ANTHONY E. ADES AND MARK J. STEEDMAN

ON THE ORDER OF WORDS

1. Introduction

A complete theory of human grammatical performance must include components of two distinct kinds. The first is a grammar, defining a set of strings which comprises all and only the sentences of the language in question. The second component is a processor, or device which applies the rules of the grammar so as to either produce or analyse the grammatical strings. The processor can be characterised in two quite independent respects. It may be defined as a particular variety of automaton, characterised as (for example) working from left to right or from right to left through the text, as applying the rules of the grammar 'top down' or 'bottom up', or as doing semantic interpretation 'in parallel' or 'in series' with syntactic processing. But the processor can also be characterised by the particular mechanism which it uses to resolve ambiguities as to which particular rule of grammar should be applied at a particular point in analysis or generation. Mechanisms of this last kind must prevent the explosion of processing demands and the inefficiencies that will result if such ambiguities are allowed to proliferate.

What follows is a theory of competence in the accepted sense: we shall have nothing further to say about efficiency, and very little to say about possible mechanisms for the resolution of ambiguity. However, there is every reason for a theory of competence to be straightforwardly related to what is known about human linguistic performance. In particular, it is an advantage in a competence theory if it is clear how it could be turned into a processor which works 'from left to right' through the text, and appears to deal with semantics as nearly as possible in parallel with syntax, as Marslen-Wilson (1973), Tyler and Marslen-Wilson (1977), Marslen-Wilson and Tyler (1980), Crain (1980), and others have suggested that the human parser does. It is only once it is clear how a human listener could so process all and only the grammatical strings of a language that the further limitations that are imposed by the mechanisms which resolve local ambiguities (and which on occasion misresolve them) can be investigated. The goal of the present paper is therefore to outline a proposal which is simultaneously a competence theory and a theory of what the human listener would have to do to construct any

Linguistics and Philosophy 4 (1982), 517–558. 0165–0157/82/0044–0517$04.20

given parsing for all and only the grammatical strings of the language, without regard to the logically posterior question of ambiguity resolution or how that *particular* parse might be arrived at.

Because it is the process of analysis that has been most extensively studied by psychologists, the model is presented as an idealised recogniser, rather than as the idealised generator that is more common in modern linguistics. The choice of direction has of course no significance for the theory of grammar. For the purpose at hand it is no more necessary to explain how this 'recogniser' might be equipped to handle ambiguities efficiently than it is for generative grammarians to explain how their 'generators' might choose which base rules to apply.

1.1. *The linguistic problem*

What one might have expected of natural languages is that their grammars would be context free (CF), with all the attendant benefits of a straightforward relation between syntax and semantics, as well as various desirable effects upon learnability and processing demands. But of course Transformational Grammar (TG) is founded on the suggestion that natural languages can *not* be usefully described with CF grammars. Most importantly, transformationalists have pointed out that significant generalisations are liable to be missed if a grammar of English fails to describe such sets of sentences as the following, not merely as grammatical, but also as related.

(1) (a) He must love her.
(b) Must he love her?
(c) Her he must love!

Chomsky formalised the relation by deriving such sets from a single underlying form, closely related to (1a). To this underlying structure, generated by a CF base grammar, could apply a number of structure changing transformations, rearranging constituents into the interrogative and topicalised forms (1b, c) among others.

However, a principal concern of Transformational grammarians has been to specify ways in which the form of the transformations that occur in natural languages are more constrained than unrestricted structure changing operations, for it is certain that they are drawn from a very limited class indeed. Moreover, even the transformations that *are* found in a language are subject to unexplained restrictions upon their applicability. For instance, having defined movement transformations that relate the numbered noun phrases in (2a) to the canonical positions marked t_1 and t_2, they must be restricted to prevent them generating (2b), for several linguists have noticed that in sentences where more than one constituent has been 'moved' by transformations, they can be linked to their respective extraction sites by nested, but not crossed, lines (Bach, 1977, p. 150; Fodor, 1978, pp. 51, 52).

(2) (a) (The wood)$_1$ is too rough for (these nails)$_2$ to be easy for me to hammer t_2 into t_1.
(*b*) *(The nails)$_1$ are too blunt for (this rough wood)$_2$ to be easy for me to hammer t_1 into t_2.

What is more, this 'Nested Dependency Constraint' applies with great generality over the languages of the world[1].

The many accounts delimiting the form that transformational rules may take and the conditions under which they may apply constitute and indispensable contribution to the study of language, and the generalisations that have emerged constitute in great part the data upon which the present study rests. However, even the most ingeniously restricted set of transformations and constraints upon them does not explain why it is that only transformations drawn from that set exist, constrained in those particular ways, since they appear to require for their processing an apparatus of a power that would be capable of performing much less restricted operations as well. The task of explaining the form of natural language will only be complete when it is shown that the restrictions stem from the involvement of some specific class of mechanism that is more restricted that which is implicated by transformations.

A particularly important class of constraints are those upon 'unbounded movements' such as the constraint exemplified above, because the negative constraints that prohibit the transformational apparatus from performing an otherwise perfectly computable operation are particularly problematic if the competence grammar is to be directly realisable as a processor, as we have suggested it should be. The only straightforward interpretation of a standard theory grammar as a processor – as opposed to the many less absurd but equally less specified interpretations – is as one which first builds a surface structure, then proceeds via the transformations to a deep structure and an interpretation. But the constraints of Ross (1967) would imply that such a processor first builds a presumably perfectly interpretable structure, via the transformational rules, but then has to reject it after consulting a list of forbidden movements or structures. The filters of Chomsky and Lasnik (1977) have similarly unpalatable implications for such a processor.

An important step towards an explanation has been provided by the

Base Generation Hypothesis, which is broadly represented in the work of Bresnan (1978), Brame (1976, 1978), Gazdar (1981a, 1981b) and Peters (1980), among others. The hypothesis eschews transformations, and generates all constituent orders directly in some form of base grammar. Gazdar (1981b), in particular, has argued persuasively that there is no good reason to regard English or any other language as being more complex than a CF language, though not that it is necessarily best described using a CF grammar alone. Nevertheless, to the extent that, in the interests of capturing generalisations, these systems include rules of grammar which (whilst they themselves only generate a CF language) are nevertheless drawn from classes of rules which potentially have greater than CF power, the question of why natural languages should include rules from such powerful classes remains unanswered. For if the mechanism can compute such rules there is no particular reason for it to confine itself to just the ones which happen to generate CF languages. And if the rules are *not* computed by a processor, then the generalisations remain unexplained[2]. Typical of such special rules are the metarules of Gazdar, and the potentially arbitrary relationships between surface syntax and deep structure or functional/semantic representation of Brame (1978), Bresnan (1978), and the related computational models of Woods (1968) and Marcus (1977), among others. Among the complications attendant upon such rules are increased difficulty of learning, non-isomorphism between syntactic and semantic rules and non-correspondence between rules of grammar and steps in processing.

1.2. *Outline*

The rest of the paper will take the following form. In the next section, we interpret certain constraints on 'movement' as suggesting that a CF grammar or some natural generalisation thereof can be devised for both 'base' and 'movement' phenomena, and point out that such grammars are directly compatible with a psychologically plausible mechanism which proceeds from left to right, parsing and building a semantic interpretation in one pass. We then present a catégorial lexicon and four rule schemata whose function is to determine word and constituent order. A related processor which applies the rules is described, without of course any discussion of the further problem of ambiguity resolution.

In Section 3, the model is applied to the basic forms of English main clauses. We confine ourselves to phenomena of 'unbounded movement' and the class of Root Transformations (Emonds, 1975), as it is these that have proved most resistent to reformulation in nontransformational terms.[3]

Section 4 shows that the phenomena associated with the Left Branch Condition of Ross (1967) can be naturally expressed *within* the grammar, rather than by ad hoc negative constraints exterior to the grammar, as can phenomena which have motivated major constraints upon Root Transformations.

In Section 5, certain extraction constraints are shown to follow from the way in which each of the rule schemata expresses generalisations about the types of semantic-syntactic entities that they combine, and from other general features of the model.

In the final section, the present proposals are briefly compared to some alternatives from both linguistic and psychological points of view.

2. The Model

2.1. *Stacking properties and Context Free-ness*

Both Bach (1977) and Fodor (1978) pointed out that the unacceptability of crossed dependencies illustrated in (2) would be predicted by a left-to-right processor that placed preposed constituents into a Push Down Store (PDS) or stack, and restored them later to their underlying positions. A mechanism of this type had already been introduced in the machine parsing literature. Woods (1973, 119), following a suggestion made by Thorne Bratley and Dewar (1968), showed that preposed items such as relative pronouns could be restored to their canonical position by placing them in a store called HOLD, from which they could later be retrieved when the site of their extraction was encountered. He speculated that the same mechanism could be used for other varieties of movements, and further conjectured that the HOLD store was a PDS. (See also Kaplan, 1973).

Something like the Nested Dependency Constraint seems also to apply to all those constructions which Emonds (1976) classified as arising from Root Transformations[4]:

(3) (a) Whom$_1$ did$_2$ you 0_2 see 0_1.
(b) [Far more serious]$_1$ was$_2$ the leak in the roof 0_2 0_1.
(c) [Leaning against the bedpost]$_1$ was$_2$ a policeman 0_2 0_1.
(d) [Up the street]$_1$ walked$_2$ Bill 0_2 0_1.
(e) [Seldom]$_1$ had$_2$ he 0_2 0_1 heard a more ridiculous proposal.

If, as these data strongly suggest[5], *all* movement is constrained by the nesting property, it becomes tempting to make a significant simplification in the theory. It is suspicious that a stack is needed to constrain movement from canonical base structure positions, for a stack will also be necessary to process the surface strings themselves, which are describable as context free. Our proposal is simply that it is the *same* stack that is responsible both for 'movements' and for processing surface CF grammar. The effect of this proposal is to remove the Nested Dependency Constraint and the stack from the orbit of performance constraints on movement and to make them a principle of grammar. The implication is that both movement and base phenomena are the manifestation of one context free or near-context free system.

