PHRASE STRUCTURE GRAMMAR*

Gerald Gazdar
Cognitive Studies Programme
School of Social Sciences
University of Sussex

July 1980

1. Introduction
2. Complex symbols
3. The interpretation of phrase structure rules
4. Rule format and strict subcategorization
5. English VP and AP rules
6. Coordination
7. Metarules
8. Passive
9. Unbounded dependencies
10. Islands


*. I am indebted, for comments, conversation, and criticism, to Emmon Bach, Lee Baker, Ken Butcher, Steve Draper, Elisabeth Engdahl, Janet Fodor, Paloma Garcia-Bellido, Steve Harlow, Frank Heny, Polly Jacobson, Theo Janssen, Arvind Joshi, Ewan Klein, Bill Ladusaw, John Lyons, Joan Maling, Barbara Partee, Stan Peters, Geoff Pullum, Andrew Radford, Ivan Sag, Aaron Sloman, Neil Smith, Armin von Stechow, Anthony Warner, Tom Wasow, and Annie Zaenen. This research was supported by grant HR 5767 from the SSRC (UK).

1. Introduction

As far as the alleged "return to structuralism" is concerned: first of all, suppose that were true - fine!

It often happens that hypotheses in the natural sciences are abandoned at a certain period because they are inadequate, but are then reconstructed later when a higher level of comprehension has been attained.

Chomsky 1970: 197.

Transformational grammars for natural languages, as currently envisaged, deploy a large number of devices: complex symbols, base rules, rule schemata, lexical insertion rules, lexical redundancy rules, movement rules, coindexing procedures, binding conventions, local and nonlocal filters, case marking conventions, feature percolation, constraints on movement, and so on. The mathematical properties of the resulting baroque systems are almost entirely unknown: we are ignorant, for example, as to whether ungrammaticality with respect to such grammars is decidable, i.e. given an arbitrary string on the terminal vocabulary, no way is known of proving that that string is not generated by the grammar. In this situation, claims by grammarians to the effect that such and such a string of words cannot be generated by their grammar merely reflect their intuitions about the apparatus they are using. These intuitions cannot be verified at present and may indeed by unverifiable in principle (i.e. if the class of grammars permitted under universal grammar generate r.e. sets).

Much work has been devoted in recent years to the question of constraining the class of available grammars for natural languages. But, with honourable exceptions (e.g. Janssen, Kok, and Meertens 1977, Lapointe 1977, Wasow 1978), this work has been free of
serious mathematical content. In view of this fact, claims in the current literature implying that one variant of TG (say one with filters but no obligatory rules) is more restrictive than another (say one with obligatory rules but no filters) are about as sensible as claims to the effect that Turing machines which employ narrow grey tape are less powerful than ones employing wide orange tape.

The strongest way to constrain a component is to eliminate it. In this paper I shall outline a type of generative grammar that exploits several of the resources of transformational grammar (e.g. phrase structure rules, rule schemata, complex symbols, feature conventions) but which, crucially, does not employ either transformations or coindexing devices. This type of generative grammar is provably capable of generating only the context-free (CF) languages and is, to all intents and purposes, simply a variant of CF phrase structure grammar.

One of the metatheoretical motivations for adopting the present approach is that the formal properties of the languages that can be generated, and of the grammars doing the generating, are relatively well understood given the considerable body of mathematical work that now exists on CF languages and grammars (see Book 1973 for a concise survey). Another motivation is that if we only allow ourselves to employ apparatus restricted to CF generative capacity, then we are making a very strong universal claim about the properties of natural languages, one which is presently unfalsified (see Gazdar & Pullum (in preparation) for discussion). Whereas, if we continue to use movement and deletion rules, non-local filters, and/or coindexing devices, then we will be working within a relatively unconstrained theoretical framework or at best one about whose constrainedness we know very little, and we would consequently only be committed to relatively weak universal claims.

In a recent paper, Levelt makes the following observation: 'if it is the Aspects-formalism that constitutes the child's innate knowledge of natural languages, then, given the Peters and Ritchie results (1973) and Gold's (1967) technical definition of learnability, natural languages are unlearnable, since the class is r.e.' (1979: 6). He points out that, given the unacceptability of this conclusion, there are only two routes out of the dilemma it poses: either one considers alternative technical definitions of learnability or else one reduces the class of grammars permitted. He distinguishes two potential ways of doing the latter: 'the intensional way consists of defining "possible grammar" in such a way that the class is small. Going from r.e. languages to decidable to context-free would be such a step' (1979: 8). The extensional way consists of stipulating the class of r.e. set inducing grammars permitted by universal grammar, for example by giving a finite list of such permissible grammars. Levelt then shows that, given Gold's learnability definition, the extensional way provides no solution to the dilemma posed above: 'reducing the cardinality of the class, or using some evaluation measure, are in themselves insufficient to guarantee learnability as long as the grammars generate r.e. languages: the child will never know how to exclude a grammar' (1979: 11). In particular, Levelt is able to prove that the finiteness, or otherwise, of the cardinality of the set of possible natural language grammars is irrelevant to learnability if the languages they characterize are nonrecursive. He shows that even in the limiting case of there being only two possible grammars for the learner to choose between, the choice cannot be made on the basis of some finite set of well-formed strings of the
language unless the two languages are completely disjoint.

Consider now the following technical definition of learnability: a language is learnable if and only if there is an algorithm that will map some finite set of trees into a grammar for the language. Intuitively, the idealization involves the language acquisition device being presented with finite sets of (surface) structures rather than finite sets of well-formed strings (= "text presentation") or finite sets of strings paired with a grammaticality judgement (= "informant presentation"). Now, for every CF grammar there exists a (frontier-to-root) finite state tree automaton that admits exactly the tree set generated by that grammar (Thatcher 1973). And there exist algorithms which will map finite sets of CF trees (not just any finite sets of trees, of course) into the tree automata that admit them (see Fu and Booth 1975, Levine 1979). Given the tree automaton for a CF tree set, it is a trivial matter to construct a CF-PSG for that tree set. No such algorithms exist for non-CF languages because no non-CF language can be induced by a frontier-to-root finite state tree automaton. 

The sentences of a natural language can be parsed. We do it all the time. Furthermore, we do it very fast (see Marslen-Wilson 1975, for relevant psycholinguistic evidence). But 'for transformational grammars, it is not known that processing time can be any less than a doubly exponential function of sentence length' (Peters 1979). Transformational grammars thus fail to provide even the beginnings of an explanation for one of the most important, and most neglected, facts about natural languages: parsing is easy and quick. Sentences of a context-free language are provably parsable in a time which is, at worst, proportional to less than the cube of the sentence length (Valiant 1975, Graham 1976). Many context-free languages, even ambiguous ones, are

- 5 -

provably parsable in linear time (Earley 1970: 99). These parsability results, and the avenues of research that they open up, provide a major motivation for constraining natural language grammars to CF generative capacity (see Sheil 1976, Kaplan 1978, for more detailed discussion).

Nevertheless, as Joshi, Levy and Yueh (forthcoming) pertinently remark, 'context-freeness should not be used as a justification for poor style'. If to do things exclusively by direct phrase structure generation was to lead inevitably to aesthetic disaster (re. simplicity, economy, generality, empirical motivation, etc), whilst competing transformational analyses were paragons of elegance and enlightenment, then one might reasonably feel inclined to reject the former in favour of the latter. However, in later sections of this paper I shall be arguing implicitly that phrase structure analyses can be at least as elegant, general, etc., and no more prone to counterexamples, than the alternative transformational accounts of the same phenomena.
2. Complex symbols

Harman (1963) deserves the credit for first seeing the
potential of PSGs incorporating complex symbols. The use of a
finite set of complex symbols (however constructed) in a PSG, in
place of the traditional finite set of monadic symbols, leaves
weak generative capacity unchanged: a CF-PSG employing complex
symbols will only be able to generate CFLs. Furthermore, if we
take grammars that induce isomorphic structural descriptions to
be strongly equivalent, then the use of complex symbols has no
effect on strong generative capacity either: every grammar
employing complex symbols generates a tree-set that is isomorphic
to the tree-set generated by some CF-PSG not employing complex symbols,
and conversely.

In the light of these self-evident and trivial observations,
it is surprising that Chomsky once saw fit to claim that 'a system
... that makes use of complex symbols is no longer a phrase
structure grammar ..., but rather is more properly regarded as a
kind of transformational grammar' (1965: 98). Transformational
grammars can move arbitrarily large constituents about, delete
constituents under identity, substitute one morpheme for another,
restructure trees, allow one to make a deep structure/surface
distinction, and generate any recursively enumerable set you care
to think of. Phrase structure grammars which employ complex
symbols can do none of these things, so it is hard to see why one
would want to refer to such grammars as 'transformational'..

For the most part, the complex symbol system adopted in this
paper is simply a variant of the type that has become standard
in recent TG. We assume a two-bar X system\textsuperscript{5} that distinguishes
between $\bar{X}$, $\bar{X}$, and X (lexical) categories, and so one component
of a complex symbol must provide an indication of this distinction.
The exact manner of this indication is immaterial: we shall
stipulate that it takes the form of an integer as the first member
of an ordered pair (following Brosnan 1976b). The other
component of a complex symbol will be a feature bundle encoding
syntactic category, subcategorization, and morphosyntactic and
morphological information.\textsuperscript{6} We will use a familiar notation for
such familiar objects. For example, $\bar{N}(\text{PRO}, \text{SG})$ represents
the phrasal [+N, -V] plural pronominal complex symbol for which
a pedantic representation might be $<2, \{+N, -V, \text{PRO, SG}, \ldots\}>$. Where features are left unspecified, then in general the node-
label stands as a variable ranging over permissible feature combinations.

