$_{NOM}$ has the men$_{DAT}$ Hans$_{ACC}$ the horses$_{ACC}$ to-feed to-teach helped.

This seems to be an obvious simplification. The important issue, however, is whether we can empirically distinguish an account using gaps from an account without gaps.

1.3 Multiple Pied-Piping

Consider sentence (14) below:

(14) John found the box in which I put the tray on which Mary placed the tin.

By analogy with previous definitions, we shall refer to this construction[2] as *multiple pied-piping*. The phrase-structural analysis of this sentence requires there to be two gaps at the end of the sentence, one corresponding to *on which*, one to *in which*. *The tray on which Mary placed the tin* is a complex NP argument to the first verb *put*. Hence, as shown below in (15), the gap-analysis gives us a nesting pattern of (filler-gap) dependencies:

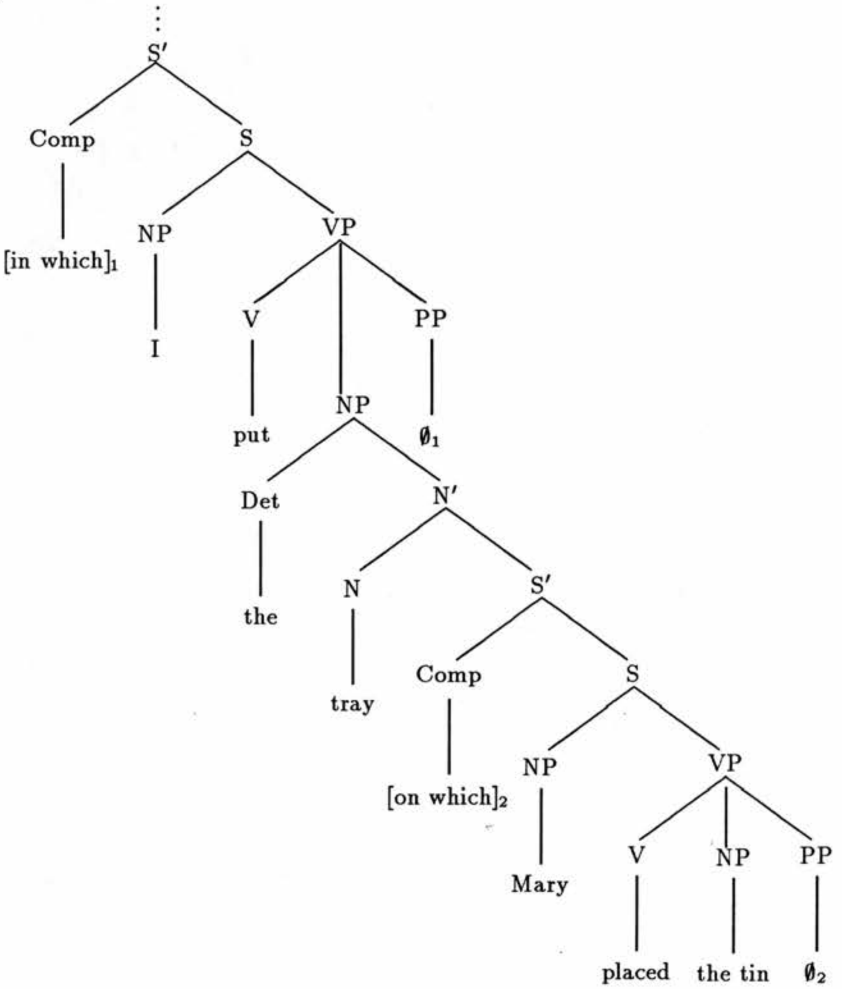
(15) John found the box [in which]$_1$ I put the tray [on which]$_2$ Mary placed the tin
$\emptyset_2$ $\emptyset_1$.

We give a traditional phrase-structural analysis in (16).

---

[2] Here we assume the obvious reading where *on which Mary placed the tin* modifies *tray*; we are not concerned with a possible alternative reading where it modifies *box*.

(16)



This structure immediately appears strange. The sentence is not hard in the way that we expect for a sentence with nested dependencies. Of course, it is not a centre-embedded sentence, and so we might conclude that stacking of fillers is actually unrelated to processing difficulty, the difficulty being due to other characteristics of the centre-embedded construction. But this would then require us to abandon the principle that fillers are placed in some memory store and then associated with the gap location. It could of course be that the permitted stack depth is simply much greater than two. However, notice that (14) can be extended indefinitely:

(17) John found the box [in which]$_1$ I put the tray [on which]$_2$ Mary placed the tin [into which]$_3$ Bill dropped the packet [in which]$_4$ Fred wrapped the matchbox [from which]$_5$ Sue took the matches $\emptyset_5$ $\emptyset_4$ $\emptyset_3$ $\emptyset_2$ $\emptyset_1$.

This sentence now has a series of five gaps at the end, arranged in a nested pattern F_1 F_2 F_3 F_4 F_5 G_5 G_4 G_3 G_2 G_1. At some point this must exceed our stack depth limitation. Yet the sentence appears to be processed in the same way that, for instance, the 'right-branching' sentence (7) is processed. The intonation patterns again reflect this judgement. There is no sense in which we are building up a stack of fillers. Its dependencies are clearly disjoint. Notice the contrast with the preposition stranding sentence (5), which behaves as though its dependencies are nested.

When sentence (17), however, is reanalysed so that the extracted element is associated directly with the subcategorizer, in each case one of the five subordinate verbs, we find that the dependency pattern changes from nested to disjoint, which is, as we have argued above, the only plausible pattern for the sentence:

(18) John found the box [in which]$_1$ I [put]$_1$ the tray [on which]$_2$ Mary [placed]$_2$ the tin [into which]$_3$ Bill [dropped]$_3$ the packet [in which]$_4$ Fred [wrapped]$_4$ the matchbox [from which]$_5$ Sue [took]$_5$ the matches.

In other words, the filler-gap account predicts that the dependency pattern is nested in both cases, whereas the alternative account, where the 'filler' is associated directly with the subcategorizer, makes the preposition stranding sentence (5) have a nesting pattern but the multiple pied-piping sentence (14) have a disjoint pattern. This alternative, no-gap, account accords with the processing evidence.

We shall now describe a model of grammar and associated processing account that can be taken to assume filler-subcategorizer association. First, however, it is worth describing why multiple pied-piping sentences of this kind are so interesting. In most unbounded dependency constructions, such as (1) and (2) above, as well as the preposition stranding sentence (5), the hypothesised gap location is adjacent to the subcategorizer of the unbounded element; hence it would be very hard to distinguish between a process linking this element to the gap from one linking it to the subcategorizer. However, with a verb taking two post-verbal arguments, such as *put*, we can extract the second argument and then arbitrarily extend the first argument so that the verb and the gap are far apart. We can then, so to speak, embed this process on the first argument. The point is that there seems to be no reason for the processor to postulate some reconstruction of canonical form; rather the filler can be linked directly to its subcategorizer. The processing evidence gives support to this argument.

2 Processing Extractions in Categorial Grammar

2.1 Categorial Grammars

A *categorial grammar* is one in which syntactic categories are treated as functions and arguments. For example, in *I saw Mary* the verb *saw* can be thought of as a function mapping the two noun phrases *Mary* and *I* into a sentence. All categorial grammars have two components: a categorial lexicon, which associates one or more categories with each word of the given language, and a set of so-called *combinatory rules* for combining categories. Unlike phrase-structure rules, combinatory rules are (usually) very general schemas, whereas individual syntactic categories are often structurally quite complex, so that most of this information about the combining properties of words is stored in the lexicon rather than in grammar rules.

A *flexible* categorial grammar is one that can assign several syntactically distinct but semantically equivalent derivations to the same string. Flexible categorial grammars are of psychological interest because they are capable of assigning left-branching structures to strings that standardly receive right-branching structures in other grammatical theories, allowing the possibility of incremental interpretation, and of linguistic interest because they offer the possibility of uniform accounts of such phenomena as extraction and coordination. In particular, most flexible categorial grammars eschew the notion of 'empty category' or 'gap', so that the syntactic account of the phenomena described in section 1 is completely different.

We shall look first at a non-flexible categorial grammar, so-called *classical* categorial grammar, before describing a flexible extension of the same grammar, the Combinatory Categorial Grammar (CCG) of Steedman (1987a).

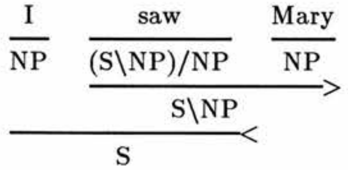
2.1.1 Classical Categorial Grammar

In classical (bidirectional) categorial grammar, category symbols are constructed from a set of basic categories and the two connectives /, \. For the purposes of this paper we shall assume that the basic categories are S (sentence), NP (noun phrase), N (common noun) and PP (prepositional phrase). For any two categories X, Y (simple or complex), X/Y is the category of strings that combine with a following adjacent string of category Y to form a string of category X, and $X\backslash Y$ is the category of strings that combine with a preceding adjacent string of category Y to form a string of category X. This can be stated concisely by means of the following two combinatory rules:

Forward application
$X/Y + Y \Rightarrow X$

Backward application
$Y + X\backslash Y \Rightarrow X$

The terms *forward application* and *backward application* refer to the semantic interpretation of the two rules as function application. For readers used to phrase-structure grammars the two above rules may appear to be the wrong way round, in that the 'mother' category appears on the right-hand side of the rule and the 'daughters' on the left-hand side. This has no real theoretical significance, but the notation is standard in the categorial grammar literature, and reflects the intuition that categorial derivations proceed from lexical categories to phrasal ones. For example, the derivation of the sentence *I saw Mary* would proceed as follows:



We use Steedman's (1987a) notation for derivations. When two categories are combined, a line is drawn underneath both of them and indexed with a mnemonic symbol to indicate the rule that was used: > for forward application, < for backward application. The resultant category is then written below the line.

Classical categorial grammar corresponds (roughly) to the assignment of structures in a traditional context-free phrase-structure grammar, so that its linguistic coverage is necessarily rather limited.

2.1.2 Combinatory Categorial Grammar

CCG is an extension of classical categorial grammar that was developed with the psychological motivation of allowing incremental derivations, and the linguistic motivation of dealing with constructions such as extraction and non-constituent coordination. The full apparatus of

CCG is described in Steedman (1987a); here we shall present only the fragment that allows us to derive the constructions discussed in section 1.

The relevant fragment of CCG employs the following four binary combinatory rules:

Forward application (indexed >)
$X/Y + Y \Rightarrow X$

Backward application (indexed <)
$Y + X\backslash Y \Rightarrow X$

Forward (harmonic) composition (indexed $>\mathbf{B}$)
$X/Y + Y/Z \Rightarrow X/Z$

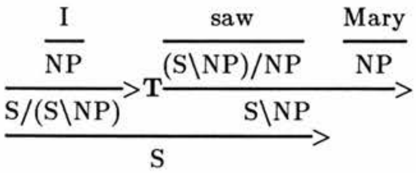
Forward (harmonic) generalized composition, order 2 (indexed $>\mathbf{B}^2$)
$X/Y + (Y/Z)/W \Rightarrow (X/Z)/W$

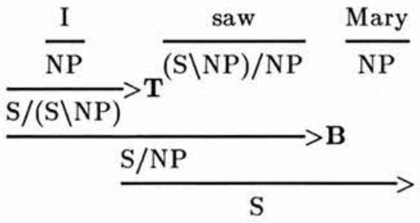
In addition to these binary rules, we include the following unary rule, which we limit to two special cases:

Forward type-raising (indexed $>\mathbf{T}$)
a. Subject type-raising: $\mathrm{NP} \Rightarrow \mathrm{S}/(\mathrm{S}\backslash\mathrm{NP})$
b. Noun type-raising over adjuncts: $\mathrm{N} \Rightarrow \mathrm{N}/(\mathrm{N}\backslash\mathrm{N})$

The rule indices indicate both the directionality of the rule and the associated semantic operation, the latter in terms of one of the *combinators* of Curry and Feys (1958); for a discussion of combinators in natural language semantics see Steedman (1987b).

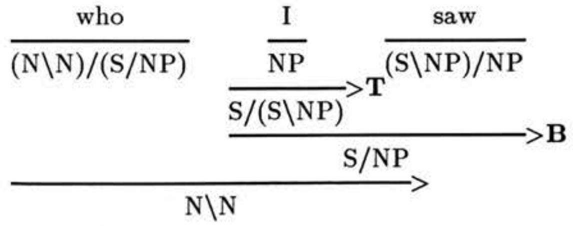
These rules allow structures other than standard right-branching ones to be assigned to strings. For example, the sentence *I saw Mary* has three possible derivations in CCG: the classical derivation given above, and the following two:





The classical derivation and the first derivation above both correspond to the traditional right-branching phrase-structure analysis; the second derivation above gives a left-branching structure to the same sentence. But the semantics of the combinatory rules ensures that the meaning assigned to the string will be the same in each case, so that from a purely semantic point of view the derivations are entirely equivalent. However, it has been claimed that the left-branching derivation is more psychologically plausible, since it appears to be more compatible with a model of processing where it is possible to gradually build an interpretation of the sentence as each new element is encountered. Steedman (1986) refers to this procedure as *incremental interpretation*. We return to this below.

Since it is possible to treat *I saw* as a constituent of category S/NP, it follows that relative clauses such as *who I saw* can be analysed without using empty categories, by treating the object relative pronoun *who(m)* as a functor taking an argument of category S/NP and returning a relative clause; we assume relative clauses to be backward modifiers of nouns, of category N\N, so that *who(m)* will be of category (N\N)/(S/NP). The derivation of *who I saw* will then proceed as follows:



Note especially that, given the above set of combinatory rules, there is no equivalent left-branching derivation. The lack of a left-branching derivation for such clauses, which is a particular feature of Steedman's grammar, is central to the account of processing load we shall describe below.[3]

Note also that it is possible to bring a special mechanism for extraction into a flexible categorial grammar, still without introducing empty categories into the syntax, as proposed by Moortgat (1988). Extracted arguments are distinguished by means of a special connective, and rules similar to the ones above govern the transmission of this connective.