This interpretation of the nesting property, like other base-generative approaches, offers the attractive possibility of a syntax and a syntactic processor that are non-autonomous in the strongest sense of the word. Since there is no need for a separate pass through the string to restore displaced constituents to canonical position before semantic interpretation can begin, it follows that semantic interpretations can be assembled in parallel with syntactic processing. Indeed, there is no need for any autonomous syntactic structure to be built at all: the semantic representation can be built directly. Moreover, once an analyser can be made to build semantic interpretations immediately, it can also be made to evaluate subexpressions while the analysis is still in progress. An example of how syntactic processing, semantic interpretation and evaluation can be combined into a single left to right pass is to be found in Davies and Isard (1972). The technique, which is well known to compiler writers (cf. Bolliet, 1968), has been exploited in computer models of language interpretation by, for example, Winograd (1972), Johnson-Laird (1977) and Steedman and Johnson-Laird (1978). A grammar of the base-generative variety therefore offers the attraction of a linguistic theory which is straightforwardly relatable to a considerable amount of evidence suggesting that human listeners not only assemble semantic interpretations as nearly as possible in parallel with syntactic processing, but also use semantics and even reference interactively to guide syntactic and lexical decisions (Marslen-Wilson, 1973; Tyler and Marslen-Wilson, 1977; Marslen-Wilson and Tyler, 1980; Crain 1980)[6].

2.2 *The Categorial Notation*

The rules that follow will be presented using the notation of Categorial Grammar (CG) (Ajdukiewicz, 1935; Bar-Hillel, 1960; Lyons, 1968; Lewis, 1970). A Categorical Grammar consists of a series of lexical entries specifying the syntactic role of each morpheme in the language. Such entries take the following form:

(4) Morpheme: X/Y

For example, determiners bear the category NP/N, identifying them as entities which combine with nouns to yield noun-phrases. For primitive categories Y is null: the category of a noun is simply N.

An item with a category of the form X/Y is to be thought of as a function. For example the category VP/NP of transitive verbs identifies them as functions from NPs onto VPs. In the first place, such functions can be thought of as mapping between syntactic entities. However, the categories can also be thought of as a shorthand for the semantics of the entities in question, and the functions can then be thought of as mapping between semantic entities, such as interpretations, intensions, or whatever. (For example, the category VP/NP of a transitive verb can be thought of as a shorthand for a function from NP intensions onto VP interpretations.) We have relegated to an Appendix our few further remarks concerning the nature of such semantics, and will continue for the most part to use the syntactic shorthand, since for present purposes it makes no difference whether semantic entities are thought of as expressions in Lambda calculus, procedures, or anything else. The important assumption is parallel to the basic assumption of Montague grammar: there is a one-to-one correspondence between syntactic categories and rules of semantic interpretation.

As befits their basically semantic nature, the functions described above do not themselves determine whether their arguments are to appear to their right or to their left. The responsibility for ordering lies with a number of 'combination rules', so called because of their direct relation to operations of a processor. These rules of the grammar are described next.

2.3. *Forward Combination*

The first of the combination rules combines a function with an argument to its right. It can provisionally be written as the following rule schema:

(5) Forward Combination

$$X/Y \; Y \Rightarrow X$$

X and Y are variables and are allowed to match any category.

Consider the following fragment of lexicon:

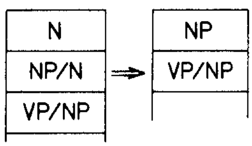
(6) the: NP/N
frog: N
find: VP/NP

An example of the operation of the Forward Combination rule is the following, where the variables X and Y match the atomic categories NP and N, respectively:

(7) $[\text{the}]_{NP/N}\ [\text{frog}]_N \Rightarrow [\text{the frog}]_{NP}$

The Forward Combination rule (5) can be thought of as a rewrite rule schema which happens to have been written down backwards. But it may be easier to imagine a process that from time to time puts the successive words of the string onto a stack, and that this and other combination rules can apply to the top two items on the stack, replacing them with the result. In the case of Forward Combination an X/Y and a Y on top of it will be replaced by the single result X. (Such a 'processor' is related to a class of Non-Deterministic Push Down Automata known as 'Shift and Reduce' parsers). For example, in parsing the phrase *find the frog*, the words *find*, *the*, and *frog*, are placed onto the stack by the automaton. Then the Forward Combination rule applies to reduce $[\text{the}]_{NP/N}$ and $[\text{frog}]_N$ to $[\text{the frog}]_{NP}$, as in the following diagram:

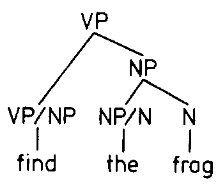
(8)



At this point the conditions for Forward Combination are met again. The VP is constructed from $[\text{find}]_{VP/NP}$ and $[\text{the frog}]_{NP}$.

The above process is isomorphic to the familiar kind of phrase structure tree:

(9)



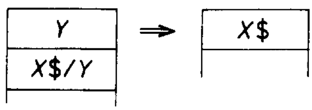
However, it should be remembered that this structure really only describes the successive stages of building a directly evaluable semantic interpretation.

At this point, the basic notation must be elaborated a little. Some verbs, of course, can take more than one argument. For example, *put* is a function from NPs onto a function from PPs onto VPs, implying a category (VP/PP)/NP. It will be notationally convenient for the presentation of the later rules to omit the brackets under a convention of 'left associativity', and to write the category of *put* as VP/PP/NP, where it is understood that the rightmost slash is the 'major' or highest level slash. Then the Forward Combination rule is generalised as follows. We define a string symbol, \$, to be either null or a string of alternating atomic category symbols and slashes beginning with a slash and ending with a symbol, (such as /NP, /PP/NP, /NP/NP/NP, and so on). The form of the Forward Combination rule is now:

(10) Forward Combination
$X\$/Y \ Y \Rightarrow X\$$

In its incarnation as a rule in an acceptor, it can be represented pictorially as follows:

(11)



It is a theoretical advantage of the categorial notation (as for other lexically-based grammars) that it is unnecessary to specify the various expansions of, for example, VP, in the grammar *and* redundantly in the lexicon. All the work is done by the categorization of, for instance, *eat* as VP/NP, *walk*, as VP/PP and *put* as VP/PP/NP. (Of course, any morpheme may have more than one categorization. The selection of the category which is appropriate to a given parsing is a consideration neither of the grammar, nor of the structure of the corresponding automaton per se, but of the ambiguity resolving mechanism).

2.4 *Backwards Combination*

The second combination rule is similar to Forward Combination, except that it combines a function with an argument to its left, or in terms of the processor, one which is beneath it on the stack.

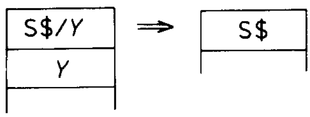
Consider such strings as:

(12) (a) Her he must love.
(b) Who are you looking at?

Each begins with a preposed constituent that has been 'moved' from the sentence that follows. Because the model we are proposing has only one stack with which to combine items, and because our rules are confined to ones that operate only on adjacent items, there is no alternative to regarding the strings *He must love* and *are you looking at* as being functions of type S/NP, that map (object) NP's onto sentences. Thus, both sentences (12) are instances of NP followed by S/NP. Because in English no other category (at least among the ones we are considering here) combines with an argument to its left, we stipulate that Backwards Combination is restricted to S-nodes, and write it thus:

(13) Backwards Combination
$$Y\ S\$/Y \Rightarrow S\$$$

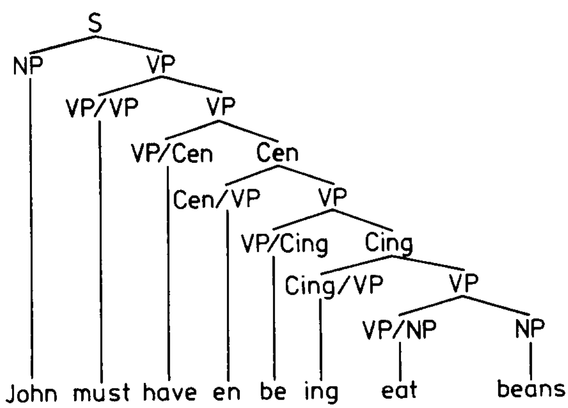
where \$ is a string of the kind defined earlier. The rule can be represented pictorially as:

(14)


Exactly what kind of entity corresponds to incomplete sentences like [He must love]$_{S/NP}$, and how such entities are built, is the subject of the next section.

2.5. *Forward Partial Combination*

For a right-branching structure, the Forward Combination Rule alone would allow no combination to take place until the rightmost lowest word had been put onto the top of the stack. Consider, for example, the sentence (15) and the structure implied by Jackendoff (1977), among others:

(15) (a) John must have been eating beans
(b)


If there is to be any possibility of constructing a semantic interpretation before *beans* is encountered, then something other than the basic Forward rule must be capable of combining one node with another before the later one is complete. For example, if the categories of *must* and *have* are:

(16) must: VP/VP
have: VP/Cen

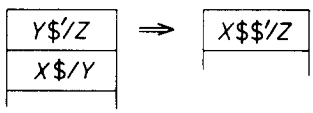
(where Cen is the category of the past participle, as in 15b), it would be natural to wish to combine them into a category VP/Cen corresponding to the semantic interpretation of the compound *must have.* In any case, whether or not this 'partial combination' is desirable a priori, it is a forced move if the processor is to build entities corresponding to incomplete clauses, of the form S$/*X*, mentioned in the account of topicalization and the Backwards Combination rule. Within the present framework, the job can be done with a single further rule schema.

The rule is:

(17) (a) Forward Partial Combination

$$X\$/Y \; Y\$'/Z \Rightarrow X\$\$'/Z$$

(b)



(where $ and $' are two (possibly different, possibly null) strings of the kind defined earlier in Section 2.3 above)[7]. Like the earlier rules, the above is to be thought of as combining the semantic interpretations associated with the two items to which it applies into a single semantic interpretation. Just as the operations of Forward and Backward Combination correspond to the basic semantic operation of *application* of a function to its argument, so Partial Combination corresponds to the equally basic and well-defined semantic operation of *composing* two functions of category $X\$/Y$ and $Y\$'/Z$ into a single function of category $X\$\$'/Z$. Intuitively it is obvious that if you have a function of type VP/VP for *must* and one of type VP/Cen for *have,* then you know everything necessary to build their composition – a function of type VP/Cen for *must have.* The semantics that we have in mind is discussed briefly in the Appendix. There is considerable precedent for rules of this type, both within Categorial Grammar (Ajdukiewicz, 1935; Bar-Hillel, 1960; Geach, 1972), and from one other linguist concerned with processing and direct assembly of semantic representations (Kimball, 1975)[8].