We assume that there are general, putatively universal,
conventions of feature distribution. The most important of these
is what we shall call the Head Feature Convention (HFC, hereafter):

\begin{equation}
\text{(2.1) HFC: In a rule of the form } \Delta \rightarrow \ldots \delta \ldots \text{ where }
\delta \text{ is the head of } \Delta, \delta \text{ carries all the features associated with } \Delta.
\end{equation}

We may define "head of", at least for the purposes of this paper,
as follows:

\begin{equation}
\text{(2.2) In a rule of the form } \Delta \rightarrow \ldots \delta \ldots, \delta \text{ is the head of } \Delta \text{ if and only if }
\begin{align*}
(i) \quad & \Delta = \langle i, \{+N, +V\} > \\
(ii) \quad & \delta = \langle j, \{+N, +V\} > \\
(iii) \quad & j \leq i \\
(iv) \quad & \text{there is no } \delta' (\delta' \neq \delta) \text{ introduced by this rule such that } \delta' = \langle k, \{+N, +V\} > \text{ where } k \leq j
\end{align*}
\end{equation}

Something rather similar to HFC is implicit or explicit in a lot
of recent transformational grammar (see, e.g. Baker 1978: 336 ff. Hellan 1977: 90-91) but it has not often been fully appreciated just how much follows from such a convention. For example, take English subject-verb agreement. Suppose that the tensed S expansion rule is this:

\[
(2.3) \quad \overline{V} \rightarrow \overline{N} [\{\alpha\}] \overline{V} \quad \text{where} \quad \alpha \text{ ranges over permissible combinations of agreement features.}
\]

Note that we are following Jackendoff (1977) here in taking S to be the maximal projection of V. Now the HFC, taken together with (2.3) and the other phrase structure rules we would need in any case, suffices to capture all the straightforward facts about subject-verb agreement in English. The following tree will make it clear how this is so:

\[
(2.4)
\]

We do not have to locate the subject noun, look up what agreement features it is carrying, then locate the first tensed verb in the \( \overline{V} \) and copy the noun's agreement features across. Instead (2.3) ensures that both \( \overline{N} \) and \( \overline{V} \) carry the same agreement features, and the HFC ensures that these find their way onto the relevant head noun and head verb. The HFC also provides a very simple solution to the problem of ensuring the correct order of tense and participial affixes in constructions involving iterated auxiliary verbs - see Gazdar, Pullum and Sag (1980) for detailed support of this claim.

The rule given in (2.3) is, in fact, a finite rule schema which collapses \( \overline{P} \) phrase structure rules where \( \overline{P} \) is the number of permissible combinations of agreement features. Most of the rule to be given in this paper are, implicitly or explicitly, finite rule schemata of this general type. Employing such finite rule schemata does not effect the type of grammar we are using: each rule schema can always be expanded into some finite set of non-schematic CF-PSG rules and the resulting grammar will simply be a CF-PSG. However, finite rule schemata do provide an elegant way of capturing syntactic generalizations.

We make two additions to the orthodox feature system outlined above, one minor, the other major. The minor modification allows the feature bundle component to include as features the names of certain coordinating morphemes, complementizers and prepositions. The major modification consists in allowing ordered pairs of complex symbols, as so far characterized, also to count as part of the nonterminal vocabulary of the grammar. This innovation will be discussed in detail in section 8, below.
3. The interpretation of phrase structure rules

In the LSLT theory, the phrase-structure grammar generates derivations, which determine P-markers. Alternatively, one might interpret the rules of the phrase-structure grammar as "tree conditions" that determine the set of P-markers in a way that one could proceed to define. Plainly, nothing is at stake in this choice of interpretations. ... and obviously it makes no difference whether ... we consider the rules to be rewriting rules or "tree conditions".


There are many ways of interpreting the formalism of a phrase structure grammar but only two of these ways need concern us here. One way, the "Logical Structure of Linguistic Theory" way, interprets a phrase structure rule as a rewriting rule, a rule which maps strings into strings. Thus the rule in (3.1):

\[(3.1)\] \( S \rightarrow NP \ VP \)

is a function which maps strings of the form \( X-S-Y \) into strings of the form \( X-NP-VP-Y \). The derivation of some terminal string is the set of all the strings that arise in the mapping from the initial symbol to that terminal string. Given certain restrictions, which we shall return to, a tree may then be defined on the basis of the derivation.

The second way of interpreting PS rules, due originally to Richard Stanley (see McCawley 1968: 39), is to treat them as node admissibility conditions. A node labelled S in a tree is admitted by the rule in (3.1) if and only if that node immediately and exhaustively dominates two nodes, the left one labelled NP and the right one labelled VP. A tree is analysed by the grammar if and only if every non-terminal node is admitted by a rule of the grammar. Under this interpretation, then, phrase structure rules are well-formedness conditions on trees. There is no notion of a derivation and it makes no sense to order the rules.

Chomsky claims, in the quotation which heads this section, that the choice between these interpretations is a matter of no consequence. He is simply wrong about this as we shall see below. Sampson points out, in a review of LSLT, that 'one thing at stake is that Chomsky's approach forces him to impose two quite arbitrary restrictions on phrase structure rules, namely that no rule may rewrite any symbol A as either the null string or as a sequence including A; both of these forbidden types of rule frequently seem appropriate in the description of real languages, and under the alternative view of phrase-structure grammars there is no objection to them' (1979: 368). Among the familiar rules that these arbitrary restrictions prohibit are the following:

\[(3.2)\] a. \( \text{NOM} \rightarrow \text{NOM} S \)
   b. \( \text{NP} \rightarrow \text{NP} S \)

\[(3.3)\] a. \( \text{VP} \rightarrow V \ VP \)
   b. \( \text{NP} \rightarrow \text{PRO}, \text{SUBJ} \rightarrow e \)

\[(3.5)\] \( \text{COMP} \rightarrow e \)

Both competing analyses of relative clauses (3.2), a common treatment of auxiliary verbs (3.3a), the rule that a verb like make needs (3.3b), subject pronoun drop in Romance languages (3.4), and null COMP expansion (3.5) are all eliminated.
Chomsky has claimed that phrase structure grammars have the weak generative capacity of unrestricted rewriting systems (arbitrary Turing Machines) if one symbol is taken to be "blank" (i.e. the empty string $\varepsilon$) (1977: 132). And in a standard transformational textbook we find the following: "it is possible to prove that the class of languages generable by phrase-structure grammars with free deletion rules is exactly the same as the class of languages generable by unrestricted rewriting systems. We are looking for restrictions on grammars that will narrow down the class of possible languages. But by including deletion rules in our phrase structure grammars, we would open the back door after carefully closing the front door. This argument must be taken on faith at this point" (Bach 1974: 42). The claim being made by Bach and Chomsky is misleading to say the least: it is only true of context-sensitive (CS) PSGs under a rewriting interpretation of the rules. It is not true of CF-PSGs under any interpretation (Bar-Hillel, Perles & Shamir 1961:85), and it is not true of a CS-PSGs grammars under a node admissibility interpretation. The only effect on weak generative capacity of allowing $\varepsilon$-productions in PSGs (whether CF or CS) under a node admissibility condition interpretation, is that the language consisting solely of the empty string can then be generated.

Perhaps the most important mathematical difference between the alternative interpretations of PSGs only emerges when we consider CS rules like that shown in (3.6):

\[(3.6) \ A \rightarrow \ \omega/\Psi \ \Psi \]

In a surprising and neglected paper, Peters and Ritchie (1969, 1973) showed that CS-PSGs under a node admissibility interpretation can only analyze CF languages. However, as is well known, such grammars under a rewriting interpretation can generate any CS language. There is thus a well-defined algebraic sense in which the node admissibility interpretation of PSGs is more restrictive than the rewriting interpretation. Since there is, at present, no reason whatsoever for believing that any natural languages are strictly CS (i.e. CS but not CF), it is proper to assume the more restrictive interpretation of PSGs, namely the node admissibility interpretation.

Joshi and his associates (Joshi and Levy 1977, Joshi, Levy, and Yeh, 1978, forthcoming) have recently generalized the Peters and Ritchie result to phrase structure grammars which are allowed to make reference not just to left-right tree contexts but also to top-bottom tree contexts. Thus the contextual conditions can include both proper analysis predicates of the form $\Phi \Psi$, as in (3.6) above, and domination predicates of the form $\ell(A, \Phi \Psi)$ which holds of a node labelled $A$ just in case there exists a path from the root of the tree to some terminal symbol which contains a substring the string $\Phi A \Psi$. They then define a class of phrase structure rules called "Local constraints" that have the form shown in (3.7):

\[(3.7) \ A \rightarrow \ \omega/C_A \]

where $C_A$ is any Boolean combination of proper analysis and domination predicates.

For example, a grammar of English might include the following pair of rules:

\[(3.8) \ a. \ \overline{\sigma} \rightarrow \ \text{that} \ \sigma \\
   b. \ \overline{\sigma} \rightarrow \ \sigma/\Psi(\sigma)\]
The local constraint in (4.8b) will ensure that no S which is immediately dominated by S will appear without the complementizer that.

Phrase structure grammars which employ local constraints can only analyze CF languages.

Any local filter, in the sense of Chomsky and Lasnik (1977), can be mapped into local constraint type contextual conditions on phrase structure rules. It follows that augmenting a phrase structure grammar (under a node admissibility condition interpretation) with a local filter component will have no effect on the weak generative capacity of the overall grammar - it will still only be able to generate CF languages. Stan Peters has pointed out to me that this result can be proved directly, and more elegantly, by defining local filters in tree automata terms, but I will not pursue that here.10

It is worth noting, since it is by no means intuitively obvious, that the n-s parsability result for CF languages holds also when the PSG defining the set of parse trees employs e-productions (see Aho and Ullman 1972: 320ff) and when it incorporates local constraints (see Joshi, Levy, and Yueh, forthcoming, for proof).