2.2 Shift-Reduce Processing

It is argued in Ades and Steedman (1982) and in Steedman (1986) that, given a flexible theory of syntactic structure such as CCG, an incremental model of human sentence processing can be formulated in terms of a stack-based shift-reduce processor with a 'reduce-first' strategy. Such a processor will always find a left-branching derivation of a string in preference to a right-branching one if one exists, and will generally find a right-branching derivation if no left-branching derivation exists, but at an increased cost to the processor.

Suppose for simplicity that each lexical item has only one syntactic category, and that all combinatory rules are binary. We may regard the processor as maintaining a stack of categories, which is initially empty. Each time the processor encounters a new word, it shifts its category onto the stack, and then tries to reduce the top two categories on the stack by applying any one of the combinatory rules.[4] If a reduction is possible, the two old categories are replaced by the resultant category, and the processor attempts to reduce again. This continues until no more reductions are possible, and the processor moves on to the next word. If a single category remains on the stack at the end of the string, the parse has succeeded; in particular, if the category S remains, the string is a sentence.

This strategy is complicated by the presence of unary rules such as type-raising, and so here we shall make the simplifying assumption that no *syntactic* unary rules are used; instead, certain lexical items are listed as having two alternative categories. Thus proper names and personal pronouns are listed as having the two categories NP and $S/(S\backslash NP)$, and common nouns as having the two categories N and $N/(N\backslash N)$. In addition, we take determiners as having the two categories NP/N and $(S/(S\backslash NP))/N$, since they may specify either subject or non-subject noun phrases. It should be emphasized that these are not genuine categorial

---

[3] This feature is not shared by all flexible categorial grammars. For example, the 'predictive combinators' of Wittenburg (1987) and the completely flexible system of Lambek (1958) would both allow the following reduction:

$$(N\backslash N)/(S/NP) + NP \Rightarrow (N\backslash N)/((S\backslash NP)/NP)$$

Glyn Morrill (p.c.) has suggested that processing load for such grammars could be modelled in terms of the complexity of resultant categories, rather than the stack-based method described here.

[4] Given the above rule set, only one combinatory rule can apply to any two given categories, so there is no problem of nondeterminism in this case. However, this will not be true for an arbitrary grammar.

ambiguities; they simply represent alternative combinatorial possibilities. This means that we must introduce an element of nondeterminism into the processor by assuming that it always chooses the appropriate category for such lexical items.

We illustrate the above process by giving derivations for *I saw Mary* and *who I saw* in this fashion. The first column contains the next word in the string, and the second its category. This category is then shifted onto the stack, and the reductions listed in the third column (if any) are performed in that order. The fourth column shows the state of the stack, from bottom to top, after the reductions have been performed.

Word	Category	Red'ns	Stack
I	S/(S\NP)		S/(S\NP)
saw	(S\NP)/NP	>**B**	S/NP
Mary	NP	>	S

Word	Category	Red'ns	Stack	
who	(N\N)/(S/NP)		(N\N)/(S/NP)	
I	S/(S\NP)		(N\N)/(S/NP)	S/(S\NP)
saw	(S\NP)/NP	>**B** >	N\N	

In the first example, where a left-branching derivation is possible, a single reduction is performed at every step, and the stack depth never increases past one. But in the second, where no left-branching derivation exists, two categories accumulate on the stack, which requires two reductions at the last step.

2.3 Complexity of Processing

Given the above model, it is reasonable to hypothesize that there is some correlation between the number of categories that accumulate on the stack and the processing difficulty. For example, consider the multiple subject relative sentence (7) and the multiple object relative sentence (4). The two sentences would be processed as follows:

Word	Category	Red'ns	Stack
I	S/(S\NP)		S/(S\NP)
saw	(S\NP)/NP	>**B**	S/NP
the	NP/N	>**B**	S/N
farmer	N/(N\N)	>**B**	S/(N\N)
who	(N\N)/(S\NP)	>**B**	S/(S\NP)
owned	(S\NP)/NP	>**B**	S/NP
the	NP/N	>**B**	S/N
dog	N/(N\N)	>**B**	S/(N\N)
which	(N\N)/(S\NP)	>**B**	S/(S\NP)
chased	(S\NP)/NP	>**B**	S/NP
the	NP/N	>**B**	S/N
cat	N	>	S

Word	Category	Red'ns	Stack		
the	(S/(S\NP))/N		(S/(S\NP))/N		
cat	N/(N\N)	>**B**	(S/(S\NP))/(N\N)		
which	(N\N)/(S/NP)	>**B**	(S/(S\NP))/(S/NP)		
the	(S/(S\NP))/N		(S/(S\NP))/(S/NP)	(S/(S\NP))/N	
dog	N/(N\N)	>**B**	(S/(S\NP))/(S/NP)	(S/(S\NP))/(N\N)	
which	(N\N)/(S/NP)	>**B**	(S/(S\NP))/(S/NP)	(S/(S\NP))/(S/NP)	
the	(S/(S\NP))/N		(S/(S\NP))/(S/NP)	(S/(S\NP))/(S/NP)	(S/(S\NP))/N
farmer	N	>	(S/(S\NP))/(S/NP)	(S/(S\NP))/(S/NP)	S/(S\NP)
owned	(S\NP)/NP	>**B** >	(S/(S\NP))/(S/NP)	S/(S\NP)	
chased	(S\NP)/NP	>**B** >	S/(S\NP)		
fled	S\NP	>	S		

In the first example the stack depth remains constant with each new relative clause, but in the second the stack depth increases by one each time. This leads to the correct prediction that processing difficulty remains more or less constant as the number of multiple subject relative clauses is increased, but increases with the number of multiple object relative clauses.

Suppose we now apply this processing model to a multiple pied piping sentence such as (14) above. If we assume *in which* to be a single lexical item of category (N\N)/(S/PP),[5] we obtain the following result:

Word	Category	Red'ns	Stack	
John	S/(S\NP)		S/(S\NP)	
found	(S\NP)/NP	>**B**	S/NP	
the	NP/N	>**B**	S/N	
box	N/(N\N)	>**B**	S/(N\N)	
in which	(N\N)/(S/PP)	>**B**	S/(S/PP)	
I	S/(S\NP)		S/(S/PP)	S/(S\NP)
put	((S\NP)/PP)/NP	>$\mathbf{B}^2$ >**B**	S/NP	
the	NP/N	>**B**	S/N	
tray	N/(N\N)	>**B**	S/(N\N)	
on which	(N\N)/(S/PP)	>**B**	S/(S/PP)	
Mary	S/(S\NP)		S/(S/PP)	S/(S\NP)
placed	((S\NP)/PP)/NP	>$\mathbf{B}^2$ >**B**	S/NP	
the	NP/N	>**B**	S/N	
tin	N	>	S	

Although the stack depth rises locally to two on the subject of each relative clause, it falls to one each time the verb is reached, so that there is no global increase in stack depth with each relative clause. It follows that if further similar relative clauses were added (as in (17)) the overall processing load should not increase. In other words, in processing terms this sentence should behave similarly to the multiple subject relative sentence, which appears to be the case.

Contrast (5), the corresponding multiple preposition stranding construction:

---

[5] This category can be derived either by assigning the polymorphic category $((\mathrm{N}\backslash\mathrm{N})/(\mathrm{S}/X))\backslash(X/\mathrm{NP})$ to *which* (Szabolcsi 1987) or by adding a specific pied piping rule to the grammar (Morrill 1988).

Word	Category	Red'ns	Stack		
John	S/(S\NP)		S/(S\NP)		
found	(S\NP)/NP	>**B**	S/NP		
the	NP/N	>**B**	S/N		
box	N/(N\N)	>**B**	S/(N\N)		
which	(N\N)/(S/NP)	>**B**	S/(S/NP)		
I	S/(S\NP)		S/(S/NP)	S/(S\NP)	
put	((S\NP)/PP)/NP	$>\mathbf{B}^2$	S/(S/NP)	(S/PP)/NP	
the	NP/N	>**B**	S/(S/NP)	(S/PP)/N	
tray	N/(N\N)	>**B**	S/(S/NP)	(S/PP)/(N\N)	
which	(N\N)/(S/NP)	>**B**	S/(S/NP)	(S/PP)/(S/NP)	
Mary	S/(S\NP)		S/(S/NP)	(S/PP)/(S/NP)	S/(S\NP)
placed	((S\NP)/PP)/NP	$>\mathbf{B}^2$	S/(S/NP)	(S/PP)/(S/NP)	(S/PP)/NP
the	NP/N	>**B**	S/(S/NP)	(S/PP)/(S/NP)	(S/PP)/N
tin	N	>	S/(S/NP)	(S/PP)/(S/NP)	S/PP
on	PP/NP	>**B** >	S/(S/NP)	S/PP	
in	PP/NP	>**B** >	S		

Here the stack depth increases globally with each new relative clause, and hence if further similar relative clauses were added the sentence should become harder to process. In processing terms this sentence should pattern with the multiple object relative sentence, which again appears to be the case.

It is fairly clear why such a stack-based model should explain the contrast in processability between sentences such as (12) and (13): in (12) there is no accumulation of irreducible categories on the stack, because all dependencies are local, whereas in (13) it is impossible to reduce the main verb and the three NPs. But how does it capture filler-subcategorizer association in cases of extraction? In all such cases, the filler has a category that we might schematically represent as $X/(Y/Z)$, where Z would be the category of the extracted element in a filler-gap analysis. This means that a category of the form Y/Z (or a function into Y/Z) has to be built up before a reduction can take place, and this will happen exactly when the subcategorizer, of category W/Z (or a function into W/Z) for some W, is reached.

It should be noted that the above rule set does not cover all possible cases of extraction, and in particular extraction from non-peripheral positions (e.g. *the tray [which]*$_1$ *I put* $\emptyset_1$ *in the box*) is not covered. Steedman proposes a so-called *disharmonic* composition rule to cover such constructions, but this makes incorrect processing predictions when applied to a sentence such as (19):

(19) John found the tin which Mary placed on the tray which I put in the box.

The reason for this is that in Steedman's theory as it stands the PP *on the tray which I put in the box* must be constructed in full before it can be combined with the verb *placed.* It follows that if further relative clauses were added in the usual fashion the stack depth would increase, which clearly does not tie in with the processing data since the sentence can be extended indefinitely in the manner of (8) or (17) without becoming difficult to understand. However, note that the filler-subcategorizer dependencies in (19) are disjoint:

(20) John found the tin [which]$_1$ Mary [placed]$_1$ on the tray [which]$_2$ I [put]$_2$ in the box.

This leads one to suppose that a version of the theory where *John found the tin which* was reducible with *Mary placed* would make more appropriate processing predictions; this would be the case, for instance, if extractional connectives were introduced into CCG instead of disharmonic rules. Processing considerations appear therefore to be able to help discriminate between alternative versions of categorial grammar.

3 Alternative Analyses Using Gaps

We have argued that filler-gap association is impossible as a mechanism for processing unbounded dependencies, because it forces nested dependency patterns on constructions where this is clearly wrong. A possible counter-argument is that gaps do in fact have processing reality, but that we have made incorrect assumptions about their location. Is it necessary that they are mandatorily sited at the 'canonical' location, as we have assumed?

3.1 An Analysis Using Heavy NP Shift

Let us return to sentence (14), with the standard analysis (15) (repeated here as (21)):

(21) John found the box [in which]$_1$ I put the tray [on which]$_2$ Mary placed the tin $\emptyset_2$ $\emptyset_1$.

We can reanalyse this in filler-gap terms by arguing that *the tray [on which]$_2$ Mary placed the tin* $\emptyset_2$ is in fact a heavy NP shifted beyond the final gap $\emptyset_1$. This means that this gap is now adjacent to the verb *put* and hence we have a disjoint dependency pattern; the structure might be argued to be as in (22) below:

(22) John found the box [in which]$_1$ I put $\emptyset_1$ the tray [on which]$_2$ Mary placed the tin $\emptyset_2$.

In comprehension, it becomes impossible to assume that no lexical material has been moved past the gap. Therefore it will not always be possible given a string of words to uniquely determine where the gaps appear; in particular, we cannot assume that the gap is to be found between the words where the corresponding argument would occur in the canonical sentence. This obviously has far-reaching implications about how we can interpret psycholinguistic results. We now also need a theory of what the conditions are which make the parser postulate this analysis (a complication unnecessary in the filler-subcategorizer theory). Incrementally, we have no way of knowing that *the tray* will prove to be the beginning of a heavy constituent (we certainly do not have to stress *the tin*), so we have no reason to locate the gap before *the tray*. Unless we always assume heaviness (which is bizarre), we will be forced to backtrack in order to position the gap directly after *put*. This is very counterintuitive; there is certainly no introspective evidence for this kind of garden-path.

However, there are more conclusive arguments against this analysis. First, heavy-shifting itself is usually thought of as leaving a gap (movement analyses go back to Ross (1967); see also Postal (1974)). The psycholinguistic meaning of gaps due to rightward extraction is unclear, as we have indicated, but nonetheless we find ourselves with a non-disjoint pattern of filler-gap dependencies, which even with rightward movement surely cannot be without processing cost:

(23) John found the box [in which]$_1$ I put $\emptyset_3$ $\emptyset_1$ [the tray [on which]$_2$ Mary placed the tin $\emptyset_2$]$_3$.

When the sentence is extended the dependency pattern becomes even more contorted very quickly. In order to avoid this, we have to argue that heavy NP shift does not leave a psychologically real gap. This is a possible position, but seems unlikely.

More conclusively, we can construct multiple pied-piping sentences using verbs with a ditransitive subcategorization frame like *give*. These verbs do not have an optional heavy-shifted argument configuration; on the pragmatically sensible reading, (24) is out:

(24) *I gave the book the woman in the heavy winter coat.

Now, (25) does not produce any processing difficulty:

(25) John wrote the book which Mary gave the slave who Tom sold the nobleman.