The rule of Partial combination implies that items like [John will have]$_{S/Cen}$ may on occasion be in some sense constituents, corresponding to fully interpreted semantic representations. However, the theory is more orthodox than it might appear in this respect: the classical constituency relations and the associated semantic interpretations are defined in the lexicon. The claim is simply that there are stages in processing where representations of what are in classical terms incomplete constituents are assembled. As stated above the claim appears to follow independently from both the facts of preposing (12) and from the facts about psychological processing and its interaction with semantics. (English does not appear to include a rule of 'Backward Partial Combination', although the treatment of affixes below bears some resemblance to such an operation. However, since the same arguments from semantic coherence and psychological plausibility would apply, we anticipate its occurrence in other languages). It will be noted that the assembly of such 'partial' representations can accomplish much more than the construction of autonomous, uninterpreted syntactic structures. The entities produced by partial combination correspond to fully interpreted functions that may be applied to arguments of the appropriate category, without being modified by a syntactic process such as movement of a constituent into the structure.

2.6. *Affix-verb Compounds and the Introduction of the S-node*

The lexical categories that have been introduced so far have been uncontroversial, and are typically found in other categorial grammars. Our treatment of the affixes, and particularly of Tense, is, however, slightly different from those others.

Verb-affix compounds like *having* and *been* are composed of two parts each of which plays an important role in the combination of adjacent elements. We shall assume that it is words, not morphemes, that are entered to the top of the stack, but that the category of compound words can be determined from the categories of their constituent morphemes. As suggested in (15), following Jackendoff (1977), we take *en* and *ing* as complementizers over VP, i.e. functions from VPs onto complements categorized as Cen and Cing respectively. Since every verb has a category of the form VP$, and such affixes are functions over VPs, the Affix Cancellation rule can initially be stated as:

(18) Affix Cancellation

$$\mathrm{VP\$}.X/\mathrm{VP} \Rightarrow X\$$$

(The dot indicates that the morphemes are parts of a single word.) Thus, *eating* becomes categorized as Cing/NP, and *eaten* as Cen/NP:

(19) (a) $\underset{\mathrm{VP/NP}\ \mathrm{Cing/VP}}{\mathrm{eat.ing}} \Rightarrow \underset{\mathrm{Cing/NP}}{\mathrm{eating}}$

(b) $\underset{\mathrm{VP/NP}\ \mathrm{Cen/VP}}{\mathrm{eat.en}} \Rightarrow \underset{\mathrm{Cen/NP}}{\mathrm{eaten}}$

The Tense affix combines with verb stems in the same way. However, its category is somewhat more complex. We assume Tense to be a function from NP (subjects) onto functions from VPs onto Sentences:

(20) Tense: S/VP/NP

The above category makes Tense the 'glue' that attaches a subject and a predicate. However, since Tense always comes attached to the verb as an affix, it acts by modifying the verb under the Affix Cancellation rule, which must be elaborated so that it will apply to the more complex category of Tense. It is written:

(21) Affix Cancellation

$$\mathrm{VP\$}.X/\mathrm{VP\$'} \Rightarrow X\$\$'$$

(\$ and \$′ are two (possibly null, possibly different) strings of the sort that sort discussed previously). The rule and the categorization for Tense have the effect of making Tense map verb stems (which are all functions onto VP having the form VP\$) onto functions onto S of the form S\$/NP, which first consume a subject NP, then consume the argument(s) originally required by the verb itself. The following compounds receive the following values:

(22) (a) $\underset{\mathrm{VP/NP_o}}{\text{eat}}\ .\ \underset{\mathrm{S/VP/NP_s}}{\text{ed}} \Rightarrow \underset{\mathrm{S/NP_o/NP_s}}{\text{ate}}$

(b) $\underset{\mathrm{VP/VP}}{\text{will}}\ .\ \underset{\mathrm{S/VP/NP_s}}{\text{s}} \Rightarrow \underset{\mathrm{S/VP/NP_s}}{\text{will}}$

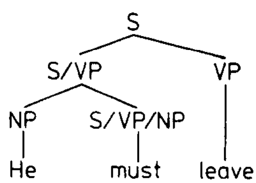
(c) $\underset{\mathrm{VP/Cen}}{\text{have}}\ .\ \underset{\mathrm{S/VP/NP_s}}{\text{ed}} \Rightarrow \underset{\mathrm{S/Cen/NP_s}}{\text{had}}$

(d) $\underset{\mathrm{VP/PP/NP_o}}{\text{put}}\ .\ \underset{\mathrm{S/VP/NP_s}}{\text{ed}} \Rightarrow \underset{\mathrm{S/PP/NP_o/NP_s}}{\text{put}}$

(The subcripts identifying NP's as either object or subject are included only for the reader's convenience: the semantic representation of the affix and verb stem, together with the Affix Cancellation rule will automatically associate NP arguments with the appropriate (surface[9]) function).

The account of tense offered here induces such highly unorthodox analyses as the following:

(23)



A related analysis has recently been proposed by Schmerling (1980) on independent syntactic grounds. However, we offer no further justification for this startling analysis at this point, except that it is the one that works.

The four rules are presented together in their final form in the

Appendix, together with some further remarks on the semantics of Forward Partial Combination.

2.7. *Summary of the Model*

The possibility of any two items combining (in whatever order), and the result of their combination, is determined solely by their categories, which in turn simply reflect their semantics. The categorial lexicon can thus be thought of as defining a grammar or class of grammars which is 'free' with respect to the linear order of functions and arguments. To that extent, it resembles an order-free base grammar. It is the four combination rules, which are equivalent to production rule schemata, that have the role of determining what linear order(s) two combinable items may occur in. Other languages may have (somewhat) different lexicons and/or different restrictions on rule schemata drawn from the same restricted class. (For example, (24) suggests that some more general form of backward combination occurs in German VP.)

(24) ... dass er Eier gegessen haben muss
... that he eggs eaten have must
'... that he must have eaten eggs'

The rules by themselves are rules of grammar in the usual sense of the term. But they can, with the addition of a single Push Down Stack, be thought of as specifying a Non-Deterministic Push Down Automaton (NDPDA), when taken in conjunction with a trivial control mechanism which can be informally stated as follows:

(25) Until the string is empty and no rule matches the topmost items on the stack, either (a) apply a rule to the topmost items and replace them with the result, or (b) put the next word on top of the stack.

The above is all that is needed to convert the 'competence' grammar into a model of the steps a human listener performs while undertaking a given parse. (It ignores, as ever, the separate question of how a listener might choose *which* parse(s) to pursue). Any base generative grammar can be embodied in such an automaton. What is interesting about this one is that, because of the Partial Combination rule, it is immediately compatible with the direct word-by-word assembly of a semantic interpretation in a single left-to-right parse, even for incomplete right-branching constituents, without the mediation of an autonomous syntactic representation.

3. Basic sentence constructions

The simple model outlined in the preceding section, together with lexical specifications of the kind introduced there, will accept a wide variety of English main clause constructions and will refuse to accept virtually all the ungrammatical ones. In the following sections, the possible permutations of a simple clause containing a subject an auxilliary and a transitive VP are first examined exhaustively, and then further main clause constructions are considered.

3.1. '*He must love her*'

There are twenty-four permutations of the words *He, must, love,* and *her,* only a few of which are grammatical. In order to demonstrate the model, it will be useful to consider all twenty-four. The categories of *he* and *her* are assumed to be NP, *love* is VP/NP, and *must* is VP/VP.S/VP/NP (a function from VPs onto VPs, combined with Tense), which reduces by Affix Cancellation to S/VP/NP. The examples are presented, for clarity, in terms of the automaton rather than the grammar rules alone. The working of the automaton will be illustrated using the convention that two items which combine will be underlined with the result written underneath the line. The line will be indexed with the initials of the rule that applied. Its result can of course take part in a further combination: thus the sequence of underlinings down the page represents the successive states of the automaton. (Affix Cancellation will not be shown). Consider, first, the 'canonical' form of the sentence:

(26) (a) He must love her

NP S/VP/NP

———————— B

S/VP

Once *He* and *must* have been put on the stack, Backwards Combination (13) applies and leaves [He must]$_{S/VP}$ on top of the stack. When the main verb *love* is entered, the conditions are met for Forward Partial Combination:

(26) (a) He must love her

S/VP VP/NP

———————— FP

S/NP

which leaves [He must love]$_{S/NP}$ on top of the stack. Finally, the object *her* is entered and Forward Combination applies:

```
(26) (a) He must love   her
         ------------   ---
            S/NP         NP
         ------------------F
                S
```

The pass completes with the symbol S on top of the stack, which is the necessary and sufficient condition for a sentence to be grammatical according to the model.[10]

The analysis of the Question form is very similar

```
(26) (b) Must     he   love    her
         -------  --   -----   ---
         S/VP/NP  NP   VP/NP   NP
         -----------F
            S/VP
             --------------------FP
                    S/NP
                    ------------F
                         S
```

The process is the same as for the 'canonical' (26a), except that *must* and *he* are combined by the Forward rather than the Backward rule. Indeed, it is clear that Subject and tensed verb can occur in either order, so long as they are adjacent.

```
(26) (c) Her  he must  love    ('Topicalisation')
         ---  -------  -----
         NP    S/VP    VP/NP
              ---------------FP
                   S/NP
         ---------------B
               S

     (d) Who(m)  must he  love?  ('Wh-question')
         ------  -------  -----
           NP     S/VP    VP/NP
                 ---------------FP
                      S/NP
         -----------------------B
                 S
```

(As it stands, of course, the model does not explain why (26d) is possible only with a Wh-element, as opposed to **Her must he love*, which is just comprehensible but archaic. Neither does it explain why the subjects of

(26c, e) must be 'given' anaphors such as pronouns, rather than 'new' NPs. We return to these questions in Section 4.1.)