I will not in fact employ CS rules, local constraints or local filters anywhere in the present paper. My purpose in the preceding digression was to establish (i) that the interpretation of phrase structure rules affects their properties in an important way, and (ii) that the class of CF-language-inducing PSGs is much more interesting formally than most linguists currently appear to believe.

4. Rule format and strict subcategorization

Since we are interpreting phrase structure rules as node admissibility conditions rather than as string-to-string mapping rules, we will not use the familiar rewrite arrow notation for PS rules, but instead use a notation which reflects more directly the relation such rules bear to the (sub)trees that they admit. Instead of (4.1)

\[ (4.1) \quad \text{\overline{V} } \to \text{\overline{N} } \text{\overline{V}} \]

then, we will write

\[ (4.2) \quad [\text{\overline{V}} \text{\overline{N} } \text{\overline{V}}] \]

and analogously for all other rules.

We assume that each syntactic rule in the grammar should be associated with a semantic rule which gives the meaning of the constituent created by the syntactic rule as a function of the meaning of the latter's parts. We further assume that the semantic rules should take the form of rules of translation into intensional logic. These two assumptions commit us to what Bach (1976: 2) has called the rule-to-rule hypothesis concerning the semantic translation relation. We take a rule of grammar to be a triple of which the first member is an arbitrary integer - the number of the rule (the role of which will become apparent shortly), the second member is a PS rule, and the third is a semantic rule showing how the intensional logic representation of the expression created by the PS rule is built up from the intensional logic representations of its immediate constituents. We will use a Montague-like prime convention in the semantic rules: \text{\overline{N}} stands for the (complex) expression of intensional logic which is the
translation of the subtree dominated by $\overline{\text{N}}$, $\text{run}'$ is the constant of intensional logic which translates the word $\text{run}$ in English, etc. And we adopt a further prime convention for intensions: we let $\alpha''$ stand for the intension of $\alpha'$, thus $\alpha'' = \overset{\prime}{\alpha}'$. Within this framework the celebrated S = NP VP rule of English would be stated thus:

\[(4.3) \langle 1, \left[ \begin{array}{c} \overline{\text{N}} \\ \overline{V} \end{array} \right], \overline{V}'(\overline{N}') \rangle\]

We will use "rule" to refer both to the triple and to its second and third members (sometimes qualifying the latter with "syntactic" and "semantic", respectively) but this should not cause confusion.

Notice that the semantic rule in (4.3) is not the one adopted by Montague (1973) in FTQ in which the NP is a function taking the VP as argument. Instead we are taking VPs to denote functions from NP intensions to truth values, following Montague's (1970) earlier treatment in his "Universal Grammar". This latter treatment has recently been strongly motivated on phonological, syntactic, and semantic grounds by Thomason (1976), Keenan and Faltz (1978), Keenan (1979a), and Bach (1980b). For example, this way of doing things makes it easy to ensure that (4.4a) does not entail (4.4b):

\[(4.4)\]

\[a. \text{ A unicorn seems to be approaching.}\]

\[b. \text{ There exists an entity such that that entity seems to be approaching.}\]

But making sure that (4.4a) does not wind up entailing (4.4b) is difficult to do if one retains the FTQ rule.

Let us turn now to rules for PP: 11

\[(4.5) \langle 2, \left[ \begin{array}{c} \overline{\text{P}} \end{array} \right], \overline{P}' \rangle\]

A language like Latin marks NPs which stand as indirect object, passive agents, etc., with morphological case-marking as dative, ablative, etc. In English one finds a class of PPs where in Latin one would have NPs and these PPs are distinguished, not by casemarking but by choice of preposition (cf. Jackendoff 1977: 80-81, Fodor 1978: 444 n 12). Suppose then that such PPs carry the name of the particular preposition as a feature so that the grammar employs complex symbols of the form $\overline{P}_{\text{to}}, \overline{P}_{\text{of}}, \overline{P}_{\text{for}}$, and $\overline{P}_{\text{by}}$. These PPs can nevertheless still be expanded by means of the regular PP rules as given in (4.5) and (4.6). The Head Feature Convention will carry the feature down onto the prepositions that are the heads of such PPs, and then the following rule can realize the feature as the relevant preposition:

\[(4.7) \langle 3, \left[ \begin{array}{c} \overline{\text{P}} \\ \overline{\text{P}} \end{array} \right], \lambda r(r) \rangle\]

\[(3)\]

where $\alpha \in \{\text{to, for, by, of, ...}\}$ and $r$ is of type $\langle<e,t>,t\rangle$.

The semantic rule here is an identity function mapping NP extensions into themselves. Thus this kind of PP will end up having exactly the same meaning as the NP it dominates. This analysis makes the claim then that in such PPs the preposition does not carry any independent meaning but serves merely to indicate the argument place occupied by the NP. This is in marked distinction to other PPs which we would want to get assigned an adverbial-type meaning that varied in a regular way depending on the particular preposition involved. There is some evidence for the semantics given in (4.7), consider the following pairs of sentences:
(4.8) a. Kim gave a book only to Sandy.
    b. Kim gave a book to only Sandy.

(4.9) a. Kim bought a book only for Sandy.
    b. Kim bought a book for only Sandy.

(4.10) a. Sandy was given a book only by Kim.
      b. Sandy was given a book by only Kim.

These pairs are truth-conditionally synonymous as (4.7) predicts they will be. But now consider similar minimal pairs where a "real" preposition is involved:

(4.11) a. Kim put books only on the boxes.
      b. Kim put books on only the boxes.

(4.12) a. Kim left only after two days.
      b. Kim left after only two days.

Clearly these pairs are not truth-conditionally synonymous, nor would we expect them to be. There is a related bit of evidence for the semantics in (4.7): in Spanish "real" PPs can be modified by como (how) and casi (almost) whereas subcategorization PPs cannot be so modified (I am indebted to Paloma Garcia-Bellido for this observation). This would be predictable if subcategorization PPs had NP-like meanings rather than PP-like meanings.

Any grammar for a natural language has to provide some way of capturing the fact that different lexical items of the same (gross) syntactic category can have different distributions. In a transformational grammar of a traditional kind this fact is described using the devices of strict subcategorization rules and selectional restrictions. We assume without argument that all the work done by the latter should be done by the semantics or pragmatics. One obvious way to do strict subcategorization in a phrase structure grammar is by means of CS rules like that in (4.13):

(4.13) V + throw/ ____ NP up

As noted in section 3, above, the use of CS-PSG rules, interpreted as node admissibility conditions, would not allow any non-CF languages to be generated. However, the approach to strict subcategorization implicit in rules like (4.13) would be to all intents and purposes equivalent to the Aspects positions on strict subcategorization. Heny has recently pointed out that the latter succeed in missing a rather significant generalization:

The internal structure of every strict subcategorization feature, including those that have to be included in the lexical specification of an item (such as +[(____ NP-Manner) on frighten]), precisely mirror the order and optionality of the elements in the PS rule expanding the node immediately dominating the item in question. Thus, the lexicon will necessarily repeat information, time and time again, which is at least in part already extractable from the PS rules. No verb can have the feature +[(____ Manner-NP) in English, because no PS rule or combination of PS rules introduces the elements V, NP and Manner in the relevant order. This problem can be viewed entirely from within the lexicon or from the point of view of the interaction of the lexicon and the PS rules. Looked at either way something is amiss. Heny 1979: 339-340.
And Carlson and Roeper draw attention to:

A potential difficulty with the phrase structure rules for the English VP. Attempts to write detailed VP PS rules, such as we find in Jackendoff (1977), invariably allow for expansions that do not occur. Carlson and Roeper 1980: 162.

Jackendoff's VP expansion rule looks like this:

\[(4.14) \quad \overline{V} \rightarrow V \ (NP) \ (Prt) \ ((^{\text{NP}} \ _{\text{Adv}})) \ (^{\text{Adv}} \ _{\text{QP}}) \ (PP) \ (PF) \ (S)\]

(Jackendoff 1977: 64)

Assuming this rule, and assuming that throw is subcategorized as in (4.13), our grammar will generate the following string:

\[(4.15) \quad *\text{Kim threw the meal up Sandy slowly to Lee in the woods that Sandy loved Lee.}\]

Example (4.15) is not a sentence of English. Two solutions present themselves, neither being very attractive. The syntactic solution is to augment (possibly to convention, see Chomsky 1965: 111) the rule in (4.13) and every other subcategorization rule with a whole list of negative contextual conditions to rule out most of the possibilities introduced by (4.14). This seems an ungainly way to circumvent the problem that (4.14) creates. The semantic solution is to argue that (4.15) is, in fact, grammatical, but that it cannot be interpreted by the semantic component of the grammar. The logic of this line of argument leads one to abandon syntactic restrictions on subcategorization altogether, otherwise one ends up in the peculiar position of claiming that whilst (4.15) is bad for semantic reasons, (4.16) is bad for syntactic reasons:

\[(4.16) \quad *\text{Kim throw.}\]

A serious problem with a purely semantic approach to subcategorization emerges when we consider minimal pairs such as those shown in (4.17) - (4.21):¹²

\[(4.17) \quad a. \ \text{Kim is likely to throw up.}\]
\[b. \quad *\text{Kim is probable to throw up.}\]

\[(4.18) \quad a. \ \text{Sandy made Kim throw up.}\]
\[b. \quad *\text{Sandy forced Kim to throw up.}\]

\[(4.19) \quad a. \ \text{Sandy forced Kim to throw up.}\]
\[b. \quad *\text{Sandy made Kim a second helping.}\]

\[(4.20) \quad a. \ \text{Sandy spared Kim a second helping.}\]
\[b. \quad *\text{Sandy deprived Kim of a second helping.}\]

\[(4.21) \quad a. \ \text{Sandy deprived Kim of a second helping.}\]
\[b. \quad *\text{Sandy spared Kim of a second helping.}\]

There does not seem to be any way in which the clear differences in acceptability of these examples can be made to follow from the meaning differences, if any, between the lexical items involved.