Sentences like (25) are open to an irrelevant (here, pragmatically infelicitous) reading where the slave is given to the book, which is why we have so far been using verbs like *put* instead. But the argument is the same here, except that the heavy-shifted reading is now impossible. We cannot in this way avoid the 'canonical' gap-assignment shown below:

(26) John wrote the book [which]$_1$ Mary gave the slave [who]$_2$ Tom sold the nobleman $\emptyset_2$ $\emptyset_1$.

Now, this sentence type can be extended in exactly the same way as (14) (although the results are pragmatically rather strange!):

(27) John wrote the book which Mary gave the slave who Tom sold the senior slave who Bill sold the slavemaster who Sue sold the king.

The intended interpretation does not become hard to parse, and the intonation pattern is still that of a multiple subject relative sentence like (7). It is not possible to argue that (26) or (27) involves heavy NP shift. So a theory assuming that gaps are used in processing cannot avoid a nesting pattern of dependencies by this method.

3.2 An Analysis Using Extraposition from NP

It is conceivable that the processor uses another trick to avoid nested dependencies, where sentence (14) is in fact given a representation (28):

(28) John found the box [in which]$_1$ I put the tray $\emptyset_1$ [on which]$_2$ Mary placed the tin $\emptyset_2$.

What we have done here is extraposed the relative clause *[on which]$_1$ Mary placed the tin* $\emptyset_2$ past the gap $\emptyset_1$.

This account faces many of the same problems as the heavy NP shift account. The parser needs arbitrary strategies to decide when to extrapose material over the gap, and, unless it always behaves this way, it will be forced to backtrack; as mentioned above, there is no evidence of any garden-path effects. The effect on the interpretation of experimental results would again be considerable.

More importantly, if the extraposed element leaves a gap in its canonical location, as in the heavy NP shift example, we end up with as complicated a dependency pattern as the nested pattern we are trying to avoid:

(29) John found the box [in which]$_1$ I put the tray $\emptyset_3$ $\emptyset_1$ [[on which]$_2$ Mary placed the tin $\emptyset_2$]$_3$.

Although extraposition from NP does not leave a gap in an argument position, there must be some kind of semantic association. Consider sentence (30) below:

(30) How enthusiastically does John play the violin?

This sentence involves an extraction from an adjunct position, and so there is clearly a filler, *how enthusiastically*, which must be linked with the location after *violin*. This suggests that gaps are licensed by the presence of fillers rather than the absence of arguments, and hence that if heavy NP shift leaves a gap then so does extraposition from NP. As mentioned above, it can be argued that rightward movement does not produce gaps; however, even then we have to explain how the extraposed modifier is associated with the element it modifies, and so the pattern of associations will still be complicated.

The best defence against this argument, however, again involves ditransitives like (25). Note first that extraposition from NP over a second NP is always highly marked, and is especially bad with *that*-less relatives, e.g. (31):

(31) ?*Mary gave the slave the punishment Tom had always treated well.

Therefore, although we could give such an analysis to (25), it would be very odd to do the same with (32), where the relative pronouns have been removed:

(32) John wrote the book Mary gave the slave Tom sold the nobleman.

But since this sentence is not difficult to process, no extraposition can have occurred from between *the book* and *Mary*. And yet again the sentence type can be extended indefinitely. Thus, as with the heavy NP shift analysis, there are certain sentence types with canonically nested filler-gap dependencies where extraposition from NP would not be permitted. If we retain gaps in our theory of processing, we are forced to assume a nesting pattern of dependencies.

4 Discussion

4.1 Centre-Embedding and Nested Dependencies

When we reanalyse unbounded dependencies so that the filler associates with its subcategorizer rather than with a gap, we find that the multiple preposition stranding sentence remains nested but that the multiple pied-piping sentence becomes disjoint in its dependencies. Notice that the multiple preposition stranding sentence involves centre-embedding, but that the multiple pied-piping sentence does not. We shall argue that nesting dependency patterns are equivalent to centre-embeddings, and thus point to an explanation for the difficulty of centre-embedded constructions: all the fillers have to be retained separately in memory, and it is difficult to construct the dependency relations between them and their subcategorizers.

Multiple pied-piping sentences involve nested filler-gap dependencies but disjoint filler-subcategorizer dependencies, and are easy to process. In contrast, the following German multiple subject relative sentence has disjoint filler-gap dependencies (33a) but nested filler-subcategorizer dependencies (33b):

(33) a. Der Bauer [der]$_1$ $\emptyset_1$ das Mädchen [das]$_2$ $\emptyset_2$ den Jungen küßte schlug ging.
b. Der Bauer [der]$_1$ das Mädchen [das]$_2$ den Jungen [küßte]$_2$ [schlug]$_1$ ging.
The farmer$_{NOM}$ who$_{NOM}$ the girl$_{ACC}$ who$_{NOM}$ the boy$_{ACC}$ kissed hit went.
'The farmer who hit the girl who kissed the boy went'

This is another centre-embedded sentence, and native speakers find it difficult to process. Once again nested filler-subcategorizer dependencies correlate with centre-embedded constructions. The location of the gaps, and the pattern of filler-gap dependencies, is again shown to be irrelevant to processing complexity.

However, we can go further than this, because it is possible to show that centre-embedded constructions that involve no extractions at all are hard to process. Again, however, when we analyse them in dependency terms we find a nested pattern (though now it is unnecessary to make a distinction between filler-subcategorizer and other non-local dependency relationships). We have already given a German example of this, (13), reproduced here as (35), which is hard to process:[6]

(35) Johanna hat den Männern Hans die Pferde füttern lehren geholfen.
Johanna$_{NOM}$ has the men$_{DAT}$ Hans$_{ACC}$ the horses$_{ACC}$ to-feed to-teach helped.
'Johanna has helped the men to teach Hans to feed the horses'

In general, we can argue that the difficulty of processing centre-embedded sentences cannot be explained by the pattern of filler-gap dependencies. On the other hand, an account in terms of filler-subcategorizer (and other non-local) dependencies is adequate. Hence there is no processing argument for the existence of gaps.

In CCG terms, dependency patterns are determined by the reduction of functors and arguments, and centre-embeddings correspond to the accumulation of irreducible categories on the stack. It follows that by this definition nested dependencies and centre-embeddings must coincide.

4.2 Psycholinguistic Implications

In recent years there have been a number of psycholinguistic results which appear to support the notion that a gap is positioned at the extraction site and the filler is associated with the gap. For example, gap-location and filling procedures presumably involve a processing load. Hence processing difficulty at gap sites has been taken as evidence for their psychological reality. Crain and Fodor (1985) and Stowe (1986) have given demonstrations of this processing difficulty on the basis of what has been called the 'filled-gap effect'. Here the processor has encountered a filler, and is looking for a gap. Stowe uses (36) and (37) below:

(36) My brother wanted to know if Ruth will bring us home to Mom at Christmas.

(37) My brother wanted to know who Ruth will bring us home to at Christmas.

---

[6] Note that Government-Binding theory (Chomsky 1981) postulates *PRO* subjects to the two infinitives:

(34) Johanna hat [den Männern]$_1$ PRO$_1$ [Hans]$_2$ PRO$_2$ die Pferde füttern lehren geholfen.

However, the *PRO*-antecedent dependency pattern is disjoint, and so cannot serve as an explanation for the processing difficulty of the sentence.

Using a word-by-word self-paced reading time technique, she showed that subjects take longer to process *us* in (37) than (36). The gap, she argues, is sited by the processor after *bring* in (37). Hence, when the word *us* is encountered, we realise that the gap cannot be located directly after *bring*, and so we are forced to backtrack. This therefore shows that at least in this case the parser locates a gap in an incrementally possible position.[7]

There are also studies which show that at least some properties of the filler are reactivated at the gap location. Thus, Swinney, Ford, Frauenfelder and Bresnan (1988) (cited in Nicol (in press)) have used a cross-modal lexical decision task to show that the meaning of the antecedent phrase is reactivated at the gap location, with sentences like (38):

(38) The policeman saw the boy $[\text{that}]_1$ the crowd at the party accused $\emptyset_1$ of the crime.

They found facilitation for associates of the noun phrase associated with the correct antecedent of the gap (but not, for instance, of the other noun phrases), *the boy*, at the gap location but not just prior to the gap location. This indicates that the gap causes a reactivation of the properties of the filler; there is not merely some residual activation. This can be taken as another demonstration of the psychological reality of filler-gap dependencies, conceived of as a kind of anaphoric link.

However, it is possible to interpret these results without having to postulate any empty categories at all. In these experiments, the supposed gap location is adjacent to the subcategorizer. For instance, in Swinney et al's experiment, the priming effect could be due to the establishment of the dependency between the filler and the verb *accused*. When the dependency is forged, the already-processed half of the dependency becomes salient once again, and so not surprisingly its properties are reactivated. Establishing of a dependency, and realising that one has been wrongly established, will involve a processing load; this can be seen as an explanation of the 'filled-gap effect'.[8] Notice how this fits neatly into the categorial grammar model we have proposed above: the dependency is established by the reduction of the two categories on the top of the stack together. Reduction involves a processing load (as does backtracking from a reduction we later realise to have been misguided). It will also cause relevant components of the lower of the two categories to be reactivated.

We would therefore predict the above results to the same extent that a theory using gaps would. However, we would also predict processing complexity and, more important, reactivation in cases where there is no gap but where a dependency can be established. Hence in a multiple pied-piping sentence like (14), repeated here as (39), we would predict reactivation of a filler like *box* not at the end of the sentence but rather at the verb *put*:

(39) John found the box in which I put the tray on which Mary placed the tin.

This would not be a feasible experiment because the filler is too close to the verb, but the sentence could easily be modified, for instance by giving *put* a long subject rather than *I*. We would also expect reactivation of the subject in any construction with SOV word order, such as a subordinate clause in German, at the verb. In this case it would be very difficult to claim

---

[7] The experiment of course tells us more than this; it shows us that the parser is very quick at using available information, and that it tries to discharge fillers quickly.

[8] Note that this effect is only explicable in a gap-based account if the gap is postulated immediately at the subcategorizer; the next word does not have to be waited for. This is however directly predicted by the filler-subcategorizer account. Tanenhaus, Garnsey and Boland (forthcoming) present evidence that this is not merely an artefact of the self-paced reading time methodology.

that this reactivation was due to the presence of any kind of empty category. Likewise, in (35), we would predict that *den Männern* would get reactivated at *geholfen*, of which it is the object (or perhaps at *lehren*, of which it is the controller). This sentence contains no extractions, and Government-Binding theory postulates an associated *PRO* immediately after *den Männern*, so again there can be no explanation for reactivation in terms of empty categories. We would thus have further evidence that reactivation is in fact due to the establishment of dependencies with elements that have temporarily become inactive.

4.3 Gap-Filling and NP-Trace in Passive Constructions

It has been argued by Bever and McElree (1988) that there is experimental evidence for the psychological reality of NP-trace and antecedent-empty category association in passive sentences like (40):

(40) [The astute lawyer]$_1$ who faced the female judge was suspected e_1 constantly.

The e_1 indicates the location of the NP-trace in GB theory. Bever and McElree show that subjects acknowledge that the probe word *astute* is present in (40) quicker than in a similar sentence without a NP-trace. They argue that this shows that the antecedent of the NP-trace, *the astute lawyer*, is reactivated at the trace position. This presumably means that the antecedent must be held in memory until it can be associated with its trace (though it may well not be obvious until the passive verb that it is actually an antecedent for a trace).

Presumably, having to hold a large number of NP-trace antecedents in memory together will stretch memory resources; all of them will have to be reactivated at some point. We can, however, construct a nested pattern of antecedent-NP-trace dependency pairs in much the same way as we did with Wh filler-gap sentences above. Again, we appeal to verbs which can be ditransitive, such as *give*. In a well-attested dialect of English, there are two distinct passives, so that not only (41) but also (42) is acceptable under the pragmatically felicitous reading:

(41) John was given a book.

(42) A book was given John.

The GB representation of (42) has the trace after *John*, because *give John a book* is acceptable, but not **give a book John.* Hence the filler, *a book*, has to be held in memory until after *John.* As with the pied-piping sentences, let us extend the example in the following way:

(43) A bone was given a dog which was given a slave.

The representation for this sentence is as follows within GB theory (leaving out Wh-traces):

(44) [A bone]$_1$ was given [a dog]$_2$ which was given a slave e_2 e_1.

As before, we find that we have a nested dependency pattern, and therefore a stack of fillers to remember, until we reach the end of the sentence when we discharge them all. This is clearly not in keeping with the lack of processing difficulty for this sentence, and again its intonational pattern is that of a sentence with uncontroversially disjoint dependencies like (7). The sentence type can be indefinitely extended, so that the NP-trace theory would predict multiple nested dependencies. This would suggest that there is no psychological reality to NP-trace in passive constructions; instead the subject is associated directly with the passive verb by normal dependency relations. Bever and McElree's results therefore cannot be interpreted in the way they suggest.

4.4 Gap-Filling and Word-Order Freedom

The notion that gaps are located between particular words in a string assumes that there is a canonical form that the string is a deviation from. In a language with a considerable degree of word-order rigidity such as English this may be plausible, but in other languages the notion of canonicality may not be meaningful. Hence in cases of extraction, how can the parser decide where the gap is? In a totally free word order language, extraction of course cannot occur, but the interesting case would be a language with clear constituency boundaries, and clear cases of extraction from constituents, but freedom within the constituent. For instance, if a language permits both OV and VO word order and extraction, it is unclear whether the correct structure would be Wh...[V Gap] or Wh...[Gap V]. We may have some set of pragmatic heuristics to make the decision as to where to locate the gap, or we may always have a canonical order for a particular language or particular construction. There may be an interaction with some principle like locating the gap as soon as possible. But any such method would be ad hoc and require additional grammatical mechanisms for dealing with one construction type. The alternative is that there is no reconstruction of a canonical form at all in these cases, but rather that the 'extracted' element is simply associated with its subcategorizer directly. Not having gaps in the grammar thus allows a considerable simplification of the parsing strategies we use.