The model handles the one remaining grammatical permutation as follows:

(26) (e) Love her he must ('VP Preposing')

VP NP S/VP/NP

S/VP B

S B

It allows just one further construction, and no more: the following is not grammatical, but is accepted:

(26) (f) *Love her must he

VP S/VP/NP NP

S/VP F

S B

It should be obvious that no model of this kind which will accept (26d, e) – that is which will allow VP Preposing and the Object Wh-Question, – will rule out (26f). One might argue that the unacceptability of the above is parallel to the unacceptability of **Her must he love*. That is, preposing with inversion is only grammatical in English when the preposed entity is a Wh-item. The unacceptability of (26f) follows from the fact that there is no Wh-form for VPs in English[11].

Not one of the remaining eighteen permutations of the four words is accepted by the rules. The reason is that the form of the rules places very powerful constraints on possible rearrangements, which appear to correspond directly to the grammatical possibilities. The most important constraint is the following direct consequence of the fact that the combination rules can operate only on the top two items of the stack, and that the same stack is used both for 'movement' and structure building:

(27) The Adjacency Corollary.

The rules are unable to combine two items that are separated on the stack by a third, unless the intervening item can first be combined with one or the other of them.

(This is a corollary of the model, not an additional assumption).

Because Tense has the category S/VP/NP, it follows that after it has been combined with its verb stem by Affix Cancellation, the first item to combine with the S-node must be the Subject. Because of the corollary (27) it follows that neither the verb phrase nor any component of it can intervene between Subject and tensed verb in an English clause. Among the remaining eighteen permutations, the following twelve are ruled out for this reason. (In some cases there are other reasons as well.)

(26) (g) *He *love her* must

(h) *He *love* must her

(i) *He *her* must love

(j) *He *her love* must

(k) *Must *her love* he

(l) *Must *love her* he

(m) *Must *her* he love

(n) *Must *love* he her

(o) *Her he *love* must

(p) *Her must *love* he

(q) *Love must *her* he

(r) *Love he *her* must

In fact virtually the only way that any material can ever intervene between the Subject and the tensed verb of an English clause under the adjacency corollary is when it constitutes a function of a category that allows it to combine with the tensed verb first, by Forward Partial Combination. We assume adverbials like *obviously* and *frequently* to bear such a category in sentences like *She frequently visits her mother.* The intervening material in Subject extractions like *Who [do you think] took the money*? is shown in Section 5.4 below to constitute a function over tensed S[12].

Another consequence of the rules and the form of the categories is that verbal complements must always follow the verb in question, unless the verb has already been absorbed into the S-node by partial combination, and the complement is topicalised (26c, d, e). The reason is that the Backwards rule applies only to functions of the form S$ which yield an S. Furthermore, the Backwards Combination rule permits only *complete* constituents to be preposed and there is no 'Backwards Partial Combination' rule in English.

(26) (s) *Her love he must

NP VP/NP S/VP B

———— *B

———— *

(t) *Her love must he
(u) *He must her love
(v) *Must he her love

For the same reason, the main verb cannot be preposed:

(26) (w) *Love he must her

VP/NP S/VP B NP

———— *

(x) *Love must he her

3.2. *Generality of Restrictions on English main clauses*
We shall not examine all English main clauses in such exhaustive detail. However, they are similarly constrained by the restrictions inherent in the model, as follows.

3.2.1. *Subject and tensed verb must be contiguous.* As already noted, given the rules and the categorization of Tense as S/VP/NP, tensed verb and Subject may occur in either order but cannot be separated by material from the VP, by the Adjacency Corollary (27). Consider, for example, the position of directional adverbs:

(28) (a) Jones came along ('Canonical Form')

NP S/PP/NP PP

———— B
S/PP

———— F
S

(b) Along came Jones ('Directional Adv. Preposing')

PP S/PP/NP NP

———— F
S/PP

———— B
S

(c) Along Jones came ('Adverb Topicalisation')

PP NP S/PP/NP

———————— B
S/PP

———————————— B
S

(d) *Jones along came

NP PP S/PP/NP

(e) *Came along Jones

S/PP/NP PP NP

(f) *Came Jones along

S/PP/NP NP PP

———————— F
S/PP

———————— F
S

The ungrammatical strings (d) and (e) are excluded by the category of Tense and the Adjacency Corollary.

The last permutation, **Came Jones along*, is wrongly accepted by the model, because the Adjacency corollary is obeyed. This overgeneralisation has nothing to do with the position of the PP. It is rather due to a more general problem concerning the interaction of Tense and the arguments of verbs. The model also predicts that the following non-grammatical results of Subject-Tense inversion should be acceptable:

(29) (a) *Slept John?
(b) *Walked he up the street?
(c) *What ate he?
(c) *Apples ate he
(e) *Stands Scotland where it did?

Such strings have provided the motivation for a Do-Support/Deletion transformation[13]. We return to the question of how to exclude this and the earlier overgeneralisations in a later section, but note that it is at least encouraging that the overgeneralities are all constructions which are found in such nearly-related languages as Dutch and German, as well as in antique forms of English, and might therefore be expected to be governed by minor rules.

A striking example of the requirement that Subject and tensed verb should not be separated by any part of the VP is afforded by the unacceptability of sentences where a verb, itself in 'canonical position' with respect to the tensed verb, is interposed between the two:

(30) (a) Into the garden Maud was walking

PP NP S/Cing/NP Cing/NP

S/Cing B

S/PP FP

S B

(b) *Into the garden was walking Maud

PP S/Cing/NP Cing/PP NP

In stating the transformation Directional Adverb Preposing, Emonds (1976, 38) had to make special reference to the restriction, exemplified above, that it can only apply where there is no auxilliary verb. But this is not an isolated constraint. No such material can ever intervene between Subject and tensed verb, as can be seen from the following:

(31) (a) She might have kissed him
(b) Might she have kissed him
(c) Who might she have kissed
(d) Him she might have kissed
(e) *Might have she kissed him
(f) *Might him she have kissed
(g) *Who she have might kissed
(h) *Who might have she kissed, etc.

3.2.2. *Restrictions on the position of verbs* The other restriction that was pointed out in Section 3.1 followed from the restricted nature of backward combination in English. Unless the verb's complement has been preposed to sentence initial position, it must immediately follow the verb. This observation generalizes to other constructions.

(32) (a) *She must eaten apples have

S/VP Cen VP/Cen

(b) *She has standing on the corner been

S/Cen Cing Cen/Cing

In neither case can the derivation proceed any further, as the last two items (*have* and *been*) cannot combine with their complements, nor can the complements combine with the S-node. The model permits the fronting of participial phrases as instances of topicalization, analogous to the fronting of Object NP (26c) or an Adverb (28c)[14]:

(33) (a) (She said she would be standing at the bus-stop, and)

```
Standing at the bus-stop   she was
------------------------   -------
          Cing              S/Cing
          ------------------------B
                     S
```

(b) (I cant believe I've eaten the whole thing, but)

```
Eaten the whole thing   I have
---------------------   ------
         Cen             S/Cen
         ---------------------B
                  S
```

3.2.3. *Preposing from VP.* In Section 2.3 the category of the verb *put* was assumed to be VP/PP/NP, as in:

(34)

```
   Put       the frog   on the table
---------    --------   ------------
VP/PP/NP        NP           PP
--------------------F
       VP/PP
       ------------------------------F
                    VP
```

The combination of *put* with Tense by Affix Cancellation yields S/PP/NP/NP (22d). The model permits either the NP or the PP to be preposed:

(35) (a)

```
On the table   he       put        the frog
------------   --   -----------    --------
     PP        NP   S/PP/NP/NP        NP
               -----------------B
                   S/PP/NP
                    ---------------------F
                              S/PP
     ---------------------------------B
                    S
```

(b) This frog he put on the table

```
This frog   he put     on the table
---------   --------   ------------
   NP       S/PP/NP         PP
---------------------B
        S/PP
         ----------------------F
                   S
```

The model also allows fronting from within a PP ('Preposition Stranding', a topic which will be further discussed below):

(36) This table he put the frog on

```
This table   he put    the frog   on
----------   -------   --------   -----
    NP       S/PP/NP      NP      PP/NP
             ---------------F
                  S/PP
                   ---------------------FP
                           S/NP
----------------------------------B
                S
```

Finally, the model rules out any sentence where the PP occurs before the NP Object (except when the PP is topicalized to sentence-initial position)[15]:

(37) *He put on the table the frog

```
*He put    on the table   the frog
-------    ------------   --------
S/PP/NP         PP           NP
```

4. The Left Branch Condition and the Root Transformation Hypothesis

4.1. *Removing Overgeneralizations and Distinguishing Clause Functions*

The account presented above is intended to suggest that the basic facts of English main clause constructions can be made to follow from just four very restricted rule schemata, together with an order-free categorial lexicon. However, there are a number of ways in which the model as it stands overgeneralises, and which might seem to cast doubt upon the claim. Its acceptance of ****her must he love***, ****love her must he***, and of sentences violating DO-support have already been mentioned (cf. Sections 3.1 and 3.2.1). A further class of overgeneralisations concerns sentences involving 'double topicalisations', such as the following, all of which are accepted by the model as it stands:

(38) (a) *On the table a frog she put
(b) *In the garden apples he ate.
(c) *The pink one her I gave.

(The obvious extension to complex sentences with sentential complements that we discuss later would if nothing were done provide an opportunity for even more grotesque multiple topicalisations, like **the pink one her Fred Susan I said thought gave.*) There are two sorts of question that are posed for any theory by such overgeneralisations. First, and most generally, does their exclusion involve any increase in the power of the system? (in particular, does it require negative constraints?) Secondly, how are the necessary restrictions to be explained? As in the discussion of DO-support (cf. Note 13) we shall attempt no more in the present paper than to show that the phenomena can be handled without any extension to the power of the model, without claiming that the proposals offered here are correct or explanatory. One solution to the problem is to provide specific instances of the combination rules, which attach features identifying the derivational history of sentences as being +/− topicalised, +/− inverted, +/−Wh-fronted, and so on. The only effect of the elaboration is to slightly restrict the class of sentences that the model accepts, and to capture the different functional categories of clausal constructions within the grammar. It leaves the class from which the rules are drawn, and the power of the grammar, essentially unchanged.