Coordination interacts with CS subcategorization in a technically problematic manner. Assume that both throw and hand have \([^\_\_\_\_\_\_\_\_NP \ PP]\) among their subcategorization frames. Now consider how one might generate example (4.22):

\[(4.22) \quad \text{Kim threw and handed things to those outside.}\]

Clearly, handed meets the subcategorization requirement, being left-adjacent, as it is, to an NP followed by a PP. Equally clearly, throw does not meet the subcategorization requirement since
it is left-adjacent to and-V-NP-PP. A transformational grammar which is prepared to employ a rule of coordination reduction will not run into this problem, of course. In a phrase structure grammar using CS subcategorization the only way to handle coordination of lexical categories seems to be to introduce a special convention for interpreting the context condition in such cases. Such a convention will itself lead to problems. Thus presumably both prefer and promise will have (___NP ___VP) among their subcategorization frames. If so, the convention just alluded to will ensure that our grammar generates the string in (4.23):

(4.23) *Kim preferred and promised Sandy to go.

It is quite possible that the problems with CS lexical insertion discussed above are amenable to more or less satisfactory solutions. However, rather than pursue that here, I propose to develop an alternative approach to strict subcategorization, one which only employs context-free rules. The format for rules, as outlined at the beginning of this section, enables us to capture the unruly and idiosyncratic syntactic facts of subcategorization in a fairly elegant way. Suppose we have a rule of grammar 

\[ \text{(4.24) } <n, \ldots \text{C...}, \ldots> \]

\[ \text{(4.25) } <n, \ldots \text{C...}, \ldots> \]

\[ \text{[n]} \]

This use of rule numbers as subcategorization features eliminates the need for context-sensitive rules of lexical insertion. A context-free PS rule can allow 

\[ \text{C} \[n \] \]

To dominate only those lexical items permitted in the context defined by rule n. A direct consequence of this is that the proposals regarding subcategorization stipulated by Chomsky (1965: 96ff) fall out as theorems in the present system. Chomsky stipulates that a category C can be subcategorized only for material a and b such that C is introduced by a rule a \[ CS \] (1965: 99). Consider an example:

\[ (4.6) <9, [\text{V} \text{ V} \text{ N} \text{ V}] \ldots> \]

\[ \text{where V[9] = \{hand, sing, throw, give, \ldots\}} \]

Rule 8 says that a VP can consist of a V(9) followed by an NP followed by a dative PP. And among the lexical items that can be dominated by V(9) are hand, sing, throw, give, etc. Nothing outside \[ V \] could be relevant to deciding whether or not hand, say, can be inserted. This is what is guaranteed by Chomsky's rule "V = CS/\[a\], where a is a string such that Va is a VP" (1965: 96). Note that the use of complex symbols enables us to avoid the charge usually levelled against such context-free phrase structure proposals for lexical insertion, namely that by distinguishing V\[i\] from V\[j\], say, we lose generalizations about verbs (e.g. that they all take tense). We do not lose the generalizations since V[i] and V[j] have at least two features in common (namely [+V, +N]) and it is this fact which accounts for the generalizations that can be made.

Before ending this section, we need to say something about the assumptions we are making with respect to inflectional morphology. Lexical categories may bear morphosyntactic features, thus a tree may, for example, contain a node labelled V[9, +PRP],
where +PRP indicates a present participle, immediately dominating handing. Like Lapointe (1980), we assume that such forms are given by the lexicon directly and not constructed by affixation of -ing to hand by some syntactic rule as Affix Hopping or the syntactically triggered morphological rules of Pullum and Wilson (1977). In the case of a feature like +PRP the phonological shape of the word form will be fully predictable by a general lexical redundancy rule, whereas in the case of a feature like +NEG the lexicon will need to specify the idiosyncratic forms (e.g. won't/*willn't) and the accidental gaps (e.g. *awn't). See Gazdar, Pullum, and Sag (1980) for detailed proposals concerning the role of morphosyntactic features in stating the regularities of the English auxiliary system.

5. English VP and AP rules

The rules to be given in this section combine the approach to subcategorization developed above with (i) Bresnan-style claims (e.g. in her 1978) about syntactic categories and constituent structure, and (ii) a Montague-based approach to semantics. A similar Bresnan-Montague marriage has already been exploited very successfully by Klein (1978), Ladusaw (1979), and by McCloskey (1979) in his grammar of Modern Irish, and the present proposals are indebted to those works. In this kind of approach, all the semantic work done in a classical transformational grammar by lexically governed syntactic rules like Equi and Raising is done by a combination of lambda abstraction and meaning postulates. Since the syntactic proposals which are the main focus of this paper are almost entirely independent of the details of the semantics adopted, I shall hardly go into the latter at all, and instead simply refer the reader to Thomason (1976) and Dowty (1978) where the relevant issues are given serious consideration.19

Following Hust and Brame (1976: 251), we will assume that all verbs are marked with a feature indicating whether or not they are transitive (we will use (±TRN) for this purpose). Crosslinguistically, there are compelling reasons for rejecting Chomsky's claim that such a feature 'can be regarded merely as a notation indicating occurrence in the environment ___NP' (1965: 93). One obvious problem is that his claim will lead us to define transitivity in four different ways in order to accommodate (i) SVO and VOS, (ii) OVS and SOV, (iii) OSV, and (iv) VSO languages. If there are crosslinguistic generalizations to make about transitive verbs then they will certainly be missed by treating...
transitivity merely as a contextual condition on lexical insertion. Another problem is that in certain Micronesian languages, e.g. Kusaiean, verbs may appear in both transitive and intransitive forms in the environment ___NP, and it is their transitivity, not their adjacency to NP, that determines the applicability of passive, the possibility of adding NP modifiers, and the position of verb suffixes (Comrie 1979: 1064-1065). And in Hindi-Urdu both transitive and intransitive verbs can occur in the environment NP V ___ # but the subject will only take ergative case if the verb that appears in this environment is a transitive one. Amritavalli (1979) argues at length that it is impossible to capture the relevant generalizations about ergative case marking and passivization in Hindi-Urdu if transitivity is identified with [NP ___ #]. He concludes 'that verbs should be marked both for transitivity and for strict subcategorization features, and that these features are independent of each other' (ibid: 91). Arguments suggesting that this conclusion carries over to English are developed at some length in Bach (1980a) and we will henceforth assume its correctness without further discussion. In addition, we will adopt Amritavalli's markedness convention (ibid: 92) which we can formalize as follows in the present framework:

In a rule of the form:

\[
[\overline{V} \hspace{1em} V \hspace{1em} X]
\]

the unmarked value for a is + if

\[X = \overline{N} Y,\] and - otherwise.

Following Gazdar, Pullum and Sag (1980) we take Bresnan's "\(\overline{V}V\)" category to be an infinitive verb phrase whose head is the untensed, uninflected auxiliary verb to. Thus:

\[
(5.1) \langle 5, \{\overline{V} \hspace{1em} V \hspace{1em} \{BSE\} \}, \lambda F \hspace{1em} V' (\cdot \overline{V}' (F))\rangle
\]

where to is the only item of category V[5].

Here the feature BSE on the embedded \(\overline{V}\) ensures that the head of that \(\overline{V}\) appears in the bare infinitive form (via the HFC); see Gazdar, Pullum and Sag (ibid) for detailed discussion of this rule which is, in fact, merely an instance of a much more general rule schema for the introduction of auxiliary verbs.

Following Gazdar (forthcoming), we take the familiar "S" category to be a sentence optionally marked by the feature +C (complementizer), hence \(S = V[+C]\). Here \(V[-C]\) is an ordinary tensed sentence, and \(V[+C]\) expands in virtue of the following rule:

\[
(5.2) \langle 6, \{\overline{V} \hspace{1em} that \hspace{1em} \overline{V}' \hspace{1em} (\cdot \overline{C})\} \rangle
\]

Most verbs and adjectives that take sentential complements subcategorize for \(\overline{V}[+C]\), but a few require \(\overline{V}[+C]\) and consequently the that is obligatory in their complements (see Shir 1977: 62-3 for relevant data). Henceforth we will abbreviate \(\overline{V}[+C]\) as \(\overline{V}\).