5 Conclusion

Assuming that recursive nesting of dependencies must at some point become hard to process, we have argued that certain sentences that appear to contain nested dependencies cannot in fact do so. We have proposed that this is because there is no such thing as a filler-gap dependency, and that instead the filler is associated with its subcategorizer directly. This seems to simplify the linguistic apparatus needed, and also to explain the data correctly. This view of dependencies fits in with categorial grammar approaches to processing. We have interpreted psycholinguistic results on the processing of long-distance dependencies as indicating the establishment of dependency relations between linguistically realised elements rather than supporting the postulation of empty categories and the association of antecedents with them.

Acknowledgements

We would like to thank Kathrin Cooper, Robin Cooper, Elisabet Engdahl, Mark Hepple, Pauline Jacobson, Glyn Morrill, Antonio Sanfilippo, and Mark Steedman for their comments and suggestions. We acknowledge the support of ESRC studentship C00428722002 and SERC studentship 88306971.

References

Ades, A. and Steedman, M. J. [1982] On the Order of Words. *Linguistics and Philosophy*, **4**, 517–558.

Bach, E., Brown, C. and Marslen-Wilson, W. [1986] Crossed and Nested Dependencies in German and Dutch: a Psycholinguistic Study. *Language and Cognitive Processes*, **1**, 249–262.

Bever, T. G. and McElree, B. [1988] Empty Categories Access their Antecedents during Comprehension. *Linguistic Inquiry*, **19**.

Chomsky, N. [1981] *Lectures on Government and Binding.* Dordrecht: Foris Publications.

Crain, S. and Fodor, J. D. [1985] How can Grammars Help Parsers? In Dowty, D. R., Karttunen, L. and Zwicky, A. M. (eds.) *Natural Language Parsing: Psychological, Computational and Theoretical Perspectives*, 94–128. Cambridge: Cambridge University Press.

Curry, H. B. and Feys, R. [1958] *Combinatory Logic*, Volume I. Amsterdam: North Holland.

Fodor, J. D. [1978] Parsing Strategies and Constraints on Transformations. *Linguistic Inquiry*, **9**, 427–473.

Lambek, J. [1958] The Mathematics of Sentence Structure. *American Mathematical Monthly*, **65**, 154–170.

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

Morrill, G. [1988] Extraction and Coordination in Phrase Structure Grammar and Categorial Grammar. PhD Thesis, Centre for Cognitive Science, University of Edinburgh.

Nicol, J. [in press] What the Parser Knows about the Grammar: Psycholinguistic Evidence. To appear in *Proceedings of WCCFL VIII.*

Postal, P. M. [1974] *On Raising.* Cambridge, Mass.: MIT Press.

Ross, J. R. [1967] Constraints on Variables in Syntax. PhD Thesis, MIT. Indiana University Linguistics Club.

Swinney, D., Ford, M., Frauenfelder, U. and Bresnan, J. [1988] On the Temporal Course of Gap-Filling and Antecedent Assignment during Sentence Comprehension. In Grosz, B., Kaplan, R., Macken, M. and Sag, I. (eds.) *Language Structure and Processing.* CSLI, Stanford, CA.

Steedman, M. J. [1986] Incremental Interpretation in Dialogue. ACORD Deliverable 2.4, Centre for Cognitive Science, Edinburgh.

Steedman, M. [1987a] Combinatory Grammars and Parasitic Gaps. *Natural Language and Linguistic Theory*, **5**, 403–439.

Steedman, M. [1987b] Combinators and Grammars. In Oehrle, R., Bach, E. and Wheeler, D. (eds.) *Categorial Grammars and Natural Language Structures.* Dordrecht: D. Reidel.

Stowe, L. A. [1986] Parsing WH-constructions: Evidence for On-Line Gap Location. *Language and Cognitive Processes*, **1**, 227–245.

Szabolcsi, A. [1987] Bound Variables in Syntax. In *The Sixth Amsterdam Colloquium*, Institute for Language, Logic and Information, University of Amsterdam, 1987.

Tanenhaus, M. K., Garnsey, S. M. and Boland, J. [forthcoming] Combinatory Lexical Information and Language Comprehension 1. To appear in Altmann, G. (ed.) *Cognitive Models of Speech Processing: Psycholinguistic and Computational Perspectives.* Cambridge, Mass.: MIT Press.

Wanner, E. and Maratsos, M. [1978] An ATN Approach to Comprehension. In Halle, M., Bresnan, J. and Miller, G. A. (eds.) *Linguistic Theory and Psychological Reality.* Cambridge, Mass.: MIT Press.

Wittenburg, K. [1987] Predictive Combinators: a Method for Efficient Processing of Combinatory Categorial Grammar. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, Stanford University, Stanford, Ca., 6–9 July, 1987.

Dependency and Constituency in Categorial Grammar

Guy Barry and Martin Pickering

March 26, 1991

1 Introduction

It has been seen as an advantage of some systems of categorial grammar (Steedman 1987; Moortgat 1988) that they allow the definition of some notion of 'flexible constituency'. This has been argued to allow accounts of syntactic phenomena such as coordination and extraction, as well as other phenomena such as incremental interpretation and intonational structure.

In phrase-structure theories of grammar, category symbols are merely labels for constituents, and so by definition everything that can be given a category must be a constituent. In categorial grammars, on the other hand, the symbols used reflect more general combining properties of strings. To avoid confusion we shall use the word *type* rather than *category* to refer to these symbols. Nevertheless, there is still an implicit assumption that any string that can be given a type is a constituent. We shall refer to this assumption as the *type-constituent correspondence* hypothesis.

In the Lambek calculus, every string can be given a type, so type-constituent correspondence would predict that every string is a constituent. If the notion of constituency has any independent linguistic use, then we cannot maintain type-constituent correspondence within the Lambek calculus. This indicates that we must either abandon the Lambek calculus as a linguistic framework or abandon type-constituent correspondence.

The former approach suggests the use of a more restricted calculus such as Steedman's Combinatory Categorial Grammar (CCG). By means of a particular set of type-combining rules, Steedman attempts to assign types to all and only those strings that can be regarded as constituents. This approach is open to two criticisms. Firstly, the type-combining rules are proposed on the basis of specific linguistic phenomena rather than general principles, so that the set of constituents generated is in some sense arbitrary, and (as we shall argue) leads to some anomalies in linguistic description. This could perhaps however be rectified by a more principled choice of rules. Secondly, though, Steedman's approach has no way of characterizing strings that are not constituents, even though it appears (as we shall also argue) that the characterization of such strings is necessary for the description of certain linguistic phenomena. We shall go into more detail on both these points in section 5.

In this paper we shall investigate the alternative approach of retaining the general Lambek framework, but dropping type-constituent correspondence. Instead we shall argue from the standpoint of dependency, by proposing a definition of dependency within the Lambek calculus, and from it deriving a new notion of constituency, which we shall refer to as *dependency constituency*. We shall argue that the notion of a dependency constituent has applications in the syntactic description of coordinate constructions, as well as in the development of an incremental account of sentence processing. We believe that it has applications in other areas of linguistics such as the syntax of extraction, and intonational structure.

2 Dependencies and Dependency Constituents

2.1 Heads and Dependents

All notions of dependency rely on a primitive notion of *head*; roughly speaking, the head is the element on which other elements depend. For example, consider the following sentence:

(1) Bill talks to Mary.

Three approaches seem plausible. If we assume that heads and dependents are both phrases, then given standard phrase-structure assumptions *talks to Mary* is the head of the sentence and has the single dependent *Bill*, and *talks* is the head of *talks to Mary* and has the single dependent *to Mary*. Alternatively, if we assume that heads are words and dependents are phrases, then *talks* can be taken as the head of the sentence and as having the two dependents *to Mary* and *Bill*. Finally, if we assume that heads and dependents are both words, then *talks* can be taken as the head of the sentence and as having the two dependents *to* and *Bill*, and *to* can be taken as having the single dependent *Mary*.

There is no primitive notion of head in categorial grammar, but the notion of semantic/syntactic *functor* is often taken to be head-like. This raises two questions; should categorial grammar be thought of as having phrasal or lexical heads and dependents, and what is the relationship of functors to 'traditional' (linguistically motivated) heads?

To answer the first point, let us limit our attention for the moment to applicative categorial grammar (**AB**). **AB** can be thought of as having phrasal heads and phrasal dependents if we adopt the following definition (where the semantic type $Y \rightarrow X$ collapses the two syntactic types X/Y, $Y\backslash X$):

Definition 1 *When a string of a functor type $Y \rightarrow X$ combines with a string of an argument type Y to form a string of type X, the functor string is the* phrasal head *and the argument string the* (phrasal) dependent.

Thus in *Bill talks to Mary* of type S, *talks to Mary* of type NP\S is the phrasal head, and *Bill* of type NP is the dependent. Similarly, in *talks to Mary* of type NP\S, *talks* of type (NP\S)/PP is the phrasal head, and *to Mary* of type PP is the dependent.

But we may also regard **AB** as having lexical heads and phrasal dependents if we adopt the following alternative definition:

Definition 2 *When a word of a functor type $Y_1 \rightarrow (\ldots \rightarrow (Y_n \rightarrow X)\ldots)$ combines with strings of argument types $Y_1, \ldots, Y_n$ to form a string of type X, the functor word is the* lexical head *and the argument strings the* (phrasal) dependents.

Thus, in *Bill talks to Mary* of type S, *talks* of type (NP\S)/PP is the lexical head, and *to Mary* of type PP and *Bill* of type NP are the dependents.

In other words, we can derive the notion of a lexical head from that of a phrasal head by a process similar to 'uncurrying' of functions. But it seems more natural to view dependents as phrasal, since functions take expressions of phrasal rather than lexical types as arguments. (This view will be reinforced when we consider the full Lambek calculus below.)

The second point begs the question of what is the traditionally accepted notion of head, about which there is widespread disagreement. We shall not go into the issues here; see Zwicky (1985) and Hudson's (1987) reply to Zwicky for a comprehensive discussion. Hudson gives a great deal of evidence which suggests that, for constructions involving complementation, most syntactically motivated definitions of head in fact coincide with the notion of semantic functor. However, the reverse appears to be true in modifier constructions, where according to Hudson's criteria for headship (and nearly all linguistic theories) the modified element appears to be the head, whereas on semantic grounds the modifier is standardly assumed to be the functor (as witnessed by the most common categorial type assignments N/N, N\N, (NP\S)\(NP\S) for adjectives, relative clauses and adverbs respectively).

There are three possible solutions to this problem, none of them entirely satisfactory. We could deny the equivalence of head and functor, but this would require us to define a separate theoretical primitive of 'head' independently of the categorial grammar formalism; we shall not pursue this further. Alternatively we could claim that modifiers are heads, although as we shall see in subsection 2.3 this approach leads to an unsatisfactory notion of constituency. Finally we could claim that modified elements are functors, which seems to be supported by the fact that the distinction between modifiers and optional arguments is not always clear.

We shall formalize the above description of modifiers by assuming that modifiers have atomic types, and that modifiable types are functions over zero or more premodifiers and zero or more postmodifiers (cf. HPSG (Pollard and Sag 1987)). We can achieve this by extending the categorial machinery with a 'Kleene star' structural operator, so that X^* means 'zero or more occurrences of X'.[1] We shall classify noun premodifiers as AdnPre, noun postmodifiers as AdnPost, verb premodifiers as AdvPre and verb postmodifiers as AdvPost. Thus lexical nouns will be categorized as

(AdnPre*\N)/AdnPost*,

from which the types N, AdnPre\N, AdnPre\(AdnPre\N), N/AdnPost etc. are derivable. Similarly intransitive verbs will be categorized as

(AdvPre*\(NP\S))/AdvPost*,

transitive verbs as

((AdvPre*\(NP\S))/AdvPost*)/NP,

and so on. Since it would be unwieldy (and uninformative) to write down these complex types in every derivation, we shall usually write only the derived type that is appropriate to the particular derivation. For example, *man* will be given the type N in *the man walks*, AdnPre\N in *the old man walks*, N/AdnPost in *the man who I like walks* and so on.

2.2 Dependency in the Lambek Calculus

When we try to generalize the ideas of head and dependent developed above for **AB** to the full Lambek calculus **L**, we run into a problem, since the roles of functor and argument can be reversed. Suppose we assume that head-dependent relations are based on the functor-argument relations implicit in lexical type assignments. Then, for instance, *John walks* consists of a function *walks* of type NP\S applying to an argument *John* of type NP, and so *walks* is the head. But if we assign the higher type S/(NP\S) to *John*, then *John* appears to be the head. If the basic meaning of *John* is represented as **john** in the first case, then its meaning in the second case will be $\lambda f.f$ **john**, where f is a variable of type $\mathrm{NP} \rightarrow \mathrm{S}$; when *John* is type-raised, the dependency relations implicit in the lexical assignments are lost. We shall refer to a meaning representation as *dependency-preserving* if it maintains the head-dependent relations derivable from lexical assignments. More formally:

---

[1] In terms of the structural operators proposed in Morrill, Leslie, Hepple and Barry (1990), X^* is equivalent to $X^{+?}$.