For example, let us assume that partially complete sentences which have already found their subjects, like [He must]$_{S/VP}$, are distinguished from those like [must]$_{S/VP/NP}$ by a feature +/−subject – a feature that we take to be related to the phenomenon of subject-verb agreement. Let us further assume that sentences which are topicalised are distinguished from those which are not by a feature +/−topic. The lexical category for the tense morpheme will carry the features −subject and −topic, since it has yet to combine with either, and these values will be inherited by the tensed verb via the Affix Cancellation rule. We can then define an instance of the Backward Combination rule as follows:

(39) $$X \; \underset{\left[\begin{smallmatrix}+subject\\-topic\end{smallmatrix}\right]}{S\$/X} \Rightarrow \underset{\left[\begin{smallmatrix}+subject\\+topic\end{smallmatrix}\right]}{S\$}$$

(39) expresses the fact that topicalised sentences differ in functional category from non-topicalised ones. It also expresses the fact that sentences with more than one topic are ungrammatical, for the rule

applies only to S nodes which have not previously been marked by the same rule as +topic. The rule therefore excludes all of the strings (38).

Assuming that Auxilliary verbs are distinguished as such in the lexicon, the DO-support phenomena can be captured in a similar way, by replacing the single Forward Combination schema by a number of instances, one of which only allows a tensed verb to find its subject by Forward Combination when it is marked as an Auxilliary. This apparatus could also be extended to include features identifying Yes/No and Wh-questions. Similar instances of the general rules could then be provided to rule out main clauses like **Where he saw a tree* and **Over there did he see a tree*, and complements like **apples I know eats Mary*. Such a modification is equivalent to introducing (for example) a rule schema specifying constituent order for the relative clause into a PS grammar (cf. Gazdar, 1981a). Nothing in the account that follows hinges upon such details. We shall therefore pursue the matter no further, but simply assume that some such feature system could be devised to exclude these overgeneralisations of the basic grammar, without any extension to its intrinsic power, or any loss of the generalisations about the form of those specific rules that the basic theory captures.

4.2. *The Left Branch Condition*

Ungrammatical strings like **love he must her* (26w) were ruled out by the grammar because the main verb is separated from its complement and because backwards combination is restricted to combining functions with arguments of the form Y rather than $Y\$$. This same absence of a 'Backwards Partial Combination' rule in English also rules out the following sentences, which are of a kind which has forced the introduction of constraints like the A-over-A Principle (Chomsky, 1964), the Left Branch Condition (Ross, 1967) and the Relativised A-over-A principle (Bresnan, 1976).

(40) (a) *Whose did you peel grapes

*Whose	did you peel	grapes
NP/N	S/NP	N

(b) *How strong is she a woman

*How strong	is she	a woman
NP/NP	S/NP	NP

On the assumption that *whose* in *whose grapes* is NP/N, any derivation of (40a) blocks because *whose* is not adjacent to *grapes*, and because Backward Combination cannot pick up incomplete objects like NP/N. On the assumption that the category of *How strong* in *How strong a woman* is NP/NP, (40b) blocks in the same way.

4.3. *The Root Transformation Hypothesis*

A number of similar phenomena that Emonds (1976) adduces in support of his two major constraints on Root Transformations are also accounted for by the model as it stands. For example, the following ungrammatical string was ruled out by Emonds on the grounds that it would involve two Root Transformations, VP Preposing and Topicalization, applying in a single clause.

(41) (a) *Apples eat I may

$\overline{\text{NP}}$ $\overline{\text{VP/NP}}$ $\overline{\text{S/VP}}$

As shown in example (26s), such strings are ruled out because verbs must combine with their complements by Forward Combination, and because [I may]$_{S/VP}$ can only backward-combine with a complete VP. The following instances where two Root Transformations have applied, are ruled out for exactly the same reasons:

(41) (b) VP Preposing and Directional Adverb Preposing:
*Into the house run they did
(c) Topicalization and Adverb Preposing:
*The garden in we found the apples
(d) Topicalization and Directional Adverb Preposing:
*This house into they ran

(The sentences in (38) might also be considered as instances where two Root Transformations have applied. They were ruled out by our stipulation that sentences can have only one topic, expressed in the restricted version of the Backward Combination rule (39).)

Emonds' second major constraint on Root Transformations was that they could not apply in subordinate clauses. Without going into the question of how relative clauses attach to their matrices, it is simplest to assume that English subordinate clauses are constructed by the same rules as main clauses. Assuming that relativisation is another instance of topicalization, the following non-sentences which in part motivated Emonds' constraint are ruled out for precisely the same reasons as their main clause analogues (38, 41).

(42) (a) *This is the table on which the frog he put
(b) *This is the pink one which her I gave
(43) (a) *These are the apples which eat I may
(b) *This is the house into which run they did
(c) *This is the garden which in we found the apples
(d) *This is the house which into they ran

However, the following sentence, which appears to allow Directional Adverb Preposing in a subordinate clause, contradicts Emonds' hypothesis, but is correctly predicted to be grammatical by the present model:

(44) This is the rock from which springs the Ganges

Emonds' hypothesis also excludes cases where the Root Transformation has applied in a subordinate clause that is not a relative, but is for example a verbal complement:

(45) *He realised did John come for dinner.

It seems reasonable to suppose that the anomaly arises because of the subcategorisation of the verb *realise* and the functional category of the subordinate sentence, both of which can be expressed in the feature system outlined in Section 4.1.

5. Extraction phenomena

5.1. *Noun-phrase Constraints*

In Section 2 it was argued on both linguistic and psychological grounds that incomplete entities like [He must love]$_{S/NP}$ must have a place in the grammar. However, no similar argument could be advanced for the reality of entities like *[He must love the]$_{S/N}$. English does not allow preposings like

(46) *Woman he must love the.

As it stands the Forward Partial Combination Rule will allow the formation of such a construction by partially combining [He must love]$_{S/NP}$ and [the]$_{NP/N}$. What is needed is a way of restricting the Forward Partial rule to the combination of 'verb-like' argument functions.

In fact, the device of restricting rules to apply only to certain sorts of category has been implicit all along, in the proviso that Backwards Combination apply only with functions yielding S, and in the restriction of Affix Cancellation to affixed verbs. The required restriction on the Forward Partial Combination rule differs only in that it refers to a whole class of node-types, rather than to a single type. In order to impose this restriction we adopt the following tenets of the *X*-bar hypothesis (Chomsky, 1970; Jackendoff, 1977):

(47) (a) All the basic category symbols, both terminal and non-terminal, are considered to be associated with a bundle of grammatical features.

(b) The major categories Nominal, Verbal, Adjectival and Prepositional are considered to be associated with specific combinations of the major features +/− N and +/− V.

(c) All constituents are considered to be of the form

X^n = (Specifier) X^{n-1} Complement*

– where X is a major category, and X^{n-1} is termed the 'head' of the constituent X^n.

Under this notation, for example, the nounphrase might be regarded as having the determiner as specifier, the noun as head, and a relative clause or prepositional phrase as an (optional) complement.

It will be noted that this notation draws a distinction between two varieties of function category which has not hitherto been made in the present theory, in which both heads and specifiers are merely functions of the form $X\$/Y$. The difference between them is expressed in (47c) above, which makes it clear that while 'Head' functions, such as nouns and verbs, are already associated with major features, 'Specifier' functions are not. On the contrary, when specifiers combine with their head they yield an entity which bears the features of that head. We therefore make the following additional stipulation, which is already implicit in the X-bar system:

(48) No combination, and in particular no partial combination, can apply where the argument term does not bear major features.

It follows that there can occur no partial combination of specifiers which have not yet found their heads, and therefore that specifiers (a class which, with Jackendoff (1977), we take to include determiners, complementisers and affixes) cannot 'strand', unlike heads such as prepositions and verbs. Thus, for example, [He must love]$_{S/NP}$ and [the]$_{NP/N}$ cannot combine because the determiner is a specifier and bears no major features. A slightly more stringent restriction upon Forward Partial Combination will enable us to capture several further constraints on extraction.

5.2. *The NP Constraint*

Besides the general constraint that functions can only combine with arguments that bear major features, we can further specify *which* major features a particular rule allows, just as we restricted the functions to which the Backwards Combination rule could apply. The following restriction upon the Forward Partial Combination rule will rule out sentences that would violate the NP constraint of Horn (1974) and Bach and Horn (1976)[16]

(49) Forward Partial Combination

$$X\$/Y \quad \underset{[\ldots,-N,\ldots]}{Y\$/Z} \Rightarrow X\$\$'/Z$$

Consider, for example:

(50) *This man I burned a book about.

The construction would be possible if *I burned a book about* could be formed into an incomplete entity of type S/NP. There are two possible views of the internal structure of Complex NPs like *a book about this man*, but on neither view will the revised form of Partial Combination allow the extraction. If the NP is regarded as $[[\text{a}]_{NP/N}$ $[\text{book about this man}]_N]_{NP}$ then it is blocked because $[\text{I burned}]_{S/NP}$ cannot combine with $[\text{a}]_{NP/N}$. If the NP is $[[\text{a book}]_{NP/PP}$ $[\text{about this man}]_{PP}]_{NP}$, then it is blocked because $[\text{I burned}]_{S/NP}$ cannot partially combine with $[\text{a book}]_{NP/PP}$. Both combinations are excluded because NP naturally does not bear the feature −N. (Following Bach and Horn, 1976, we take it that in one reading of *I wrote a book about this man*, the PP is dominated by VP, not by NP. That is to say that in terms of the present theory, one lexical entry for *write* identifies it as VP/PP/NP. Hence the acceptability of *This man I wrote a book about*, analogous to (36).)

Extraction from other complex NPs is blocked for just the same reason as in (50): they are not dominated by a node bearing the feature −N. For example:

(51) (a)

*This man	I met	a	girl	who knew
NP	S/NP	NP/N	N/S	S/NP

(b)

*This man	I met	a girl	who knew
NP	S/NP	NP/S	S/NP

5.3. *Preposition Stranding*

If Prepositional Phrases are included in the class of −N constituents and Prepositions are their heads, as is implied by both Chomsky (1970) and Jackendoff (1977), then the grammar also expresses the fact that extraction from within PP in post verbal position is allowed in English[17]

(52) This table he put the frog on (cf. example 36).

It is worth pointing out that the model does not require the node PP to be characterised as extractable per se. It has already been shown how other general features of the model for English rule out preposition stranding in PPs that have already been fronted (41c, 43c). The account appears to have a descriptive advantage over theories that label PP as a bounding node in English to explain the ill-formedness of (41c, 43c), but which must then invoke an 'escape zone' hypothesis to explain the acceptability of preposition stranding in post verbal position. (Cf. Koster, 1980).