In the examples that follow, the a-line defines a rule of number \(n\), the b-line lists example lexical items of category \(V[n]\) (or A[n]), and the c-line gives an example of a constituent admitted by the rule.

\[
(5.3) \begin{array}{lll}
\text{a.} & \langle 7, \{\overline{V} \hspace{1em} V'\} \rangle & \\
\text{b.} & \text{run, die, eat, sing, ...} & \\
\text{c.} & \text{runs.}
\end{array}
\]
(5.4) a. \(<8, \{V, V \overset{\lambda}{=} N\}, V'(N')\>
b. eat, sing, love, give, close, ...
c. eats Fido.
(5.5) a. \(<9, \{V, V \overset{\lambda}{=} N, V' \overset{\lambda}{=} P\} \overset{\lambda}{=} N', V''(P')(N'')\>
b. hand, give, sing, throw, ...
c. hands Fido to Kim.
(5.6) a. \(<10, \{V, V \overset{\lambda}{=} N, V' \overset{\lambda}{=} P\} \overset{\lambda}{=} N, V''(P')(N'')\>
b. buy, cook, reserve, ...
c. buys Fido for Kim.
(5.7) a. \(<11, \{V, V \overset{\lambda}{=} N, V' \overset{\lambda}{=} N\} \overset{\lambda}{=} N, V''(N')(N'')\>
b. spare, hand, give, buy, ...
c. spares Fido a bath.
(5.8) a. \(<12, \{V, V V\} \overset{\lambda}{=} N, V''(V'')\>
b. know, believe, expect, ...
c. knows that Fido runs.
(5.9) a. \(<13, \{V, V \overset{\lambda}{=} N V\} \overset{\lambda}{=} N, V''(N')(N'')\>
b. promise, ...
c. promises Kim that Fido runs.
(5.10) a. \(<14, \{V, V \overset{\lambda}{=} N V\} \overset{\lambda}{=} N, V''(N')(N'')\>
b. persuade, tell, ...
c. persuades Kim that Fido runs.
(5.11) a. \(<15, \{V, V \overset{\lambda}{=} V\} \overset{\lambda}{=} N, V''(V'')\>
b. try, ...
c. tries to run.
(5.12) a. \(<16, \{V, V \overset{\lambda}{=} V\} \overset{\lambda}{=} N, V''(V'(P'))\>
b. tend, happen, ...
c. tends to run.
(5.13) a. \(<17, \{V, V \overset{\lambda}{=} V\} \overset{\lambda}{=} N, V''(V')(V')\>
b. want, prefer, expect, ...
c. wants to run.
(5.14) a. \(<18, \{V, V \overset{\lambda}{=} N \overset{\lambda}{=} V\} \overset{\lambda}{=} N', V''(N')(V'')\>
b. want, prefer, ...
c. wants Fido to run.
(5.15) a. \(<19, \{V, V \overset{\lambda}{=} N \overset{\lambda}{=} V\} \overset{\lambda}{=} N', V''(N')(V'')\>
b. expect, believe, ...
c. expects Fido to run.
(5.16) a. \(<20, \{V, V \overset{\lambda}{=} N \overset{\lambda}{=} V\} \overset{\lambda}{=} N', V''(N')(V'')\>
b. persuade, ask, force, ...
c. persuades Fido to run.
(5.17) a. \(<21, \{V, V \overset{\lambda}{=} N \overset{\lambda}{=} V\} \overset{\lambda}{=} N', V''(N')(V'')\>
b. make, ...
c. makes Fido run.
(5.18) a. \(<22, \{V, V \overset{\lambda}{=} N \overset{\lambda}{=} V\} \overset{\lambda}{=} N, V''(V'(N'))\>
b. promise, ...
c. promises Kim to run.
(5.19) a. \(<23, \{V, V \overset{\lambda}{=} N \overset{\lambda}{=} V\} \overset{\lambda}{=} N, V''(V'(P'))\>
b. seem, appear, ...
c. seems to Kim to run.
(5.20) a. \(<24, \{V, V \overset{\lambda}{=} N \overset{\lambda}{=} V\} \overset{\lambda}{=} N, V''(N')(V'')\>
b. be, ...
c. is stupid.
(5.21) a. $\langle 25, \{ \neg V \ V \ A \}, \lambda \Phi(\neg\Phi'((\neg A')(\Phi))) \rangle$
   b. seem, appear, ...
   c. seems stupid.

(5.22) $\langle 26, \{ \neg A \ A \}, \ A' \rangle$

(5.23) a. $\langle 27, \{ \neg A \ A \}, \ A' \rangle$
   b. stupid, open, closed, ...
   c. stupid.

(5.24) a. $\langle 28, \{ \neg A \ P \}, \lambda \Phi(\Phi'((\neg P')\Phi')) \rangle$
   b. known, attracted, drawn, ...
   c. attracted to Fido.

(5.25) a. $\langle 29, \{ \neg A \ P \}, \ A'((\neg P')\Phi') \rangle$
   b. uninhabited, unloved, ...
   c. uninhabited by man.

(5.26) a. $\langle 30, \{ \neg A \ V \}, \lambda \Phi((\neg\Phi'((\neg V')\Phi))) \rangle$
   b. likely, ...
   c. likely to run.

(5.27) a. $\langle 31, \{ \neg A \ V \}, \lambda \Phi(\lambda x: A'((\neg\Phi')((\Phi'x)(\Phi'x)))) \rangle$
   b. eager,
   c. eager to run.

(5.28) a. $\langle 32, \{ \neg A \ P \ V \}, \ A'((\neg V')((\neg P')) \rangle$
   b. eager, ...
   c. eager for Fido to run.

In rules 22, 23 and 29 above, we are assuming the following convention for the semantics of optional arguments:

(5.29) Optional Argument Convention:

If $\beta' = \ldots (\alpha') \ldots$, where $\beta$ immediately dominates an optimal constituent $\alpha$ and $\alpha'$ is of type $\langle s, \langle e, t \rangle, t \rangle$, then when $\alpha$ is omitted, $\beta' = \ldots (\hat{\Phi}(\text{Ex}(\Phi(x)))) \ldots$

This has the effect of ensuring existential quantification into missing argument positions. We exploit this convention in the treatment of agentless passives given in section 8, below.

The rules given above combine together to admit trees like the following:
6. Coordination

CF-PSGs, as standardly defined, are subject to the restriction noted in (6.1):

(6.1) A grammar contains at most finitely many rules of the form $A \rightarrow \omega$
where $A \in V_N$ and $\omega \in (V_N \cup V_T)^+$.  

This restriction can be seen to pose a technical problem when one recalls that natural language coordinate structures are (i) flat rather than nested, and (ii) indefinitely extensible. Linguists typically provide for these facts by proposing rules of the general form shown in (6.2):

(6.2) $\alpha \rightarrow \alpha \ldots \alpha$ where $\alpha \in V_N$

This is at odds with the restriction noted above since, on the intended interpretation, (6.2) is neither a well-formed CF-PSG rule, nor does it stand for a finite set of such rules.

This problem is entirely an artefact of the restriction in (6.1). Suppose we consider a class of grammars, identical to the CF-PSGs as standardly defined except insofar as they are subject to the restriction given in (6.3) rather than that of (6.1):

(6.3) A grammar contains at most finitely many rules of the form $A \rightarrow \omega$
where $A \in V_N$ and $\omega$ is a regular expression over $V_N \cup V_T$.

This class of grammars will generate all and only the languages generated by CF-PSGs as restricted by (6.1).

We will henceforth adopt the restriction of (6.3) in preference to that of (6.1), and we will refer to the grammars so characterized as regular CF-PSGs.

Natural languages appear to exhibit only four types of regular CF-PSG rules for coordination. These are shown in (6.4):

(6.4) a. $\alpha \rightarrow \alpha^+$
     [3]
b. $\alpha \rightarrow \alpha \omega^+$
     [3]
c. $\alpha \rightarrow \alpha^*$
     [3]
d. $\alpha \rightarrow \alpha \omega^+$
     [3]

Here $\alpha^+$ is the positive closure of $\alpha$ (i.e. $\alpha, \alpha\alpha, \alpha\alpha\alpha, \ldots$). There are three parameters of language variation: (i) the range of permissible values of $\alpha$, (ii) the ways in which the coordinate feature $\beta$ is spelt out, and (iii) which of (6.4a-d) are employed. Thus in English $\alpha$ can stand for any syntactic category, $\beta$ is always spelt out as and or or as the left daughter of $\alpha[\beta]$, and both (6.4c) and (6.4d) are to be found in the grammar but not either (6.4a) or (6.4b). Thus:

(6.5) a. *And Kim, and Sandy, Hilary.
     b. *And Kim, Sandy, Hilary.
     c. Kim, Sandy, and Hilary.
     d. Kim, and Sandy, and Hilary.

In Latin, $\beta$ can be spelt out as "que" as the right sister of the first work in $\alpha[\beta]$.

We are now in a position to define the rules needed to handle constituent coordination in English:
As can be seen from (6.9), our rules correctly provide for the facts about English coordinate surface structure noted by Ross (1967: 90-91), namely that the coordinating word always forms a constituent with the immediately following conjunct and is not simply a sister of all the conjuncts. Indeed, given our universal characterization of possible coordination rules in (6.4) above, there is no way it could be a sister of all the conjuncts.
7. Metarules

A generative grammar typically gives a recursive definition of the sentences (and of their associated structural descriptions) of a language. It does not seem to have occurred to linguists until recently that it might be possible to give an inductive definition of the set of rules in the grammar. Such an inductive definition can be seen as a grammar for the grammar. This "hypergrammar" can express generalizations about the language generated by the grammar that are not themselves expressible in the latter. In this paper, and in Gazdar (forthcoming), Gazdar, Pullum and Sag (1980), crucial use is made of what we refer to as metarules. These can be seen as clauses in the inductive definition of the grammar. Consider, by way of analogy, how the syntax of propositional calculus is standardly given. One begins by listing or enumerating a set of (atomic) sentences, and then one says "if A is a sentence, then A is a sentence" and "if A and B are sentences then A A B is also a sentence" and so on.

We can formulate a grammar for the grammar of a natural language in much the same way. We begin by listing a set of (atomic) rules, and then we say things of the form "if R is a rule of format R, then F(R) is also a rule, where F(R) is some function of R". We will refer to such statements as metarules. Metarules in this sense, have a precursor in the shape of Vergnaud's (1973) "lexical transformations" (subsequently developed in Roepke and Siegel (1978)): these are statements of the following form: "if a lexical item L has a subcategorization frame F of format F, then L is also to be assigned a subcategorization frame T(F), where T(F) is some function of F".