Definition 3 *A lambda-calculus meaning representation is* dependency-preserving *iff it does not involve abstraction of a variable that occurs as a functor within it.*

As pointed out in Morrill, Leslie, Hepple and Barry (1990), there is a direct correspondence between such lambda-terms and derivations in the 'natural deduction' style formulation of **L**. Thus definition 3 is equivalent to the following:

Definition 4 *A derivation in* **L** *is* dependency-preserving *iff it does not discharge an assumption that forms the major premise of an Elimination inference.*

So for example consider the six derivations below, and their associated lambda-terms. The three in (2) are dependency-preserving, and the three in (3) are not:[2]

(2) a.

$$\frac{\dfrac{\text{the}}{NP/N} \qquad \dfrac{\text{dog}}{N}}{NP}{\scriptstyle /E}$$

the dog

b.

$$\frac{\dfrac{\dfrac{\text{John}}{NP} \qquad \dfrac{\dfrac{\text{likes}}{(NP\backslash S)/NP} \qquad [NP]_1}{NP\backslash S}{\scriptstyle /E}}{S}{\scriptstyle \backslash E}}{S/NP}{\scriptstyle /I_1}$$

$\lambda x^{NP}[\mathbf{likes}\ x\ \mathbf{john}]$

c.

$$\frac{\dfrac{\dfrac{\text{will}}{(NP\backslash S)/VP} \qquad \dfrac{\dfrac{\text{see}}{VP/NP} \qquad [NP]_1}{VP}{\scriptstyle /E}}{NP\backslash S}{\scriptstyle /E}}{(NP\backslash S)/NP}{\scriptstyle /I_1}$$

$\lambda x^{NP}[\mathbf{will}\ (\mathbf{see}\ x)]$

(3) a.

$$\frac{\dfrac{\dfrac{[NP/N]_1 \qquad \dfrac{\text{dog}}{N}}{NP}{\scriptstyle /E} \qquad \dfrac{\text{runs}}{NP\backslash S}}{S}{\scriptstyle \backslash E}}{(NP/N)\backslash S}{\scriptstyle \backslash I_1}$$

$\lambda f^{N\rightarrow NP}[\mathbf{runs}\ (f\ \mathbf{dog})]$

b.

$$\frac{\dfrac{\dfrac{\text{that}}{SP/S} \qquad \dfrac{\dfrac{\text{Harry}}{NP} \qquad [NP\backslash S]_1}{S}{\scriptstyle \backslash E}}{SP}{\scriptstyle /E}}{SP/(NP\backslash S)}{\scriptstyle /I_1}$$

$\lambda f^{NP\rightarrow S}[\mathbf{that}\ (f\ \mathbf{harry})]$

c.

$$\frac{\dfrac{\dfrac{[((NP\backslash S)/NP)/NP]_1 \qquad \dfrac{\text{Mary}}{NP}}{(NP\backslash S)/NP}{\scriptstyle /E} \qquad \dfrac{\text{John}}{NP}}{NP\backslash S}{\scriptstyle /E}}{(((NP\backslash S)/NP)/NP)\backslash(NP\backslash S)}{\scriptstyle \backslash I_1}$$

$\lambda f^{NP\rightarrow(NP\rightarrow(NP\rightarrow S))}[f\ \mathbf{mary}\ \mathbf{john}]$

The term **the dog** is dependency-preserving because it does not involve abstraction, and the terms λx[**likes** x **john**] and λx[**will** (**see** x)] are dependency-preserving because in each

---

[2] We use a convention of left-associativity for function application, e.g. **likes mary john** means (**likes mary**) **john**.

case the abstracted variable does not occur as a functor. On the other hand, the terms $\lambda f[\mathbf{runs}\ (f\ \mathbf{dog})]$, $\lambda f[\mathbf{that}\ (f\ \mathbf{harry})]$ and $\lambda f[f\ \mathbf{mary}\ \mathbf{john}]$ are not dependency-preserving, since in each case the abstracted variable is a functor over one or more arguments.

We shall refer to strings with dependency-preserving analyses as *dependency constituents.* More precisely:

Definition 5 *A* dependency constituent *is a string (under a particular reading) whose normal-form derivation in* **L** *is dependency-preserving (equivalently, for which* some *derivation in* **L** *is dependency-preserving).*

(The equivalence of the two definitions is immediate because the normalization process never introduces new functors.) Thus the underlined substrings in (4) below are analysable as dependency constituents, whereas those in (5) are not:

(4) a. The dog runs.
b. John likes Mary.
c. John will see Mary.

(5) a. The dog runs.
b. I think that Harry left.
c. I showed Mary John.

Intuitively, we may think of the underlined substrings in (5) as having 'missing heads'; two words cannot be related because the head of one or both of them is absent from the string.

If we take the traditional notion of constituent to correspond to strings with purely applicative derivations, then the notion of dependency constituent is seen to subsume the traditional notion. The advantage of this notion of constituency is that it allows a degree of flexibility in constituent structure while still requiring that constituents are (in some sense) semantically coherent units. For example, consider:

(6) John thinks that Harry likes Mary.

Here, the substrings *that Harry*, *thinks that Harry* and *John thinks that Harry* are not dependency constituents, but every other substring (including one-word strings and the entire sentence) is, e.g. *John thinks*, *John thinks that*, *that Harry likes*, *thinks that Harry likes.*

2.3 Consequences of Dependency Constituency

Clearly the choice of head in modifier constructions will make a difference to what strings are regarded as dependency constituents. Consider for example:

(7) the tall man

(8) the man who I saw

If modifiers are regarded as heads, the underlined substring in (7) will be a dependency constituent, but the one in (8) will not be. On the other hand, if modified elements are regarded as heads, the underlined substring in (7) will not be a dependency constituent, but the one in (8) will be. The relevant proofs and lambda-terms follow:

(9) a.
$$\dfrac{\dfrac{\dfrac{\text{the}}{\mathrm{NP/N}}\quad\dfrac{\dfrac{\text{tall}}{\mathrm{N/N}}\quad[\mathrm{N}]_1}{\mathrm{N}}/\mathrm{E}}{\mathrm{NP}}/\mathrm{E}}{\mathrm{NP/N}}/\mathrm{I}_1$$
$$\lambda x^{\mathrm{N}}[\mathbf{the}\ (\mathbf{tall}\ x)]$$

b.
$$\dfrac{\dfrac{\dfrac{\text{the}}{\mathrm{NP/N}}\quad\dfrac{\dfrac{\text{tall}}{\mathrm{AdnPre}}\quad[\mathrm{AdnPre}\backslash\mathrm{N}]_1}{\mathrm{N}}\backslash\mathrm{E}}{\mathrm{NP}}/\mathrm{E}}{\mathrm{NP/(AdnPre}\backslash\mathrm{N)}}/\mathrm{I}_1$$
$$\lambda f^{\mathrm{AdnPre}\to\mathrm{N}}[\mathbf{the}\ (f\ \mathbf{tall})]$$

(10) a.
$$\dfrac{\dfrac{\dfrac{\text{the}}{\mathrm{NP/N}}\quad\dfrac{\dfrac{\text{man}}{\mathrm{N}}\quad[\mathrm{N}\backslash\mathrm{N}]_1}{\mathrm{N}}\backslash\mathrm{E}}{\mathrm{NP}}/\mathrm{E}}{\mathrm{NP/(N}\backslash\mathrm{N)}}/\mathrm{I}_1$$
$$\lambda f^{\mathrm{N}\to\mathrm{N}}[\mathbf{the}\ (f\ \mathbf{man})]$$

b.
$$\dfrac{\dfrac{\dfrac{\text{the}}{\mathrm{NP/N}}\quad\dfrac{\dfrac{\text{man}}{\mathrm{N/AdnPost}}\quad[\mathrm{AdnPost}]_1}{\mathrm{N}}/\mathrm{E}}{\mathrm{NP}}/\mathrm{E}}{\mathrm{NP/AdnPost}}/\mathrm{I}_1$$
$$\lambda x^{\mathrm{AdnPost}}[\mathbf{the}\ (\mathbf{man}\ x)]$$

This would seem to support the choice of modified element as head, since there is a clear intuitive connection between *the* and *man*, but not between *the* and *tall*. We shall assume modified elements to be heads for the rest of the paper (though we shall sometimes give alternative analyses where modifiers are assumed to be heads).

It is worth noting here that the string underlined in (11) cannot be a dependency constituent, whatever the choice of head in modifier constructions:

(11) John loves <u>Mary madly</u>.

(12) a.
$$\dfrac{\dfrac{\dfrac{[\mathrm{(NP}\backslash\mathrm{S)/NP}]_1\quad\dfrac{\text{Mary}}{\mathrm{NP}}}{\mathrm{NP}\backslash\mathrm{S}}/\mathrm{E}\quad\dfrac{\text{madly}}{\mathrm{(NP}\backslash\mathrm{S)}\backslash\mathrm{(NP}\backslash\mathrm{S)}}}{\mathrm{NP}\backslash\mathrm{S}}\backslash\mathrm{E}}{\mathrm{((NP}\backslash\mathrm{S)/NP)}\backslash\mathrm{(NP}\backslash\mathrm{S)}}\backslash\mathrm{I}_1$$
$$\lambda f^{\mathrm{NP}\to(\mathrm{NP}\to\mathrm{S})}[\mathbf{madly}\ (f\ \mathbf{mary})]$$

b.
$$\dfrac{\dfrac{\dfrac{[\mathrm{((NP}\backslash\mathrm{S)/AdvPost)/NP}]_1\quad\dfrac{\text{Mary}}{\mathrm{NP}}}{\mathrm{(NP}\backslash\mathrm{S)/AdvPost}}/\mathrm{E}\quad\dfrac{\text{madly}}{\mathrm{AdvPost}}}{\mathrm{NP}\backslash\mathrm{S}}/\mathrm{E}}{\mathrm{(((NP}\backslash\mathrm{S)/AdvPost)/NP)}\backslash\mathrm{(NP}\backslash\mathrm{S)}}\backslash\mathrm{I}_1$$
$$\lambda f^{\mathrm{NP}\to(\mathrm{AdvPost}\to(\mathrm{NP}\to\mathrm{S}))}[(f\ \mathbf{mary})\ \mathbf{madly}]$$

Some interesting consequences of this definition of dependency emerge when we consider constructions involving extraction. Firstly, as one might expect, any relative clause is a dependency constituent, e.g. the underlined fragments of the following two constructions:[3]

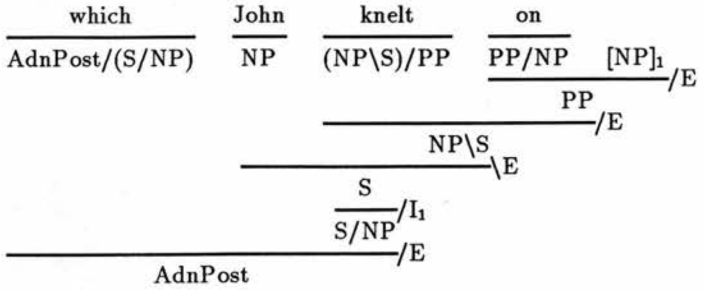
(13) the box <u>which John knelt on</u>

(14) the box <u>on which John knelt</u>

---

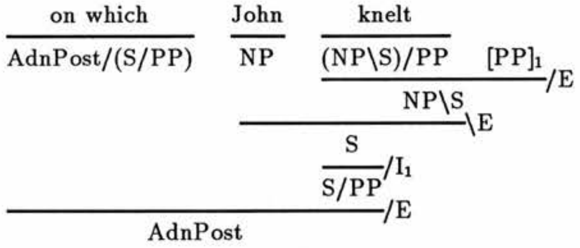
[3] We treat phrases such as *on which* and *in which* as single lexical items for the purposes of this discussion.

(15)



which (λx^{NP}[**knelt** (**on** x) **john**])

(16)

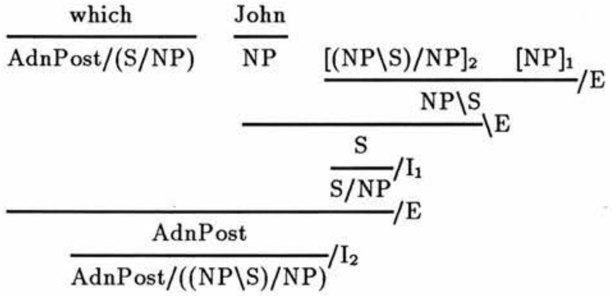


on-which (λx^{PP}[**knelt** x **john**])

On the other hand, some fragments of relative clauses are not dependency constituents, such as the two underlined below:

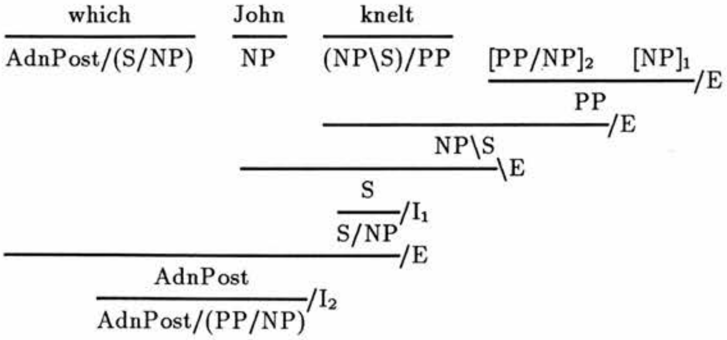
(17) the box <u>which John</u> knelt on
(18) the box <u>which John knelt</u> on

(19)



$\lambda f^{\mathrm{NP}\to(\mathrm{NP}\to\mathrm{S})}$[**which** ($\lambda x^{\mathrm{NP}}$[$f$ x **john**])]

(20)



$\lambda g^{\mathrm{NP}\to\mathrm{PP}}$[**which** ($\lambda x^{\mathrm{NP}}$[**knelt** ($g$ x) **john**])]

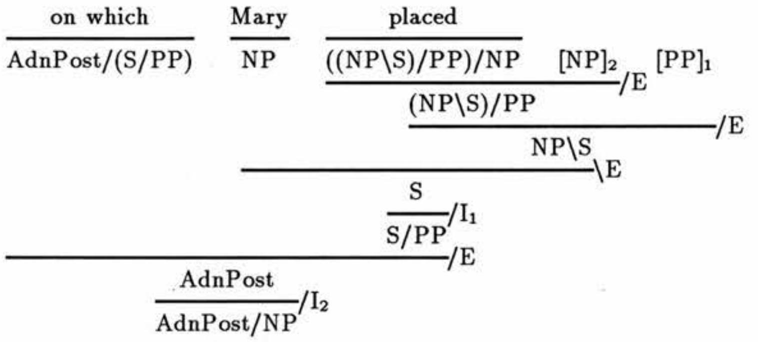
By our definition, neither derivation is dependency-preserving, since in each case an abstracted variable acts as a functor ($f^{\mathrm{NP}\to(\mathrm{NP}\to \mathrm{S})}$ in the first case, $g^{\mathrm{NP}\to\mathrm{PP}}$ in the second case). But each time the argument of the abstracted functor is not a constant but another abstracted variable, so that the 'headless' element is not a lexical item.