The suggestions advanced above are broadly related to the idea of a 'projection path' (Koster, 1978). As has been noted by Fodor (1980), the Forward Partial rule determines how right-branching structures are built up, while the restrictions upon the types of categories to which it can apply in a given language does the work of Subjacency constraints and principles of Government (cf. Koster, 1980), in determining which nodes are on the projection path.

5.4. *Extraction from Sentential Complements*

On the assumptions (a) that one lexical category of the verb *believe* is VP/S, and (b) that S nodes, like all verbal nodes, carry the feature – N, as implied by the *X*-bar hypothesis, then it follows that extraction will be possible from both subject and object positions in simple sentential complements, as follows:

```
(53) (a) Who(m)  do you think  he loves?
         ------  -----------  --------
           NP        S/S        S/NP
                    ------------------ FP
                          S/NP
          ---------------------------- B
                    S

     (b) Who  do you think   loves    him?
         ---  -----------  -------  ---
         NP       S/S      S/NP/NP  NP
                 ---------------------- FP
                        S/NP/NP
         ------------------------------ B
                   S/NP
                 ------------------------------ F
                             S
```

On the assumptions that the category of *that* identifies it as a function which we might write as Cthat/S, which maps sentences onto some sort of complement Cthat, and that *believe* is also categorised as combining with complements of category Cthat, i.e. as VP/Cthat, then extraction from object positions in *that* complements can proceed more or less as in 53a):

```
(54)   Who(m)   do you think   that     he loves?
       ------   ------------   ------   --------
         NP       S/Cthat      Cthat/S    S/NP
                               ----------------- FP
                                   Cthat/NP
                  -------------------------- FP
                              S/NP
       ------------------------------- B
                     S
```

(*Do you think* may combine with *that he loves*, just as with *he loves*. But, given the stipulation (48) and the form of the Partial Combination rule (49), it cannot combine with the specifier *that* until the latter has 'inherited' some features from its head *he loves*, and in particular the feature −N which all verbal categories including S bear.)

It remains to be explained why extraction from subject position in *that* complements is *not* allowed in English (Chomsky and Lasnik, 1977).

```
(55)   *Who   do you think   that      loves      him
       ----   ------------   -------   --------   ---
        NP      S/Cthat      Cthat/S   S/NP/NP    NP
```

In Section 4, certain features were introduced to express the different functional meanings of the various clause constructions. One of these was a feature +/− subject, which distinguished S-nodes that had already found their subjects from those that had not. A simple solution to the above problem is to further stipulate that the word *that* is categorised as a function over sentences which bear the feature +subject – that is as Cthat/S[+subject]. Given this categorisation, *that* cannot combine with $[\text{loves}]_{S/NP/NP}$, because the latter is still −subject. The derivation can therefore proceed no further.

The proposal outlined above is once again intended mainly to show that the present system can be made to obey the constraints upon 'movement' that have been identified by transformationalists, without the inclusion of any negatively stated rules. Whether or not this particular account will stand the test of time is another question, and depends of course on whether further reasons can be produced to justify the mysterious feature +/− subject, and the categorisation of the morpheme *that* with respect to it. It is, however, suggestive that in some heavily inflected languages, like Italian and Spanish, where the finite verb inflection is very similar to a personal pronoun, the equivalent of (55) *is* grammatical (Perlmutter, 1968; Chomsky and Lasnik, 1977, p. 450f). In such languages one might suspect that the finite verb inflection already bears the feature +subject.

6. Conclusions

There is no doubt that the model presented here is incomplete. Many important categories, particularly negation and the adverbials, have been entirely ignored, and the treatment of Tense and the affixes is certainly inadequate. It also remains to be seen how the many constructions that have been ignored here are to be accommodated within the framework that has been outlined. However, the fact that a standard categorial lexicon, plus the four rule schemata, seems to come close to exhaustively specifying the main clause constructions of English, and also seems to explain a number of major constraints on transformations, encourages us to compare the theory with certain alternatives, and to examine its broader implications.

6.1. *General Linguistic Properties*

In this model, the lexicon has the familiar role of specifying both the categories of words and the entities that they may co-occur with, as in other generative frameworks. However, because it is merely a projection of semantic dependencies, it is free with respect to the left-right order of functions and arguments, and to that extent also constitutes the equivalent of an order-free base. If any order is allowed by the dependencies and the constraint that none of the combination rules operates on other than adjacent entities, then the dependencies will also potentially allow the mirror image order. For any given language, it is the specific form of the combination rules that further determines which of the orders allowed by the lexicon are grammatical in that language.

The model may be contrasted in this respect with other theories in which semantic dependencies are specified not only in the lexicon, but also in syntactic base rules and/or the semantic rules (which Brame (1976, 1978) and Heny (1979) have repeatedly pointed out all look suspiciously alike). Similarly, the present model is to be contrasted with others which divide the task of specifying constituent order between different components of the grammar, such as an orthodox PS base and either a transformational component or a number of metarules which are similarly extrinsic to the grammar and the corresponding semantics and to the processor. While the reallocation of duties may be a 'notational variant' of these other theories, it seems to be one that is on the side of parsimony, and one which effects considerable savings in the number of different varieties of rules the system includes. The clear separation between surface ordering and semantic dependency may also offer the theorist a freer hand to approach linguistic universals. As relational grammarians and others have frequently pointed out, these are notably resistant to analysis with an ordered base (cf. Cole and Sadock, 1977; Venneman, 1973).

6.2. *Constraints on 'Movement'*

A grammar ought to need no more than the addition of a mechanism for ambiguity resolution in order to turn it into a theory of performance. Given this requirement, negative rules to constrain movement have a very problematic status indeed, as was pointed out earlier. Their absence from the present theory is therefore a desirable feature.

There are also advantages from a linguistic point of view in handling the constraint phenomena as the present theory does. All extraction constraints are reduced to a single underlying mechanism. Extraction is possible if and only if a structure can be 'penetrated' by Partial Combination. Thus the Left Branch Condition and many constraints upon Root constructions hold in English because it includes no 'Backwards Partial Combination' rule. Similarly, the noun-phrase constraints apply because the Forward Partial Combination rule cannot penetrate noun-phrases, so that there is no way that a 'bridge' (Shir, 1977) can be built between the moved item and the extraction site. The absence of negative rules may also have implications for language learning: but this issue is too controversial to pursue here (cf. Levelt, 1979).

The device of partial combination plays a somewhat similar role in the present theory to the metarule which introduces 'derived PS rules' in the Base-generative theory of Gazdar (1981a), in that it is the means by which certain entities such as S/X, a sentence with a 'gap', 'hole', 'trace' or 'empty node', are given grammatical status. There is some similarity between our freedom to exclude from English grammar a rule of 'Backward Partial Combination', (parallel to (17) but with the order of the 'consuming' and 'consumed' functions reversed), and Gazdar's (1981a) freedom to exclude certain derived rules under the 'Generalised Left Branch Condition', although the proposals differ in their further implications[18]. A further similarity is that the restrictions which were imposed upon Forward Partial combination in order to capture the Noun-phrase Constraint are identical to the parallel restrictions on induction of derived rules in the Gazdar framework. However, (Gazdar, 1981a; Fodor, 1981) the rule is not used to introduce new rules into the base grammar defining constituents with holes or gaps. The categorial grammar already does that. All that partial combination does is to compose 'hole categories' (that is, functions), and assemble the corresponding translation as described in the Appendix. This is more than a notational variant. Because the categorial grammar allows more than one gap in a category (for example, the verb *put* has the category VP/PP/NP) the existence in English of constructions that include two gaps, like (2a), *This wood is too rough for these nails to be easy for us to hammer into*, and of some of the double extractions that Maling and Zaenen (1981) and Engdahl (1980) have noted in Scandinavian languages, are unproblematic. The context free nature of the categorial component will also ensure that such gaps must nest, rather than cross[19]. Moreover, the apparatus of variable binding that goes with the Lambda Calculus will ensure the correct association of functions and arguments, without recourse to any further 'Storage Mechanisms' or devices like Distinguished Variables.[20]

6.3. *Psychological Properties*

Two features of the model mean that it can be translated easily into a psychologically believable process. First, because there is a one-to-one correspondence between rules of grammar and rules of semantics, it does not require that the processor build autonomous syntactic representations, however temporary, for subsequent semantic interpretation. According to this model, syntax is something that a speaker or hearer *does* in getting from strings to meanings, or vice versa, not something that is built. Of course, many other grammars offer the possibility of non-autonomy in processing. However, because it includes the Partial Combination mechanism as a rule of grammar, the present one maps directly onto a processor which can build semantic interpretations on an almost word-by-word basis, even for incomplete fragments of the right-branching structures which abound in English. The psychological advantages are considerable. Apart from the fact that it allows incomplete sentences to receive an interpretation, it seems likely that it is some such mechanism that underlies semantic and contextual facilitation effects upon ambiguity resolution (Tyler and Marslen-Wilson, 1977) and upon word recognition (Marslen Wilson and Tyler, 1980). For example, a listener possessing a semantic interpretation of 'I'm going to drop ...' or 'He might have ...' will be able to apply the functions in question to putative subsequent words and/or constituents, and thereby more rapidly identify them.

We would suggest, finally, that this sort of mechanism should be given more weight in discussions of ambiguity resolution. Recent accounts have placed greater emphasis upon structural mechanisms (Frazier and Fodor, 1978; Frazier, 1980), lookahead (Marcus, 1977) and rule-ordering (Wanner, 1980) as ways of resolving ambiguities, and upon 'garden path' sentences as evidence. However, as Crain (1980) has pointed out, all garden path effects seem to be eliminable by a suitable choice of semantic or pragmatic context. Since all accounts admit the need for some kind of semantic interaction, it seems likely that systems like the present one which allow the immediate assembly of semantic interpretations will play an important part in psychological theories of ambiguity resolution. (Indeed, it remains unclear how much *more* is needed than such a system.) If so, the present account will in some sense have come full circle from our initial exclusion of local ambiguity questions. However, such considerations will remain logically secondary to the present attempt to explain the apparent vagaries of a significant fragment of English grammar.