Consider, simply by way of example, how one might formulate a metarule to provide one with rules to generate sentences in which an auxiliary verb preceded the subject, given only that one had listed a whole lot of rules for generating V Ps with auxiliary heads. We could say something like this:

(7.1) For every rule in the grammar of the form:

\[ \{V \quad V' \quad [\alpha] \} \]

add a rule of the form:

\[ \{V \quad V' \quad [\alpha] \} \]

Thus for every phrase structure rule that can expand a tensed auxiliary V as a tensed auxiliary V followed by a V, there is a corresponding rule expanding V as V followed by a complement V which carries exactly the features carried by the complement V in the input rule. This embedded V will itself be expanded by the normal sentence expansion rule given at (4.3) above, but repeated here for convenience:

(7.2) \( \langle 1, \{V \quad N \quad V', \quad V'(n') \} \rangle \)

The HFC ensures, of course, that any features on V in rule 1 get carried onto the daughter V.

The metarule in (7.1) can be expressed more succinctly using the following notation:

(7.3) \[ \{V \quad V' \quad [\alpha] \} \quad \Rightarrow \quad \{V \quad V' \quad [\alpha] \} \]

\[ \{V \quad V' \quad [\alpha] \} \]
This notation suppresses certain conventions. The output rule of a metarule is identical to the input rule except in respect of those items specifically changed by the metarule and changes consequent upon those changes. In particular the rule number of the input remains unchanged, consequently subcategorization gets carried over from input to output. Feature specifications also will be carried over unchanged (unless the metarule itself changes them). Thus the metarule in (7.3) will map the rule shown in (7.4a) into the rule shown in (7.4b) and the latter will allow us to generate the auxiliary initial sentence shown in (7.5):

\[
(7.4) \ a. \ \langle 36, [\overline{V} \ V (\text{BSE})] \rangle, \ldots
\]

\[
b. \ \langle 36, [\overline{V} \ V (\text{BSE})] \rangle, \ldots
\]

where \( V(36) = \{\text{can, may, must, will,} \ldots \} \)

\[
(7.5)
\]

The notation employed in (7.3) makes metarules look suspiciously like transformations but appearances here are deceptive: a transformation maps trees into trees whereas a metarule maps rules into rules. If one adds transformations to a CF-PSG then (i) one is employing two quite distinct rule types, (ii) one completely changes the expressive power of one's theory, and (iii) one ends up with a grammar that assigns at least a pair of structural descriptions to each string generated. By contrast, if one adds metarules to a PS grammar then one merely enlarges, in a rule-governed way, the set of PS rules one is employing, but the overall grammar itself remains PS.

Of course, it is not sufficient merely to list syntactic metarules like (7.3). One has also to say something about the semantics of the rules that result. Each syntactic metarule needs to be given a semantic counterpart showing how the semantics for the new rule can be arrived at as a function of the semantics of the input rule and/or the constituents of the output rule. Thus the full version of (7.3) ought to look something like this:

\[
(7.6) \ \langle [\overline{V} \ V (\text{BSE})], \ \lambda \overline{\alpha}'(\overline{\beta}') \rangle \quad \Rightarrow \quad \\
\langle [\overline{V} \ V (\text{BSE})], \ \lambda \overline{\alpha}'(\overline{\beta}') \rangle
\]

We require that metarules be finitely specifiable. The only variables permitted in the structural analysis (to borrow the transformational terminology in an obvious manner) are abbreviatory ones, that is variables which range over a finite subset of \( (V_N \cup V_L)^* \). Adherence to this requirement ensures that closing
the grammar under some set of metarules will not result in an infinite set of rules being produced.

A metarule can replace the transformation known as "particle movement":

\[(7.7)\] \[
<([\overline{V} \ V \ N \ Pt \ XI], \ \overline{\beta}) \ \rightarrow \\
<([\overline{V} \ V \ Pt \ N \ XI], \ \overline{\beta})
\]

\[-\text{PRO}]

Likewise the clause-bounded "clitic preposing" transformation standardly invoked for Romance languages:

\[(7.8)\] \[
<([\overline{V} \ V \ N \ XI], \ \overline{\beta}) \ \rightarrow \\
<([\overline{V} \ N \ V \ XI], \ \overline{\beta})
\]

And metarules will allow the grammars of VSO languages to employ the category VP (cf. Dowty 1978: 112):

\[(7.9)\] \[
<([\overline{V} \ V \ X], \ \overline{\beta}) \ \rightarrow \\
<([\overline{V} \ V \ N \ X], \ \overline{\beta}(N))
\]

Anderson and Chung (1977: 22-24) provide clear evidence for the existence of a VP constituent in the surface structure of certain Breton sentences, and a case can also be made for the reality of VP-like constituents in Modern Irish (McCloskey, 1980).

In the next section we consider how metarules can provide an analysis of the passive construction.

8. Passive

Following the arguments of Keenan (1980), we take passive to be a phrasal rather than a sentential or lexical operation. A passive VP differs syntactically from the corresponding English active VP in at least two and at most three ways: (i) the verb is morphologically marked, (ii) the direct object NP is not present in the VP, and (iii) there may be a by-PP present. Everything else remains the same. Such a construction is readily susceptible of analysis by metarule:

\[(8.1)\] \[
<([\overline{V} \ V \ N \ X], \ \overline{\beta}(N)) \ \rightarrow \\
<([\overline{V} \ V \ X \ (\overline{P}), \ \lambda\overline{\beta}(\overline{\beta}(P))) >
\]

\[-\text{TRN} \ \text{PAS} \ \text{Bx}]

In words: for every active VP rule which expands VP as a transitive verb followed by NP, there is to be a passive VP rule which expands VP as V followed by what, if anything, followed the NP in the active VP rule, followed optionally by a by-PP. As discussed in the preceding section, the rule number is held constant by convention so the subcategorization feature on V will be the same for both the input and the output rule. The PAS feature will get carried onto the V in the passive VP in virtue of the Head Feature Convention discussed in section 2 above, and it will determine the passive morphology on the expansion of V. Each provides a number of arguments for \([\overline{V} \text{PAS}])\), that is 'a syntactic category of passive verb phrases, distinct from any other category in English' (1980a: 315).

The semantic manipulation in (8.1) substitutes an NP intension type variable for \(N\) in the VP translation, applies the
resulting function to the agent $\overline{P}'$, and abstracts into the resulting open sentence to produce a function with the sort of VP-type meaning we require. In the absence of a by-PP, the Optional Argument Convention discussed in section 5 above ensures that we get the existential quantification appropriate to agentless passives. Recall that the semantics of by proposed in (4.7) above ensures that the $\overline{P}$(by) has an NP-type meaning.

Passive VPs are introduced by the following phrase structure rules:

\[(8.2) <\neg \overline{V}, \overline{V}, \lambda \overline{F}(\neg \overline{V}((\neg \overline{P})))>\]

where V(37) can only be be.

\[(8.3) <\neg \overline{V}, \overline{V}, \overline{V}'>\]

where V(38) can only be get.

\[(8.4) <\overline{V}, \neg \overline{N}, \overline{V}, \lambda \overline{F}((\neg \overline{V}(\neg \overline{N})))>\]

where V(39) = \{get, have, see, hear, ...\}

To see how (8.1) works out in practice, we exhibit its output with respect to three of the VP rules given in section 5 above.

\[(8.5) <\overline{V}, \overline{V}, \lambda \overline{F}(\overline{V}(\overline{P}(\overline{P}))))>\]

\[(8.6) <\overline{V}, \overline{V}, \lambda \overline{F}(\overline{V}(\overline{V}')(\overline{P}(\overline{P}')))>\]

\[(8.7) <\overline{V}, \overline{V}, \lambda \overline{F}(\overline{V}(\overline{V}')(\overline{P}(\overline{P}'))>\]
The door was closed.

The structure in (8.13a) will be assigned the meaning of the door was not open and (8.13b) will be assigned the meaning of the door was closed by some entity.

Our analysis taken together with the rule schema for coordination given in section 6 above, predicts that both the following sentences (from Bach 1980a: 321) have two structural descriptions:

(8.14) John was attacked and bitten by a vicious dog.

(8.15) John was attacked and bitten.

The first example is assigned these structures:

b. The door

37
AUX
FIN

V
[PAS]

was

V
[PAS]

The door

37
AUX
FIN

V
[PAS]

was

V
[PAS]

closed
Following the arguments of Wasow (1977), Lightfoot (1979) we are assuming that not all passive-like expressions arise in virtue of a unitary syntactic mechanism. Thus the expressions in (8.11) are Aₐs, not instances of V(PAS), given the rules in section 5:

(8.11)  
a. known to the police.
b. attracted to Kim.
c. uninhabited by man.
d. unloved by Kim.

And a sentence like (8.12) will be assigned two topologically similar but categorically distinct structural descriptions as shown in (8.13):
The second structure induces the reading where the dog both bites and attacks, whereas the first structure induces a reading which is noncommittal with respect to the agency of the attack. Example (8.15) will also have two structural descriptions as shown in (8.17):
in order to generate the following examples:

(8.19)  a. John was said to be in Rome
b. Mary is reputed to be a genius.
c. John was seen to have left.
d. Kim was said to be like Sandy.
e. Sandy was made to like Kim.

Examples (8.19a–c) are due to Bach (1980a: 328).

Here the second structure induces the reading where the same agent was responsible for both attacking and biting, whereas the first structure induces a reading which leaves it open as to whether the same agent was responsible.\(^{11}\)

Not all passive Vφ rules arise as the product of the metarule in (8.1). The grammar of English will need to list at least the following rule:

(8.18)  \[ <40, \{V, V, \ldots \}, \lambda P(V'(\langle V'(P)\rangle)) > \]

where \(V[40] = \{\text{repute, rumor, see, say, make, ...}\}\)
9. Unbounded dependencies

Bresnan has claimed that unbounded syntactic dependencies cannot be adequately described even by context-sensitive phrase structure rules, for the possible context is not correctly describable as a finite string of phrases’ (1978: 38). Despite accurately reflecting the metagrammatical intuitions of a significant body of linguists, this claim is a complete non-sequitur. The fact that natural languages allow dependencies whose domain is not describable as a finite string of phrases has no bearing on the descriptive adequacy of any familiar class of grammars. Even finite state grammars allow for the description of certain kinds of unbounded dependency.  