Thus we see that definition 3 enables us to make some subtle distinctions. For example, the underlined phrase in (21) will be analysed as a dependency constituent, whereas that in (22) will not:

(21) the tray <u>on which Mary placed</u> the cake

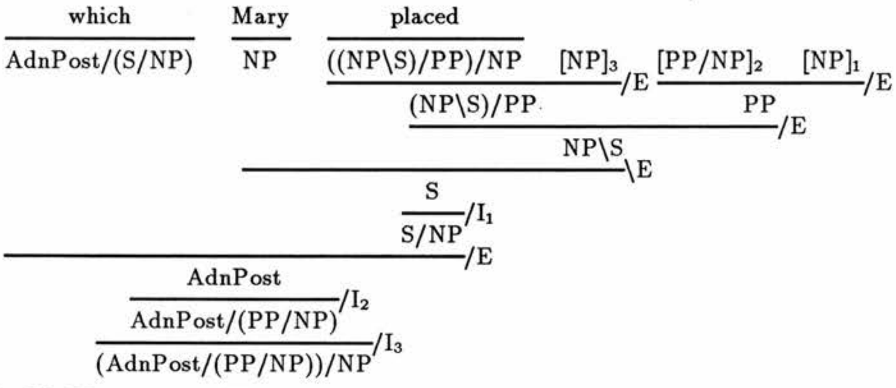
(22) the tray <u>which Mary placed</u> the cake on

(23)



λy^{NP}[**on-which** $(\lambda x^{\mathrm{PP}}$[**placed** y x **mary**])]

(24)



$\lambda y^{\mathrm{NP}} \lambda g^{\mathrm{NP}\to\mathrm{PP}}$[**which** $(\lambda x^{\mathrm{NP}}$[**placed** y $(g\ x)$ **mary**])]

In the first case only complete arguments of *placed* are abstracted, but in the second the variable g is abstracted, which is itself a functor over the abstracted variable x. This seems to capture the intuition that in (21) the relative pronoun *on which* fills an argument role of the verb *placed*, whereas in (22) the relative pronoun *which* fills an argument role not of *placed* itself but of the prepositional argument of *placed* (which is outside the string).

To summarize this section, we have attempted to capture the intuitive notion of dependency within the Lambek calculus, and from it derived a flexible notion of constituency. We have thus abandoned the assumption that any string that can be assigned a type must be a constituent. Let us now see how this notion can be applied to problems of coordination and incremental processing.

3 Coordination

3.1 Coordination of Dependency Constituents

A claim often made in favour of flexible categorial grammars is their ability to deal with a large proportion of coordination phenomena by means of a single principle, usually assumed to be as follows:

Hypothesis 1 *Two (or more) strings may be coordinated to give a string of type X iff each has type X.*

For the purposes of this discussion we shall treat the two directions of implication in hypothesis 1 as two separate hypotheses. One direction may be phrased as follows:

Hypothesis 2 *If two (or more) strings can be given the same type X, then they can be coordinated to give a string of type X.*

We shall discuss the converse later (hypothesis 4).

The precise mechanism used to implement hypothesis 2 varies from formulation to formulation (e.g. polymorphic types for conjunctions, syncategorematic rule schemas), but the principle remains the same. In this section we are concerned not with the mechanism for coordination, but with the characterization of what strings can be coordinated. To save space in examples, instead of giving full derivations we shall usually merely bracket each conjunct, and specify the type that must be assigned to each conjunct (and to the coordinate structure) for the derivation to proceed.

If we are working within the full Lambek calculus, hypothesis 2 holds true in a large class of cases. For instance, we can clearly coordinate strings that form standard phrase structure constituents, like (25) below:

(25) John [sang some songs] and [played the piano].
Type of each conjunct: NP\S

We can also deal with cases like (26) and (27), both of which are usually classified under the heading of 'non-constituent' coordination:

(26) John [will buy] and [may eat] the beans.
Type of each conjunct: (NP\S)/NP

(27) John loves [Mary madly] and [Sue passionately].
Type of each conjunct: (((NP\S)/AdvPost*)/NP)\(NP\S)

In our terms, (25) and (26) both involve coordination of dependency constituents, while (27) involves coordination of non-dependency constituents. Although these three examples look very different, in each case the structure of each conjunct is the same, in the sense that each consists of a sequence of words of the same lexical types. Hypothesis 2 will always allow coordination in such cases, since **L** can always give the same type to two strings with the same structure (in this sense).

More interesting are cases where the conjuncts have different structures, such as (28) and (29):

(28) John [plays the piano] and [sings].
Type of each conjunct: NP\S

(29) John [bought] and [may eat] the beans.
Type of each conjunct: (NP\S)/NP

However, the unacceptability of (30) shows that hypothesis 2 overgenerates:[4]

(30) *John loves [Mary madly] and [Sue].
Type of each conjunct: (((NP\S)/AdvPost*)/NP)\(NP\S)

In (28) and (29) both conjuncts are again dependency constituents, but in (30) the first conjunct is not. Note that even if we took modifiers to be heads we could still give the conjuncts in (30) the shared type ((NP\S)/NP)\(NP\S).

In general, the restrictions on coordination of unlike structures seem to be stronger than those on coordination of like structures. For example, (31) appears to be acceptable, but (as noted by Steedman) (32) does not:

(31) [I believe that John] and [Harry thinks that Mary] is a genius.
Type of each conjunct: S/(NP\S)

(32) *[I believe that John] and [Mary] is a genius.
Type of each conjunct: S/(NP\S)

Note once more that both conjuncts in (31) and the first conjunct is (32) are not dependency constituents. Similarly, (33) seems acceptable, but not (34):

(33) [two small] and [three large] oranges
Type of each conjunct: NP/(AdnPre*\N)

(34) *[two small] and [three] oranges
Type of each conjunct: NP/(AdnPre*\N)

Here, the conjuncts in (33) and the first conjunct in (34) are not dependency constituents if modified elements are analysed as heads (although they would be if modifiers were analysed as heads).

This evidence suggests that there is some correlation between coordinability and dependency constituency. In the examples where both conjuncts are dependency constituents, namely (25), (26), (28) and (29), coordination is always possible. This suggests that hypothesis 2 might be replaced by the following weaker claim:

Hypothesis 3 *If two (or more) dependency constituents can be given the same type X, then they can be coordinated to give a string of type X.*[5]

But in the examples where at least one conjunct is not a dependency constituent, coordination appears to be restricted to cases like (27), (31) and (33) where the conjuncts are structurally similar; it is not possible for the other examples (30), (32) and (34). In order to discriminate between these cases we need a more precise notion of 'like structure', which we shall discuss in the next subsection.

So far we have not examined the converse claim to hypothesis 2:

---

[4] The intended reading of (30) is that John loves Mary madly and John loves Sue. This should not be confused with the orthographically similar elliptical construction, with the meaning that John loves Mary madly and John loves Sue madly, which would have a marked intonational break before *and*.

[5] However we have no explanation for (i), which would be allowed under hypothesis 3:

(i) *I believe [that] and [that Harry thinks that] Mary is a genius.
Type of each conjunct: SP/S

Hypothesis 4 *If two (or more) strings can be coordinated to give a string of type X, then each can be given type X.*

In order to maintain this hypothesis we must have some mechanism for dealing with well-known cases of 'unlike category coordination' such as (35):

(35) John is [lucky] and [a rogue].

Whatever the type of *lucky and a rogue*, it must be assignable to both *lucky* and *a rogue.* If we say that these two items have only the lexical types AdnPre and NP respectively, and that *is* is lexically ambiguous between (NP\S)/AdnPre and (NP\S)/NP, then we are forced to conclude that no type is assignable to *lucky and a rogue.* But if we give *lucky* and *a rogue* an additional shared lexical type, PredP say, and give *is* the single lexical type (NP\S)/PredP, then assigning the type PredP to *lucky and a rogue* is consistent with hypothesis 4. Alternatively we might extend **L** with boolean operators, as suggested in Morrill (1990), to achieve a similar effect.

From hypotheses 3 and 4 it follows that two (or more) dependency constituents can be coordinated to give a string of type X iff each has type X.

3.2 Coordination of Non-Dependency Constituents

Hypothesis 3 puts no restrictions on the internal structure of the two dependency constituents that are coordinated, as we can see by comparing (25) and (28), or (26) and (29). But even when the conjuncts themselves are not dependency constituents, there appears to be no restriction on the internal structure of any substring that is a dependency constituent. For example, (36) and (37) are both as good as (27):

(36) John loves [Mary madly] and [the young woman passionately].

(37) John loves [Mary madly] and [Sue with great ardour].

This would appear to suggest a form of 'constituentwise' coordination, in which the conjuncts are matched dependency constituent for dependency constituent. We may formalize this as follows:

Definition 6 *Let α be any string. Suppose α can be divided into substrings $\langle\alpha_1,\ldots,\alpha_n\rangle$ (where $n \geq 1$) such that each α_i is a dependency constituent, but there is no i such that $\alpha_i\alpha_{i+1}$ is a dependency constituent. (It is easily shown that such a division must exist and be unique.) This is then called a division of α into* maximal dependency constituents (MDCs).

So for instance the division into MDCs of *John bought* is ⟨*John bought*⟩; the division into MDCs of *the young woman passionately* is ⟨*the young woman*, *passionately*⟩. We define further:

Definition 7 *A set of strings $\alpha, \beta, \ldots$, each with n MDCs $\langle\alpha_1,\ldots,\alpha_n\rangle$, $\langle\beta_1,\ldots,\beta_n\rangle$, ..., are* parallel *if for each i $\alpha_i, \beta_i, \ldots$ may be given the same type.*

A similar notion might be defined by including the product connective; for product-based treatments of coordination see e.g. Wood (1988), Bouma (1989).

Given this definition, we may make the following claim, which subsumes hypothesis 3:

Hypothesis 5 *Two (or more) parallel strings of type X may be coordinated to give a string of type X.*

Hypothesis 5 claims that any two (or more) strings of type X which each have n MDCs of types $\langle X_1, \ldots, X_n \rangle$ can be coordinated to give a string of type X. This covers a large class of examples of classical 'non-constituent coordination' not covered by hypothesis 3, including (27), (31), (33), (36) and (37) (repeated below as (38) to (42)):[6]

(38) John loves [Mary madly] and [Sue passionately].
Type of each conjunct: (((NP\S)/AdvPost*)/NP)\(NP\S)
Types of MDCs of each conjunct: ⟨NP, AdvPost⟩

(39) [I believe that John] and [Harry thinks that Mary] is a genius.
Type of each conjunct: S/(NP\S)
Types of MDCs of each conjunct: ⟨S/S, NP⟩

(40) [two small] and [three large] oranges
Type of each conjunct: NP/(AdnPre*\N)
Types of MDCs of each conjunct: ⟨NP/N, AdnPre⟩

(41) John loves [Mary madly] and [the young woman passionately].
Type of each conjunct: (((NP\S)/AdvPost*)/NP)\(NP\S)
Types of MDCs of each conjunct: ⟨NP, AdvPost⟩

(42) John loves [Mary madly] and [Sue with great ardour].
Type of each conjunct: (((NP\S)/AdvPost*)/NP)\(NP\S)
Types of MDCs of each conjunct: ⟨NP, AdvPost⟩

It also predicts, given some mechanism for dealing with constructions such as (35), that (43) will be grammatical:

(43) John is [pious in church] and [a rogue in the tavern].

The converse of hypothesis 5 (which subsumes hypothesis 4) also appears to be true:

Hypothesis 6 *If two (or more) strings can be coordinated to give a string of type X, then they are parallel strings of type X.*

For example, in none of (30), (32) and (34) (repeated below as (44) to (46)) are the intended conjuncts parallel, since in each case they contain different numbers of MDCs, and indeed coordination is not possible in any of them:

(44) *John loves [Mary madly] and [Sue].
Type of each conjunct: (((NP\S)/AdvPost*)/NP)\(NP\S)
Types of MDCs of first conjunct: ⟨NP, AdvPost⟩
Types of MDCs of second conjunct: ⟨NP⟩

---

[6] It must be noted that hypothesis 5 overgenerates in some cases, e.g.:

(i) *John told [Mary Bill] and [Fred Sue] was coming.
Type of each conjunct: ((((NP\S)/S)/NP)\(NP\S))/(NP\S)
Types of MDCs of each conjunct: ⟨NP, NP⟩

This example involves two functor abstractions rather than one in the formation of each conjunct, and so the association between the MDCs is in some sense weaker than in previous examples. However, some similar constructions can be acceptable, such as the following:

(ii) John told [the mothers that their daughters] and [the fathers that their sons] were all at the party.

(45) *[I believe that John] and [Mary] is a genius.
Type of each conjunct: S/(NP\S)
Types of MDCs of first conjunct: ⟨S/S, NP⟩
Types of MDCs of second conjunct: ⟨NP⟩

(46) *[two small] and [three] oranges
Type of each conjunct: NP/(AdnPre*\N)
Types of MDCs of first conjunct: ⟨NP/N, AdnPre⟩
Types of MDCs of second conjunct: ⟨NP/N⟩

Note that if we took modifiers to be heads both *two small* and *three* would be analysed as dependency constituents, and hence (46) would be incorrectly predicted to be grammatical.

The present account also rules out examples such as the following two:

(47) ?I talked [about cricket to Edna] and [to Eric about chess].

(48) ?*I gave [John a pencil] and [a pen to Mary].

It might be possible to generate (47) by some relaxation of the definition of parallelism, but there is no obvious general way to generate (48).