Appendix

1. The Grammar

1.1 *A Fragment of Lexicon*

the:	NP/N	ing:	Cing/VP
that:	Cthat/S[+ subject]	Tense:	S/VP/NP
eat:	VP/NP	put:	VP/PP/NP
be:	VP/Cing	think:	VP/Cthat
on:	PP/NP	frog:	N

1.2. *The Combination Rules*

In what follows, X, Y, Z etc. stand for atomic category symbols such as NP, VP and so on, and \$, \$′ etc. stand for members of the set containing the empty string and all strings composed of alternating slashes and atomic category symbols, and beginning with a slash and ending with a category, such as /NP, /PP/NP, /NP/NP/NP and so on. (Thus $X\$$, $X\$'$, $Y\$$ etc. stand for *either* atomic categories *or* functions like S/NP, VP/PP/NP and so on. Similarly $X\$/Y$, $Y\$'/Z$ etc. are functions over the categories Y, Z etc.onto

$X\$$, $Y\$'$ and so on.) The combination rules are written:

Forward Combination:
$X\$/Y\ Y \Rightarrow X\$$
Backward Combination:
$Y\ S\$/Y \Rightarrow S$
Forward Partial Combination:
$X\$/Y\ \underset{[..,-N,..]}{Y\$'/Z} \Rightarrow X\$\$'/Z$
Affix Cancellation:
$\underset{[..,+V,..]}{Y\$}\ .X/Y\$' \Rightarrow X\$\$'$

2. The processor

When provided with a single Push-Down store, or Stack, and the following trivial control mechanism, the above rules specify a Non-deterministic Push-Down Automaton:

> Until the string is empty and no rule matches the topmost items on the stack, *either*: (a) apply a rule to the topmost items and replace them with the result, *or*: (b) put the next word of the string on top of the stack.

3. The semantics of the combination rules

Items of category $X\$/Y$ are functions in every sense of the word, including the sense used in Montague semantics and other systems related to the Lambda Calculus. The semantics of Forward and Backward Combination is simply the application of a function to its argument.

The semantics of Forward Partial Combination is marginally less straightforward. Intuitively, though, it is fairly obvious that if we have a definition of a function that we might call $\text{I-WILL}_{S/VP}$ (the function over VP interpretations which is produced for the partial sentence *I will* by Backward combination), and if we also have a definition of the function $\text{GIVE}_{VP/NP/NP}$ (a function from NP intensions onto a function from NP intensions onto VP interpretations), then we know everything that we need to know in order to define a new function $\text{I-WILL-GIVE}_{S/NP/NP}$. The new function is simply the composition of the old ones, which can be written

$$\text{LAMBDA } x[\text{LAMBDA } y[\text{I-WILL}_{S/VP}\ (\text{GIVE}_{VP/NP/NP}\ (x)\ (y))]]$$

– where x and y are variables which the category of the function identifies as being of the type of NP intensions.

Such a function is a complete semantic interpretation of the partial sentence *I will give*: it can combine directly with an argument without further modification, and in particular without any structural insertion of further material. Moreover, all such functions can be produced automatically by a single rule which maps pairs of component functions onto their composition.

Acknowledgements

We would like to thank William Marslen-Wilson, Stephen Isard, Janet Fodor, and Phil Johnson-Laird for continual moral and conceptual support. They, and many of our colleagues in Sussex, Warwick, Edinburgh, Amherst, Austin, and Nijmegen gave us valuable comments on two earlier versions which were circulated under slightly different titles (Ades and Steedman, 1979; Steedman and Ades, 1981).

The alphabetically first author was supported by a Fellowship from the Royal Society European Science Exchange Programme, held at the Max Planck-Gesellschaft Projektgruppe fur Psycholinguistik, Nijmegen, Netherlands. The research was also supported by a grant from the Social Science Research Council and a Visiting Fellowship from the Sloan Foundation, held at the University of Texas, to the second author. Certain portions of the paper were presented to the autumn meeting of the Linguistic Association of Great Britain, Sussex, September 1979, and the AISB conference on Artificial Intelligence, Amsterdam, March 1980 (Steedman and Ades, 1980).

Reprint requests should be addressed to Steedman, at the Department of Psychology, University of Warwick, Coventry CV4 7AL, England.

Max Planck Institute for Psycholinguistics, Nijmegen
University of Warwick, Coventry

Notes

[1] The tendency to demand nested dependencies is overwhelming. There may be very limited occasions on which apparently crossing dependencies are allowed (see accounts of Dutch infinitival complements by Huybregts, 1976 and of Norwegian relative clauses by Engdahl, 1980). However, these languages still predominantly nest dependencies. There are no languages which *only* allow crossing dependencies, and there are not even any which *predominantly* cross, allowing just a little nesting.

[2] It is often argued that the rules are restricted in this way because Context Free languages can provably all be parsed using resources of time and space proportional to the cube of

the length of sentence or less. But *some* strictly Context Sensitive languages can be processed with similar economy, so their non appearance remains unexplained.

[3] Structure preserving transformations have been studied extensively within the Base Generation framework. For instance Peters (personal communication), Gazdar (1981a, 1981b), Brame (1978) and Bresnan (1978), have all proposed various means of capturing the active-passive relation without transformations. We shall assume that some such approach is feasible and compatible with the model developed in Section 3.

[4] Following Emonds (1976) we isolate and exclude from further consideration those Root Transformations that induce comma intonation and/or involve more than one clause. This class includes Left and Right Dislocations, Parenthetical Formation, and Tag Questions.

[5] The poetic or stylistic constructions of English also seem to obey the NDC, as for example:

(i) [In Xanadu]$_1$ did$_2$ Kubla Khan 0_2
[a stately pleasure dome]$_3$ decree $0_3 0_1$

See Note 1 for some remarks concerning exceptions to the NDC.

[6] There seems to be widespread belief that Woods (1973) has proved the use of such incremental interpretation and evaluation to guide syntactic processing to be less efficient than the non-interactive alternative. In fact, the experiments that Woods performed only show this to be true of one particular processor, the LUNAR project ATN, which had a particularly cumbersome interface with its semantic module. They have no implications for other processors, still less for the psychological one.

[7] The rule schema (17) is stated in its most general form, and almost certainly induces a grammar which is no longer strictly CF (Peters, personal communication) as the categorial grammar taken with the simple combination schemata is, but which is rather some limited generalisation of CF grammar. We are not sure at this point whether we need the fully general form, nor is it clear exactly what generalisation of CF grammar, if any, is involved. As far as the examples used in the present paper go, the more restricted form

(i) $X/Y\,|Y/Z \Rightarrow X/Z$

would do – which does not induce greater than CF power. The necessity for the slightly more general

(ii) $X/Y \quad Y\$/Z \Rightarrow X\$/Z$

for English rests in part on the acceptability of sentences like

```
(iii)   Who    did you    give        the book that you bought for Mary?
        ---    -------    ---------   ----------------------------------
        NP     S/VP       VP/NP/NP                  NP

               ----------------------?FP
                     S/NP/NP
              ---------------------B
                    S/NP
                 --------------------------------------F
                                  S
```

– upon which matter opinions differ. However, work in progress on certain problems in other Germanic languages leads us to believe that they require at least the more general (ii), and we have therefore chosen not to prejudge the issue. It is worth noting in this connection that the rule is very closely related to the process of inducing 'derived PS rules' in the theory of Gazdar (1981a, 1981b). We return to this comparison in the concluding section.

[8] Kimball's motives for advancing such forms of semantic representation were like our own. He wanted to explain how semantic relationships could be elaborated before the end

of right-branching structures was reached. This idea, which he attributed to Wise and Shapiro and called the 'ongoing function hypothesis', was to play a crucial role in permitting semantic guidance of parsing decisions at local ambiguity points (see Section 6.1). Speaking of the string *Tom wanted to ask Susan* . . . , and of the state of the parser after the first two words had been accepted, Kimball wrote (1975, 174): "I conjecture that at this point we have in the semantics a function which we might call (Tom want), that requires an argument".

[9] As for those cases where the surface subject or object appears to take on a different functional role, see Note 3.

[10] It will be noted that there is a second sequence in which the combination rules can apply to accept this sentence. Instead of partially combining [he must]$_{S/VP}$ and [love]$_{VP/NP}$, we could input another word [her]$_{NP}$, combine it with the verb by Forward Combination to yield a complete VP, which could then be combined with [he must]$_{S/VP}$ by a further application of Forward Combination, thus:

```
(i)      He must   love    her
         -------  ------  ---
          S/VP    VP/NP   NP
                  -----------F
                      VP
         --------------------F
                   S
```

The question of *which* analysis a processor pursues is of course a question quite external to the grammar. However the semantic interpretation of the 'partial' sentence [he must love]$_{S/NP}$ produced during the alternative analysis when applied to the object [her]$_{NP}$ results in exactly the same interpretation produced by (i) (see Appendix). There is therefore no ambiguity of sense for this sentence.

[11] We owe this suggestion to Emonds (personal communication). It is possible that *so* is in fact a Pro-VP in the sentence *So must I*, in which case (26f) *is* grammatical as predicted by the model, provided that the VP takes this form.

[12] Again, we ignore the question of parenthetical constructions.

[13] It is possible to accommodate a *Do*-support analysis within the present framework. For example, we could assume, along with transformational treatments, that main verbs are distinguished in the lexicon by some feature, and that tensed main verbs are decomposable into two elements, a tensed modal-like element and a main verb stem. The sentences (29) would then be ruled ungrammatical for exactly the same reason as:

(i) *Must love he her

– that is, the Tense bearing item is not adjacent to the subject. One way of incorporating this analysis would be to assume that when a tensed main verb VP\$.S/VP/NP is encountered, two items VP/VP.S/VP/NP (a modal plus Tense) and VP\$ (the main verb stem) are delivered, and are processed just as if they had occurred in the sentence.

However, we regard this solution to the overacceptances (29) as no more than makeshift, and a mere demonstration that a technical solution is possible within the constraints of the present model. The above solution would also necessitate a different analysis of Directional Adverb Preposing. Another solution is implicit in the feature system proposed in Section 4.1.

[14] The model will also accept the following, which seem unacceptable to us:

(i) ?Been eating beans he has
(ii) ?Have stood on the corner he will

[15] Why such strings are grammatical when the object NP is 'heavy' we do not know.