And phrase structure grammars can handle unbounded dependencies in an elegant and general way provided that we exploit the resources offered by a complex symbol system and by the possibility of making statements about the set of rules that the grammar may employ.

Let $V_N$ be the set of basic category symbols (i.e. the set of all nonterminal symbols standardly used). Then we define a set $D(V_N)$ of derived categories as follows:

$$D(V_N) = \{ \alpha/\beta : \alpha, \beta \in V_N \}$$

Suppose counterfactually that S and NP were the only basic categories; then the set of derived categories would consist of $S/S$, $S/NP$, $NP/NP$, and $NP/S$. This notation is reminiscent of categorial grammar but, despite a tenuous conceptual link, these derived categories are not to be interpreted in the way categorial grammar prescribes. The intended interpretation is as follows: a node labelled $\alpha/\beta$ will dominate sub-trees identical to those that can be dominated by $\alpha$, except that somewhere in every subtree of the $\alpha/\beta$ type there will occur a node labelled $\beta/\beta$ dominating a resumptive pronoun, a phonologically null dummy element or the empty string, and every node linking $\alpha/\beta$ and $\beta/\beta$ will be of the form $\sigma/\beta$ for some category $\sigma$. Intuitively then, $\alpha/\beta$ labels a node of type $\alpha$ which dominates material containing a hole of type $\beta$ (i.e. an extraction site on a movement analysis). So, for example, $S/NP$ is a sentence which has an NP missing somewhere.  

Defining a new set of syntactic categories is not of itself sufficient, of course, to ensure that the trees in which they figure have the property just described: we need, in addition, a set of rules to employ them.

What we have to do is define a set of rules each of which expands a derived category just as the corresponding basic rule would have done for the basic category, except that exactly one of the dominated categories is now paired with the same hole-indicating category as is the dominating category. The set of such rules will consequently allow the hole information to be "carried down" the tree.

Let $G$ be the set of basic rules (i.e. the set of rules that a grammar not handling unbounded dependencies would require). For any syntactic category $\beta$, there will be some subset of the set of the nonterminal symbols $V_N$ each of which can dominate $\beta$ according to the rules in $G$. Let us call this set $V_\beta$ ($V_\beta \subseteq V_N$). Now, for any category $\beta$ ($\beta \in V_N$) we can define a (finite) set of derived rules $D(\beta, G)$ as follows:  

$$D(\beta, G) = \{ [\alpha/\beta^{\sigma_1} \ldots \sigma_i^{\beta} \ldots \sigma_n] : \beta^{\sigma_1} \ldots \sigma_i \ldots \sigma_n \in G \}$$
An example of the application of (9.2) should make this clearer. Suppose that the set $G$ of basic rules looks like this:

\begin{align*}
(9.3) & \quad a. \ (\overline{\overline{N}} \ V \ N) \\
& \quad b. \ (\overline{V} \ V \ \overline{V}) \\
& \quad c. \ (\overline{V} \ V \ \overline{N}) \\
& \quad d. \ (\overline{N} \ V \ \overline{p}) \\
& \quad e. \ (\overline{N} \ V \ N) \\
& \quad f. \ (\overline{V} \ V \ N \ \overline{p}) \\
& \quad g. \ (\overline{N} \ V \ \overline{p}) \\
& \quad h. \ (\overline{p} \ \overline{p}) \\
& \quad i. \ (\overline{p} \ P \ N) \\
\end{align*}

Then the set $D(N, G)$ will look like this:

\begin{align*}
(9.4) & \quad a. \ (\overline{\overline{N}N} \ V) \ N \ \overline{V}/N) \\
& \quad b. \ (\overline{V}/N) \ V \ \overline{N}/N) \\
& \quad c. \ (\overline{V}/N) \ V \ N \ \overline{V}/N) \\
& \quad d. \ (\overline{V}/N) \ V \ \overline{p}/N) \\
& \quad e. \ (\overline{V}/N) \ V \ N \ \overline{p}/N) \\
& \quad f. \ (\overline{V}/N) \ V \ N \ \overline{N}/N) \\
& \quad g. \ (\overline{V}/N) \ V \ N \ \overline{p}/N) \\
& \quad h. \ (\overline{V}/N) \ V \ P/N) \\
& \quad i. \ (\overline{V}/N) \ P \ N/N) \\
\end{align*}

and the set $D(P, G)$ will look like this:

\begin{align*}
(9.5) & \quad a. \ (\overline{V}/N) \ N/P \ \overline{V}, \ (\overline{V}/P) \ N \ \overline{V}/P) \\
& \quad b. \ (\overline{V}/N) \ V \ \overline{V}/P) \\
& \quad c. \ (\overline{V}/P) \ V \ \overline{V}/P) \\
& \quad d. \ (\overline{V}/P) \ V \ \overline{p}/P) \\
& \quad e. \ (\overline{V}/P) \ V \ N/P) \\
& \quad f. \ (\overline{V}/P) \ V \ N/P \ \overline{p}) \ (\overline{V}/P) \ V \ \overline{N} \ \overline{p}/P) \\
& \quad g. \ (\overline{N}/P) \ N/P \ \overline{p} \ \overline{p}/P) \ (\overline{N}/P) \ N \ \overline{p}/P) \\
& \quad h. \ (\overline{p}/P) \ P \ \overline{p}/P) \\
& \quad i. \ (\overline{p}/P) \ P \ \overline{N}/P) \\
\end{align*}

Derived rules have no special lexical or semantic properties. Thus, all derived rules will have the same rule-numbers, the same subcategorization properties and the same semantic translations as the basic rules from which they derive. Consequently they do not need to be separately listed or separately specified since everything about them is predictable from (9.2) taken together with the basic rules.

In addition to derived rules, we also need linking rules (these will be a subset of the basic rules) to introduce and eliminate derived categories. For the majority dialect of English (British or American) we need only the following rule schema to eliminate derived categories:

\begin{align*}
(9.6) & \quad <41, \ \{a/a \ t\}, \ h_{\alpha} > \quad \text{where } a \in V_{N}
\end{align*}

Here $h_{\alpha}$ (mnemonic for 'hole') is an distinguished variable
ranging over denotations of type $\alpha$ (i.e. NP denotations if $\alpha = \text{NP}$, PP denotations if $\alpha = \text{PP}$, etc.). And $t$ is a dummy element postulated solely for phonological reasons (i.e. it will serve to block contraction), it serves no semantic function ($h_{\alpha}$ is the variable, not $t$), and for other dialects or languages we could replace $t$ with the empty string - (which would have no phonological effects) or with a proform. The rules ensure that $t$ is placed in precisely those locations where contraction-inhibiting phonological effects have been noted, and consequently the analysis faces none of the difficulties faced by the analyses criticised in Postal and Pullum (1978).

The apparatus developed above can be used to handle all constructions involving an unbounded dependency. However, since exactly the same principle is involved in every case, it will suffice here to illustrate its application by reference to just two constructions, namely topicalization and English free relatives.

At least one recent linguistics textbook (Perlmutter and Soames 1979: 230-231) argues, quite fallaciously, that the facts of English topicalization are beyond the descriptive powers of a phrase structure grammar. But the phrase structure rule schema in (9.7), taken together with the apparatus developed above, exactly captures these facts:

\[(9.7) \quad <42, \lambda t \in V/\alpha, \lambda b_{\alpha}(V/\alpha)'(\alpha')>\]

where $\alpha = \overline{x}$.

This schema will induce structures like those shown in (9.8)-(9.11):
Note how the derived rules ensure that there is at most one hole in the complement $\bar{V}/a$ and at least one hole in the complement $\bar{V}/\bar{a}$. The semantics given in (9.7) binds the distinguished variable $b_{\bar{a}}$ in the translation of $\bar{V}/a$ and applies the resulting function to $a$: this has the effect of ensuring that topicalized sentences are truth-conditionally synonymous with their un-topicalized counterparts.

Notice also how the schema in (9.7) gets the pied-piping facts exactly right. Thus we generate (9.12a) but not (9.12b):

(9.12)  

a. To John, Julie gave a copy of her book.

b. *To John, Julie gave a copy of her book to.

These examples are due to Iwakura (1980) who shows that the wh-movement analysis of topicalization proposed by Chomsky (1977) has the unfortunate property of predicting that (9.12b) is grammatical and that (9.12a) is ungrammatical.

We turn now to the English free relative construction:

(9.13)  

\[<\text{[WH ever]}\text{[S/a1]}, \ldots>, \text{where } \alpha = \overline{\text{x1}}/\overline{\text{V}}.\]
This analysis of free relatives is due, in effect, to Bresnan and Grimshaw (1978), and the structures induced by (9.12) will be virtually isomorphic to those proposed by them. The difference lies in the fact that the present framework can dispense with their rule of Controlled Pro Deletion (ibid: 370) and the coindexing convention associated with it. In (9.14) and (9.15) below we exhibit the structures that (9.13) will assign to Bresnan and Grimshaw's examples (119) and (120) (ibid: 357).

\[(9.14)\]
Note that our rule for topicalization given at (9.7) above, taken together with (9.13) makes exactly the same predictions as Bresnan and Grimshaw's analysis. I repeat their examples (their (121), ibid: 358) below:

(9.16)  
(a) Whatever town you live in I'll live in.
(b) *Whatever town you live, I'll live in.
(c) In whatever town you live, I'll live.

The major advantage of the present analysis over that proposed by Bresnan and Grimshaw, apart from its basis in a more constrained linguistic metatheory, is that it automatically predicts the badness of examples like (9.17):

(9.17)  
(a) *I will live in whatever town you live near the sea.
(b) *I will live in whatever town you live in Brighton.