To summarize, we may combine hypotheses 5 and 6 into the following principle:

Hypothesis 7 *Two (or more) strings may be coordinated to give a string of type X iff each has type X and they are parallel.*

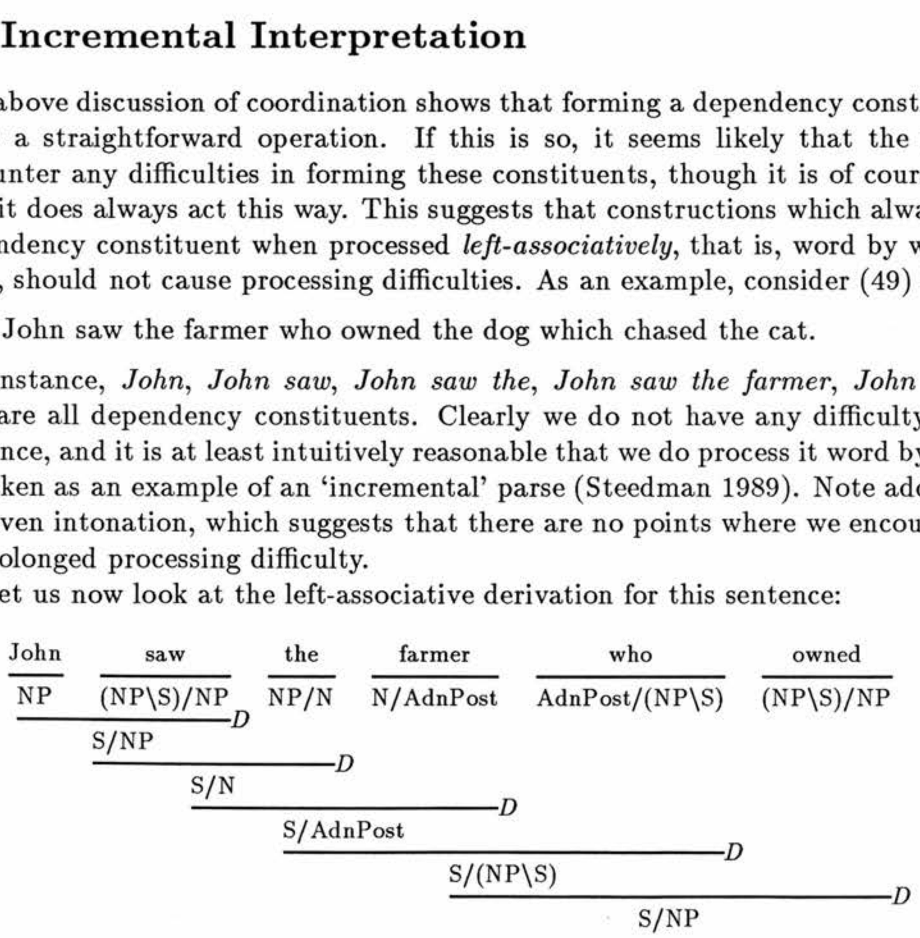
4 Incremental Interpretation

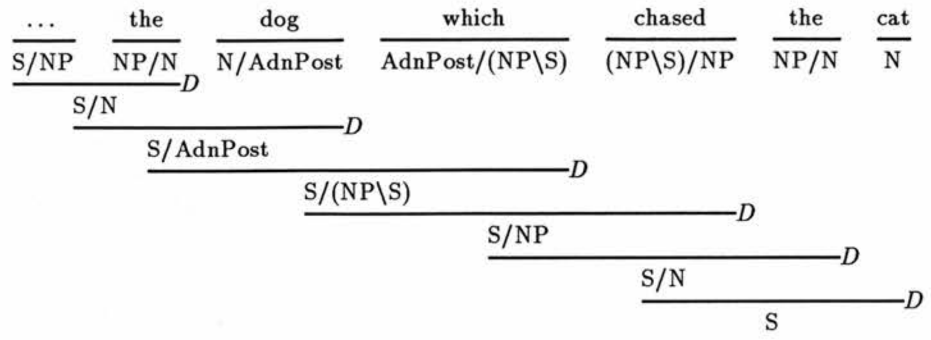
The above discussion of coordination shows that forming a dependency constituent is in some sense a straightforward operation. If this is so, it seems likely that the parser does not encounter any difficulties in forming these constituents, though it is of course not necessary that it does always act this way. This suggests that constructions which always form a single dependency constituent when processed *left-associatively*, that is, word by word from left to right, should not cause processing difficulties. As an example, consider (49) below:

(49) John saw the farmer who owned the dog which chased the cat.

For instance, *John*, *John saw*, *John saw the*, *John saw the farmer*, *John saw the farmer who* are all dependency constituents. Clearly we do not have any difficulty processing this sentence, and it is at least intuitively reasonable that we do process it word by word. This can be taken as an example of an 'incremental' parse (Steedman 1989). Note additionally that it has even intonation, which suggests that there are no points where we encounter momentary or prolonged processing difficulty.

Let us now look at the left-associative derivation for this sentence:

(50)




We have marked each combination with D, in order to indicate that they are combinations that form dependency constituents. At no point is there any need to abstract over a functor in the meaning representation. The simplicity of the operations seems to be reflected in the ease we have processing it.

Similar effects occur if we extend the sentence with further relative clauses:

(51) John saw the farmer who owned the dog which chased the cat which followed the mouse which nibbled the cheese.

The sentence does not become obviously harder to process when we extend it in this way. If we do a left-associative parse, we again see that the types generated do not become complex and we never have the need to make any non-dependency combinations. Hence we have support for the hypothesis that the formation of dependency constituents is easy for the parser.

We should note in contrast that a theory that does not allow the parser to form dependency constituents at will may encounter problems with such sentences as these. Standard phrase structure grammars would give (49) the following constituent structure:

(52) [John [saw [the [farmer [who [owned [the [dog [which [chased [the [cat]]]]]]]]]]].

A simple association between grammar and parser would assume that the parser could only form constituents when it reached the very end of this sentence. This is utterly out of keeping with the fact that the sentence, and its extensions, are straightforward to process. Hence presumably such an account has to make use of parsing principles that are not in direct correspondence with the grammar. A processing account based on dependency constituents, on the other hand, does lend itself to a parsimonious relationship between grammar and parser. See Steedman (1989) for related arguments.

So if the calculus allows a combination that forms a dependency constituent, the parser is able to make that combination. But what happens when the left-associative derivation does not solely form dependency constituents? Clearly we are still able to interpret many such sentences:

(53) The tall boy died.

There is no dependency relation between *the* and *tall*, so the left-associative parse does not always form dependency constituents. However we encounter no problem with this sentence. It seems that we can either make a non-dependency association between the first two words, or we wait until we reach *boy*, which is in a dependency relation with both words. In other words, is our method of processing entirely left-associative, or is it restricted to the formation of dependency constituents? We are not going to suggest an answer to this question, but simply discuss both alternatives.

Let us consider a nested construction as defined by Chomsky (1965):

(54) John saw the cat which the dog which the farmer owned chased.

Such constructions are uncontroversially hard to process. This is in sharp contrast to (49), even though they are 'grammatically' of similar complexity. Unlike (49), (54) does not have a left-associative derivation that only forms dependency constituents, and, unlike (53), this does not seem to be a 'local' effect. In (54), the initial substring *the cat which* forms a dependency constituent, but not *the cat which the*, so that *the dog which* must be parsed separately from *the cat which*. Similarly *the farmer* must be parsed separately from *the cat which* and *the dog which*. At the point before *owned*, then, we find that we have three unrelated dependency constituents, *the cat which*, *the dog which* and *the farmer*. Intonationally as well, we can distinguish units *John saw the cat which*, *the dog which* and *the farmer*, which are the same as the three dependency constituents.

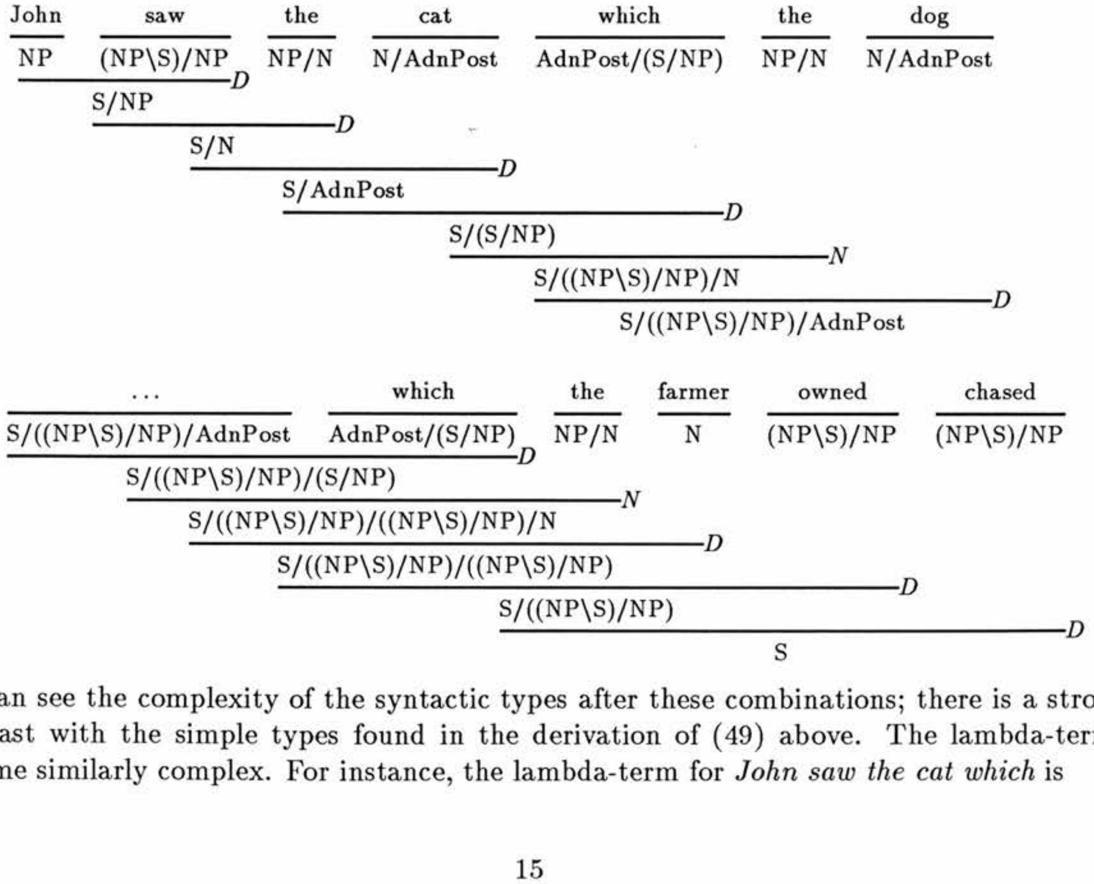
The point is even clearer when we contrast the effect found by recursing the construction in (51) above and in (55) below:

(55) John saw the girl who the rat which the cat which the dog which the farmer owned chased scratched bit.

(51) is no more complex to parse than (49), and remains parsable if we recurse the construction still further. The same cannot be said of (55), which is virtually incomprehensible. Similarly, at the point before *owned* in (55), we have four unrelated dependency constituents. Each extension of (54) increases the number of unrelated dependency constituents, in contrast to extensions of (49).

We could parse nested constructions like (54) purely left-associatively, in which case we are forced to allow the parser to make combinations that are not dependency-preserving (here notated by N):

(56)



We can see the complexity of the syntactic types after these combinations; there is a strong contrast with the simple types found in the derivation of (49) above. The lambda-terms become similarly complex. For instance, the lambda-term for *John saw the cat which* is

$$\lambda f^{\mathrm{NP}\rightarrow\mathrm{S}}[\mathbf{saw}\ (\mathbf{the}\ (\mathbf{cat}\ (\mathbf{which}\ f)))\ \mathbf{john}],$$

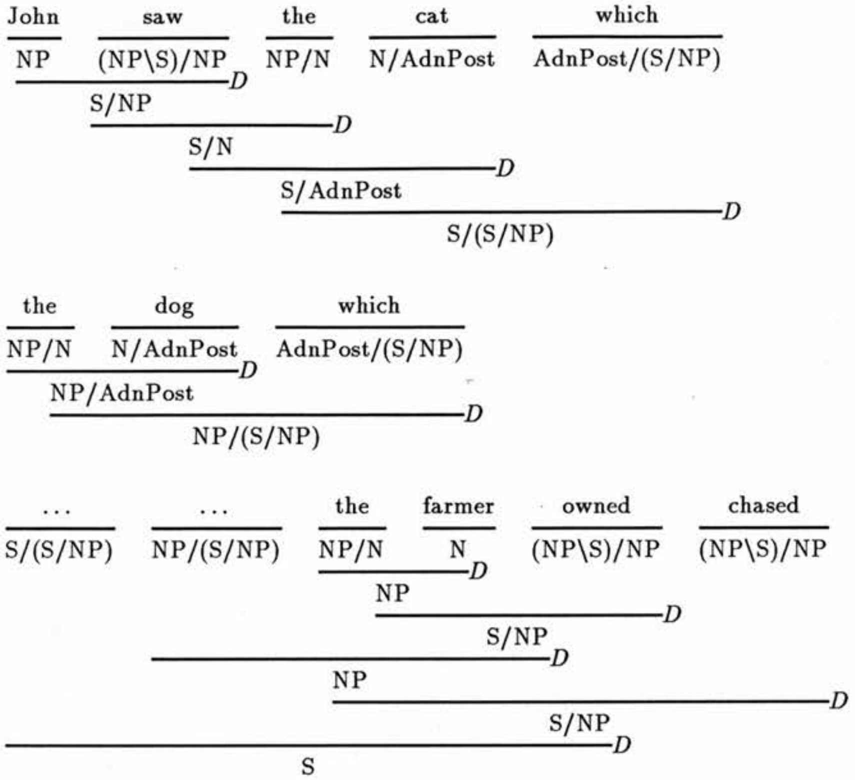
but the term for *John saw the cat which the* is

$$\lambda y^{\mathrm{N}}\lambda g^{\mathrm{NP}\rightarrow(\mathrm{NP}\rightarrow\mathrm{S})}[\mathbf{saw}\ (\mathbf{the}\ (\mathbf{cat}\ (\mathbf{which}\ (\lambda x^{\mathrm{NP}}[g\ x\ (\mathbf{the}\ y)]))))\ \mathbf{john}].$$

In the second lambda-term, the abstracted variable g acts as a functor, unlike f in the first lambda-term.