[16] The range of data covered depends upon the feature composition of the different parts of speech. For example, on the reasonable assumption that adjectives are [..., +N,...] (Chomsky, 1970; Jackendoff, 1977), the Forward Partial Combination rule as stated rules out:

(i) *Angry we made them very

However, since so much doubt surrounds the distribution of features, and even the identity of the features, we regard the present indexing of the Forward Partial Combination rule as little more than illustrative.

[17] This does not commit us to predicting preposition stranding in all languages which extract from within VP. Within other languages, the Partial Combination rule(s) may be indexed by a feature that distinguishes VP and PP, such as +V.

[18] For example, the Generalised Left Branch Constraint has the effect of prohibiting Subject extraction, thus forcing the introduction of a number of metarules which are not required under the present proposal.

[19] In fact, the Forward partial combination rule in the general form in which it is stated in (17) *will* permit a few crossed dependencies under certain very restricted circumstances which do not arise in English. However, the basic point still stands: the Nested Dependency Constraint stems from the CF grammar of which this theory is some limited generalisation.

[20] Such devices (cf. Cooper, 1978; Gazdar, 1981a) are usually considered to be in the domain of semantics rather than syntax. However, to the extent that they augment the corresponding automaton with extra storage devices, they can equally well be regarded as potentially powerful extensions to the syntax itself.

References

Ades, A. E. and Steedman, M. J.: 1979, 'On Word-order', mimeo (University of Warwick).

Ajdukiewicz, K.: 1935, 'Über die syntaktische Konnexitat', *Studia Philosophica* **1**, 1–27. English translation in Storrs McCall (ed.), *Polish Logic* 1920–1939 (Oxford University Press), p. 207–231.

Bach, E.: 1977, 'Comments on a Paper by Chomsky', in P. W. Cullcover, T. Wasow and A. Akmajian (eds.), *Formal Syntax* (Academic Press, New York).

Bach, E. and Horn, G. M.: 1976, 'Remarks on "Conditions on Transformations"', *Linguistic Inquiry* **7**, 265–299.

Bar-Hillel, Y., Gaifman, C., and Shamir, E.: 1960, 'On Categorical and Phrase Structure Grammars', *The Bulletin of the Research Council of Israel* **9F**, 1–16.

Bolliet, L.: 1968, 'Compiler Writing Techniques', in F. Genuys (ed.), *Programming Languages* (Academic Press, New York).

Brame, M. K.: 1976, *Conjectures and Refutations in Syntax* (Elsevier North Holland, New York).

Brame, M.K.: 1978, *Base Generated Syntax* (Noit Amrofer, Seattle).

Bresnan, J.: 1976, 'On the Form and Function of Transformations', *Linguistic Inquiry* **7**, 3–40.

Bresnan, J.: 1978, 'A Realistic Transformational Grammar', in M. Halle, J. Bresnan, and G. Miller (eds.), *Linguistic Structure and Psychological Reality* (MIT Press, Cambridge, Mass).

Chomsky, N.: 1957, *Syntactic Structures* (Mouton, The Hague).

Chomsky, N.: 1964, *Current Issues in Linguistic Theory* (Mouton, The Hague).

Chomsky, N.: 1970, 'Remarks on Nominalisation', in R. Jacobs and P. Rosenbaum (eds.), *Readings in English Transformational Grammar* (Blaidsell, Waltham, Mass.).

Chomsky, N. and Lasnik, H.: 1977, 'Filters and Control', *Linguistic Inquiry* **8**, 425–504.

Cole, P. and Sadock, J. M.: 1977, (eds.), *Syntax and Semantics*, Vol. 8: *Grammatical Relations* (Academic Press, New York).
Cooper, R.: 1978, 'A Fragment of English with Questions and Relative Clauses', mimeo (University of Wisconsin).
Crain, S.: 1980, *Pragmatic Constraints on Sentence Comprehension*, Ph.D. Thesis (University of California, Irvine).
Davies, D. J. M. and Isard, S.: 1972, 'Utterances as Programs', in D. Michie (ed.), *Machine Intelligence* 7 (Edinburgh University Press).
Emonds, J. E.: 1976, *A Transformational Approach to English Syntax*: *Root, Structure Preserving, and Local Transformations* (Academic Press, New York).
Engdahl, E.: 1980, *The Syntax and Semantics of Questions in Swedish*, unpublished doctoral thesis (University of Massachusetts, Amherst).
Fodor, J. D.: 1978, 'Parsing Strategies and Constraints on Transformations', *Linguistic Inquiry* **9**, 427–473.
Fodor, J. D.: 1980, 'Parsing, Constraints, and the Freedom of Expression', unpublished ms. (University of Connecticut).
Fodor, J. D.: 1981, 'Phrase Structure Parsing and the Island Constraints', paper to Conference on Human Parsing Mechanisms, Austin, Texas, March 1981.
Frazier, L.: 1979, *On Comprehending Sentences*: *Syntactic Parsing Strategies*, Ph.D. Dissertation (University of Connecticut). Indiana University Linguistics club (1979).
Frazier, L. and Fodor, J. D.: 1978, 'The Sausage Machine: A New Two-stage Parsing Model', *Cognition* **6**, 291–325.
Gazdar, G.: 1981a, 'Unbounded Dependencies and Constituent Structure', *Linguistic Inquiry* **12**, 155–184.
Gazdar, G.: 1981b, 'Phrase Structure Grammar', in G. K. Pullum and P. Jacobson (eds.), *On the Nature of Syntactic Representation* (Croom Helm, London).
Geach, P. T.: 1972, 'A Program for Syntax', in D. Davidson and G. Harman (eds.), *Semantics of Natural Language* (Reidel, Dordrecht).
Heny, F.: 1979, 'Representing Significant Generalisations in lexical Grammar', mimeo (Rijksuniversiteit Groningen).
Horn, G. M.: 1974, *The Noun Phrase Constraint* (Linguistics Club, Indiana University).
Huybregts, R.: 1976, 'Overlapping Dependencies in Dutch', *Utrecht Working Papers in Linguistics* **1**, 24–65.
Jackendoff, R.: 1977, *X-bar Syntax*: *A Study of Phrase Structure* (MIT Press, Cambridge, Mass.).
Johnson-Laird, P. N.: 1977, 'Procedural Semantics', *Cognition* **5**, 189–214.
Kaplan, R.: 1973, 'A Multiprocessing Approach to Natural Language', in *Proceedings of the First National Computer Conference.*
Kimball, J.: 1975, 'Predictive Analysis and Over-the-top Parsing', in J. Kimball (ed.), *Syntax and Semantics*, Vol. 4 (Academic Press, New York).
Koster, J.: 1978, 'Conditions, Empty Nodes and Markedness', *Linguistic Inquiry* **9**, 551–593.
Koster, J.: 1980, 'Proximates, Locals, and Dependents', in J. Koster and R. May (eds.), *Levels of Syntactic Representation* (Foris, Dordrecht).
Levelt, W. J. M.: 1978, 'On learnability, a Reply to Lasnik and Chomsky', mimeo (Nijmegen).
Lewis, D.: 1971, 'General Semantics', *Synthese* **22**, 18–67.
Lyons, J.: 1968, *Introduction to Theoretical Linguistics* (Cambridge University Press).
Maling, J. and Zaenen, A.: 1980, 'A PSR Treatment of Unbounded Dependencies in Scandinavian languages', in G. Pullum and P. Jacobsen (eds.), *The Nature of Syntactic Representation* (Reidel, Dordrecht).
Marcus, M. P.: 1977, *A Theory of Syntactic Recognition for Natural Language* (MIT Press, Cambridge, 1980).

Marslen-Wilson, W. D.: 1973, 'Linguistic Structure and Speech Shadowing at Very Short latencies', *Nature* (London) **244**, 522–523.

Marslen-Wilson, W. D. and Tyler, L. K.: 1980, 'The Temporal Structure of Spoken language Understanding: The Perception of Sentences and Words in Sentences', *Cognition* **8**, 1–74.

Peters, S.: 1980, 'Definitions of Linked Tree Grammars', Lecture notes (University of Texas, Austin).

Perlmutter, D. M.: 1968, *Deep and Surface Constraints in Syntax* (MIT, unpublished Doctoral dissertation).

Ross, J. R.: 1967, *Constraints on Variables in Syntax* (MIT, unpublished Doctoral dissertation).

Schmerling, S.: 1980, 'A New Theory of English Auxilliaries', mimeo (University of Texas, Austin).

Shir, N. E.: 1977, *On the Nature of Island Constraints* (Indiana University Linguistics Club).

Steedman, M. J. and Ades, A. E.: 1980, 'An Algorithmic Approach to English Main Clause Constructions', in S. Hardy (ed.), *Proceedings of the* 1980 *AISB Conference* (AISB, University of Sussex).

Steedman, M. J. and Ades, A. E.: 1981, 'Son of Word-order', Paper to Sloan Workshop on Processing Long-distance Dependencies, University of Massachusetts, Amherst, January 1981, Mimeo, University of Texas, Austin.

Steedman, M. J. and Johnson-Laird, P. N.: 1978, 'A Programmatic Theory of Linguistic Performance', in P. Smith and R. Campbell (eds.), *Advances in the Psychology of Language – Formal and Experimental Approaches* (Plenum, New York).

Tyler, L. K. and Marslen-Wilson, W. D.: 1977, 'The On-line Effects of Semantic Context on Syntactic Processing', *J. Verbal Learning and Verbal Behavior* **16**, 683–692.

Thorne, J., Bratley, P., and Dewar, H.: 1968, 'The Syntactic Analysis of English by Machine', by D. Michie (ed.), *Machine Intelligence* 3 (American Elsevier, New York).

Venneman, T.: 1973, 'Explanation in Syntax', in J. Kimball (ed.), *Syntax and Semantics* II (Seminar Press, New York).

Wanner, E.: 1980, 'The ATN and the Sausage Machine: Which One Is Baloney?', *Cognition* **8**, 209–225.

Winograd, T.: 1972, *Understanding Natural Language* (Edinburgh University Press).

Woods, W.: 1973, 'Transition Network Grammars', in R. Rustin (ed.), *Natural Language Processing* (Courant Computer Science Symposium 8, Algorithmics Press, New York).