It is also possible that the parser does not make these non-dependency combinations at all, but instead uses some device like a stack to store dependency constituents (see Ades and Steedman 1982). This means that our derivation is no longer fully left-associative. Here we represent the maximally incremental derivation containing only dependency combinations:

(57)



Another example of the contrast between nested and non-nested constructions can be seen in (21) and (22), repeated below as (58) and (59):

(58) the tray on which Mary placed the cake

(59) the tray which Mary placed the cake on

At the verb *put*, we have a single dependency constituent in (58) but not in (59). As pointed out in Pickering and Barry (1989, 1990), we can recurse each construction on the object noun phrase, for instance in the sentences below:

(60) John found the box in which I put the tray on which Mary placed the cake.

(61) John found the box which I put the tray which Mary placed the cake on in.

(61) is very hard to process, while (60) is easy to process. Our account predicts this directly, because (61) has three unrelated dependency constituents, *John found the box which*, *I put the tray which* and *Mary placed the cake*, at the point before the word *on*. This is not the case in (60), because each time we reach an embedded verb, we can form a single dependency constituent from all of the preceding material. In (61) we cannot form a single dependency constituent when we reach each embedded verb. Pickering and Barry contrast a dependency based account with an account making use of empty categories, and, given that the construction in (60) can be recursed, show that the latter account makes the wrong predictions for sentences like (60).

We have demonstrated that an account of sentence processing based on the construction of dependency constituents makes reasonable predictions about processing complexity. We see this as an explanation of Chomsky's (1965) claim that nested constructions are hard to process, and hence why they are so rare in natural language use. It seems therefore that the notion of a dependency constituent is useful both in explaining linguistic data such as coordination, and explaining facts about sentence processing: combining elements to form a dependency constituent is easy, but combining elements to form a non-dependency constituent is hard and not automatic.

5 Comparisons and Conclusions

5.1 Comparisons with Combinatory Categorial Grammar

We have argued that the assumption of type-constituent correspondence within the Lambek calculus must be dropped if the notion of constituency is to retain any linguistic relevance, and dealt with this by defining the notion of dependency constituency. Let us contrast this notion of constituency with the CCG notion of constituency (Steedman 1987).

As mentioned in the Introduction, CCG uses a particular set of rules, known as *combinatory rules*, to assign types to all and only those strings that the theory regards as constituents. Steedman uses a combination of **L**-valid and **L**-invalid rules, but for the purposes of this discussion we shall restrict our attention to the former, since the **L**-invalid rules are highly restricted in their applicability and might reasonably be regarded as peripheral. Thus we are viewing CCG as a subset of **L**.

CCG puts two restrictions on possible type combinations. First, it restricts them to those that can be achieved by means of a particular set of binary and unary rules, including the following:

(62) Forward application (indexed >)
$X/Y \;\; Y \Rightarrow X$
Backward application (indexed <)
$Y \;\; Y\backslash X \Rightarrow X$
Forward composition (indexed >**B**)
$X/Y \;\; Y/Z \Rightarrow X/Z$
Forward type-raising (indexed >**T**)
$X \Rightarrow Y/(X\backslash Y)$

However, CCG does not include other **L**-valid inferences such as forward division ($X/Y \Rightarrow (X/Z)/(Y/Z)$) or commutation ($(Y\backslash X)/Z \Rightarrow Y\backslash(X/Z)$). (The choice of rules is made on the basis of specific linguistic phenomena, so that there is no straightforward characterization of which rules are included and which are not.) Secondly, CCG puts type-specific restrictions on the use of the above rules. For instance, there is a stipulation that the use of the forward type-raising rule is restricted to subjects:

(63) $\mathrm{NP} \Rightarrow \mathrm{S}/(\mathrm{NP}\backslash \mathrm{S})$

Although there is a considerable overlap between CCG constituents and dependency constituents, the existence of type-raising in CCG (which is not dependency-preserving by our definition) generates some CCG constituents that are not dependency constituents. For instance, let us return to our initial examples in section 2 (repeated here as (64) and (65)):

(64) a. The dog runs.

b. John likes Mary.

c. John will see Mary.

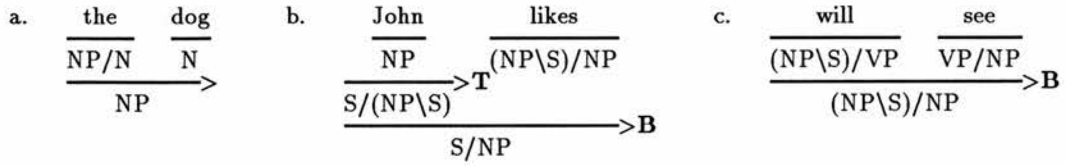
(65) a. The dog runs.

b. I think that Harry left.

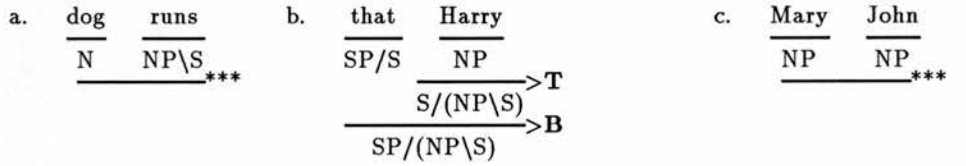
c. I showed Mary John.

Recall that the underlined fragments in (64) were analysed as dependency constituents, but not those in (65). Steedman's analysis can give a type to the fragments in (64a–c) and (65b), but not to those in (65a) or (65c):

(66)



(67)



The only point of disagreement here is *that Harry* in (65b), which is a CCG constituent but not a dependency constituent. It would however not be a CCG constituent if the non-dependency-preserving operation of type-raising were absent from the grammar. (Note that removing type-raising from the above set of rules would also stop *John likes* in (64b) from being a constituent, but this could be achieved by means of the dependency-preserving operation of commutation on *likes*.) The above account forces Steedman to analyse *I believe that John* as a constituent of type S/(NP\S), and hence to allow the forbidden coordination in (32) (repeated here as (68)):

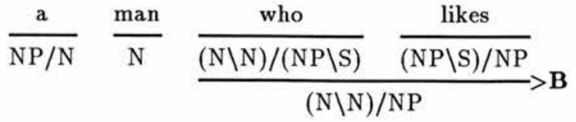
(68) *[I believe that John] and [Mary] is a genius.

It is also possible for dependency constituents not to be CCG constituents. For example, Steedman treats modifiers as functors, but forbids modified elements to type-raise over them, in order to account for their status as islands, as in (69):

(69) *Beans, I met a man who likes.

A man who likes will be analysed as a dependency constituent irrespective of whether modifiers or modified elements are assumed to be heads. We therefore cannot expect dependency constituents to give us a complete account of island constraints. However, the absence of type-raising over modifiers and of commutation means that it cannot be analysed as a CCG constituent:

(70)

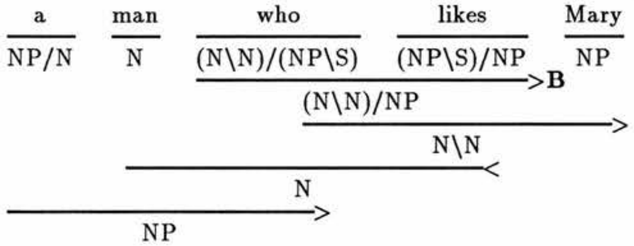


Steedman's attempt to capture island constraints merely by blocking derivations such as the above makes incorrect predictions about coordination and processing of such strings. For instance, it rules out coordinations such as the following:

(71) Most people like, but I know a man who hates, sonatas by Mozart.

This is allowed in our account because the conjuncts are both dependency constituents of type S/NP. More generally, Steedman's account always forbids a noun to combine with an incomplete relative clause. This has serious implications for processing, because it predicts that a noun cannot be combined with a following relative clause until the entire relative clause is processed. For instance, the maximally incremental derivation of *a man who likes Mary* in CCG is as follows:

(72)



Using arguments analogous to those in the previous section, this would incorrectly predict that multiple relative clause constructions such as (51) (repeated here as (73)) should *always* become difficult to process at some point, irrespective of the structure of the relative clauses themselves:

(73) I saw the farmer who owned the dog which chased the cat which followed the mouse which nibbled the cheese.

Steedman's account allows some combinations (e.g. *who owned the*), but the number of elements that have to be remembered independently still increases when the sentence is extended.

Even if the CCG rules were modified to generate all and only dependency constituents, it would still be impossible to account for phenomena that involved the formation of non-dependency constituents, such as the coordination in (27) (repeated here as (74)):

(74) John loves [Mary madly] and [Sue passionately].

The only way to generate such constructions under an assumption of type-constituent correspondence is to allow non-dependency-preserving rules. For example, Dowty (1988) deals with such constructions by using combinatory rules of backward composition and backward type-raising (he assumes the latter to be lexical):

(75) Backward composition (indexed <**B**)

$Z\backslash Y \quad Y\backslash X \Rightarrow Z\backslash X$

Backward type-raising (indexed <**T**)

$X \Rightarrow (Y/X)\backslash Y$

Although both rules are **L**-valid, only backward composition is dependency-preserving. Thus the conjuncts in (74) are constituents under Dowty's analysis, but not dependency constituents:[7]

---

[7] Here we follow Dowty in taking modifiers to be heads, but a similar argument can be constructed if modified elements are taken to be heads.

(76)
$$\dfrac{\dfrac{\dfrac{\text{Mary}}{\text{NP}}}{((\text{NP}\backslash\text{S})/\text{NP})\backslash(\text{NP}\backslash\text{S})}{<}\mathbf{T} \qquad \dfrac{\text{madly}}{(\text{NP}\backslash\text{S})\backslash(\text{NP}\backslash\text{S})}}{((\text{NP}\backslash\text{S})/\text{NP})\backslash(\text{NP}\backslash\text{S})}{<}\mathbf{B}$$

This means that Dowty is unable to prevent the forbidden coordination in (30) (repeated here as (77)), since both *Mary madly* and *Sue* can be analysed as constituents of type ((NP\S)/NP)\(NP\S):

(77) *John loves [Mary madly] and [Sue].

5.2 Further Applications

We would not be surprised if the notion of dependency constituency has other linguistic applications. Steedman has argued that flexible categorial grammars allow considerable generalizations between the constituents that are needed to deal with coordination, incremental interpretation, extraction and intonational structure. We agree with this view, except that we claim that the relevant notion is dependency constituency rather than CCG constituency. This paper has covered only coordination and incremental interpretation, but there are a number of possible applications to other linguistic phenomena. For example, the unacceptability of (78) suggests that extraction is subject to the restriction that the non-extracted part of the string must be a dependency constituent:[8]

(78) *How many boys do you think in ten like football?

However, as we have seen above, this condition is not sufficient in itself to capture all island constraints.

Dependency constituency also appears to be relevant to an account of intonational structure. Steedman (1990) shows that the range of possible intonations is greater than those that are most obviously allowable by traditional phrase structure, but that there are still many strings which do not form possible intonational units. Again we believe that the range of possible intonational units can be derived from the notion of dependency constituency.

In summary, we believe that the approach taken in this paper has the following two advantages. Firstly, the notion of constituency is defined in a principled fashion from the notion of dependency. Secondly, it is possible to give types to strings that are not constituents. This appears to be theoretically elegant and also to explain a considerable range of data.

Acknowledgements

We would like to thank Robin Cooper, Elisabet Engdahl, Mark Hepple, Janne Johannessen and Glyn Morrill for their comments and suggestions. We acknowledge the support of SERC Research Studentship 88306971 and ESRC Research Studentship C00428722002.

References

Ades, A.E. and Steedman, M.J. (1982). On the order of words. *Linguistics and Philosophy 4*, 517–558.

---

[8] Apparent counterexamples such as *Which boy do you know that plays football?* are presumably examples of extraposition, which will have to be handled separately.

Bouma, G. (1989). Efficient processing of flexible categorial grammar. In *Proceedings of the Fourth Conference of the European Chapter of the ACL*, UMIST, Manchester.

Chomsky, N. (1965). *Aspects of the Theory of Syntax.* MIT Press, Cambridge, Massachusetts.

Dowty, D. (1988). Type raising, functional composition, and non-constituent conjunction. In R. Oehrle, E. Bach and D. Wheeler (Eds), *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht.

Hudson, R.A. (1987). Zwicky on heads. *Journal of Linguistics 23*, 109–132.

Moortgat, M. (1988). *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus.* Foris, Dordrecht.

Morrill, G. (1990). Grammar and logical types. This volume.

Morrill, G., Leslie, N., Hepple, M. and Barry, G. (1990). Categorial deductions and structural operations. This volume.

Pickering, M. and Barry, G. (1989). Processing extractions without gaps. Research Paper EUCCS/RP–36, Centre for Cognitive Science, University of Edinburgh.

Pickering, M. and Barry, G. (1990). Sentence processing without empty categories. Ms., Centre for Cognitive Science, University of Edinburgh.

Pollard, C. and Sag, I. (1987). *An Information-Based Approach to Syntax and Semantics: Volume 1 Fundamentals.* CSLI Lecture Notes, Number 13, Stanford.

Steedman, M.J. (1987). Combinatory grammars and parasitic gaps. *Natural Language and Linguistic Theory 5*, 403–439.

Steedman, M.J. (1989). Grammar, interpretation and processing from the lexicon. In W. Marslen-Wilson (Ed.), *Lexical Representation and Process*, MIT Press, Cambridge, Massachusetts.

Steedman, M.J. (1990). Syntax and intonational structure in a combinatory grammar. To appear in G.T.M. Altmann (Ed.), *Cognitive Models of Speech Processing: Psycholinguistic and Computational Perspectives*, MIT Press, Cambridge, Massachusetts.

Wood, M. (1988). *A Categorial Syntax for Coordinate Constructions.* PhD thesis, University College London. Available as Technical Report UMCS–89–2–1, Department of Computer Science, University of Manchester.

Zwicky, A.M. (1985). Heads. *Journal of Linguistics 21*, 1–29.