The only ways Bresnan and Grimshaw could account for such examples are (i) appeal to as yet unspecified interpretive rules, or (ii) ordering Controlled Pro Deletion before lexical insertion, or (iii) introducing a filter to chuck out strings containing certain types of unindexed WH phrase. But the facts follow from (9.13) since derived categories must, of necessity, dominate "holes".

This is a quite general advantage of the present framework over analyses employing controlled deletion for unbounded dependencies.

Note also that the familiar c-command condition on the relation between the controlling expression and the controlled "hole" simply follows from the use of context-free phrase structure rules to generate this kind of construction: in a rule of the form $\{a, b/a\}$ or $\{a, b/a, a\}$, $a$ cannot help but c-command the hole in $b/a$. But the c-command condition on binding has to be separately stipulated in theories which employ transformations like Controlled Pro Deletion or wh-movement to handle unbounded dependencies.
10. Islands

As Brame (1978: 100 ff) has pointed out, some island constraints account for data whose problematic status is merely an artefact of the postulation of movement rules. He considers the following examples (from Chomsky 1973: 88):

(10.1) a. I believe the dog is hungry.

b. I believe the dog to be hungry

(10.2) a. The dog is believed to be hungry by me.

b. The dog is believed to be hungry by me.

If passives derived via NP-movement, then both (10.2a) and (10.2b) will be generated in the absence of any constraints. Imposition of the Tensed S condition will block (10.2a). But, in frameworks such as Brame’s or that of the present paper, which abjure all movement rules, there is no problem with this data, and consequently no need to invoke a Tensed S condition to account for it.”

Another example of an artefactual island constraint is Ross’s (1967) Coordinate Structure Constraint (CSC). Gazdar (forthcoming) shows in detail how all the phenomena covered by the CSC, together with the across-the-board violations of it, simply follow from the analysis of unbounded dependencies developed in section 9 above, and the rule schemata for coordination given in section 6 above. Briefly: a cannot be coordinated with a/β since a and a/β are not the same syntactic category. But a can be coordinated with α, and a/β can be coordinated with a/β.

Thus:

(10.3) a. The person that Kim loves and Lee hates

Kim irritates me. \( (\overline{V/N} \& \overline{V}) \)

b. The fact that Kim loves Lee and Lee hates Kim irritates me. \( (\overline{V/N} \& \overline{V}) \)

c. The person that Kim loves and Lee hates irritates me. \( (\overline{V/N} \& \overline{V/N}) \)

Numerous island constraints other than the CSC have been proposed in recent years. Unfortunately, few if any of them are as resilient to counterexample as the CSC is. Nevertheless, it is perhaps worth pausing to consider how one might formalize constraints once thought only applicable to movement rules in the framework of a theory in which no movement at all takes place.”

It will be helpful to repeat the definition of the derived rule set at this point:

(10.4) \( D(\beta, G) = \{ (\alpha/\gamma_1 \ldots \gamma_n/\beta_1 \ldots \beta_n) : \alpha_{\gamma_1} \ldots \gamma_n \in \sigma [G \& \ldots \& G] \} \)

We can formalize certain island constraints, if we wish, simply by stipulating that certain types of derived rule are not employed by a language (or by any language, if the constraint suggested is intended as a universal). Suppose we wanted to impose the A-over-A constraint. Then we could add a condition that \( a \neq \beta \) to (10.4). This would have the effect of preventing the creation of any derived rules of the form shown in (10.5):”

(10.5) \( *([a/\alpha \ldots]) \)

Ross’s (1967: 114) Left Branch Condition would involve prohibiting rules of the form shown in (10.6):”

(10.6) \( *(\overline{N/N} \ldots) \)
And his Complex NP Constraint (CNPC) would look like this:

(10.7) \[ *(N/a \overline{N} \ldots \overline{E/a} \ldots) \]

Bresnan's (1976a) generalization of CNPC, the Complex Phrase Constraint, would look something like this:

(10.8) \[ *(\overline{E/a} \overline{\beta} \ldots \overline{I/a} \ldots) \]

And Horn's (1974) NP Constraint like this:

(10.9) \[ *(\overline{N/a} \ldots \overline{E}) \]

The island constraints mentioned in (10.5)-(10.9) have in common their locality; that is to say that they can all be stated by reference to a particular class of rules. Equivalently, they could be formulated as filters on tree sets which ban all trees exhibiting certain local mother-daughter relationships. But some island constraints discussed in the recent literature (e.g. Subjacency) have not had this local character. Fodor (1980), developing an idea due to Koster (1978), proposes that nonlocal constraints on unbounded dependencies be stated on projection paths, where the latter are defined to be maximal subpaths of paths from root to terminal such that every node-label in the subpath is of the form \(a/\beta\) where \(\beta\) is a constant and \(a\) varies.\(^3\)

Thus Maling and Zaenen (1980) have proposed that the complex island facts in Italian discussed by Rizzi (1978) can be elegantly captured by the tree filter on projection paths shown in (10.10):

(10.10) Throw out all trees containing a projection paths of the form \(\ldots \overline{Q/\beta} \ldots \overline{E/\beta} \ldots \)

It is worth remarking that imposition of tree filters of the type shown in (10.10) has no effect on the generative capacity of the class of grammars employing them; the output of a CF-PSG filtered by a finite list of such tree filters will be a CFL. This can be simply proved by reference to finite state tree automata but we shall not pursue that here.

The definition of derived category as given in the previous section only allows for categories of the form \(a/\beta\) where \(a, \beta \in V_N\) and \(V_N\) simply consists of familiar complex symbols like \(\overline{N}, \overline{V}, \overline{V}\), etc. A consequence of this definition is that the formalism has the effect of imposing the \(\text{wh}-\text{island constraint}:\) there can be no more than one unbounded dependency into a constituent that has a hole in it. This is arguably correct for English although there are a number of problematic examples even here. But it is clearly incorrect for such languages as Hebrew (Reinhart 1980), Italian (Rizzi 1978), Modern Irish (McCloskey 1979), and the Scandinavian languages (see Engdahl (1980a, 1980b) and Maling and Zaenen (this volume, 1980) for extensive discussion).

Suppose we generalize the definitions of derived category (and derived rules) to permit categories of the form shown in (10.11):

(10.11) \(a/\beta_1, \ldots, \beta_n\)

Two possibilities then present themselves. We could restrict \(n\) to be some small finite number the exact value of which would constitute a parameter of permissible language variation. Thus, arguably, for Palauan, Polish and Russian we have \(n = 0\) (i.e. no unbounded dependencies whatsoever are permitted), for English \(n \leq 1\), for Icelandic \(n \leq 2\) (Maling and Zaenen 1980), while for
Swedish the upperbound value for $n$ must be at least 3 (see the data in Engdahl 1980a). For any finite value for $n$ the resulting grammar will be CF. Engdahl (1980b) has, however, developed an argument against adopting any finite bound on $n$ by analogy with the argument of Miller and Chomsky (1963) against restricting multiple center embedding.

Suppose we put no restriction on $n$ for a language like Swedish. This has the effect of taking us out of the class of CF-PFG and into the class of indexed grammars (Aho 1968).

Unrestricted indexed grammars generate the indexed languages and the latter are a proper superset of the CFLs (e.g. they include $\{a^n b^n c^n\}$) and a proper subset of the context-sensitive languages. In a recent textbook on formal languages, Hopcroft and Ullman remark that 'of the many generalizations of the context-free grammars that have been proposed, a class called "indexed" appears to be most natural, in that it arises in a wide variety of contexts' (1979: 398).

However, there is, as yet, no linguistic evidence to suggest that we need the full power of the indexed grammars. Thus, even assuming that Swedish permits arbitrarily many unbounded dependencies into a single constituent, there is no reason to believe that the stringset consisting of all and only the grammatical sentences of Swedish is anything other than a CFL. Consequently, a move towards the adoption of indexed grammars needs to be accompanied by specific proposals for constraining the expressive power of such devices in linguistically relevant ways.

FOOTNOTES

1. As Wasow has pointed out, the goal of constraining the class of languages generated by permissible grammars is 'inherently more ambitious' (1978: 82) than the goal of constraining the class of permissible grammars: 'although the class of grammars may be restricted without changing the class of languages, the converse is, of course, not true; that is, it is not possible to limit the class of languages without limiting the class of grammars' (ibid: 82).

2. Our idealization here is no more implausible than that made in Hamburger and Wexler (1973) where the learner is assumed to have access to the deep structure of each grammatical string presented.

3. However, we would not want to go as far as to claim that 'if all of English could be given in a context free (sensitive) form the learnability problem would be solved' (Keenan, 1979a: 00).

4. Unfortunately, Chomsky's discussions (1965: 210-211, 1966: 40-50) of Herman's proposals are spoilt by an irritating terminological imperialism. Consequently, he never responds to Harman's fundamental point, which I take to be this: 'The critique of phrase structure consists in the construction of a formal model of phrase-structure theory and the demonstration that this is inadequate as a complete theory of grammar. The defense of phrase structure consists in repudiating the model of constituent structure, as Chomsky defines it, and for replacing it with a model which obviates the original criticisms' (1963: 610-611).

5. See Harris (1951), Chomsky (1970), Selkirk (1972), Bresnan (1976b), Emonds (1976), and Jackendoff (1977) for various proposals.

6. E.g. WH, PRO (as in Chomsky 1977), LOC, TEMP, DIR (Bresnan and Grimshaw 1978), agreement features (e.g. Wasow 1975: 374), etc.

8. For arguments that agreement is phrasal rather than lexical see Keenan (1979a: 00-00).

9. The preceding discussion constitutes a refutation of the claim that verb agreement instances a 'grammatical process that is beyond the generative capacity of ... the CF-PSG' (Grinder and Elgin 1973: 65).


11. We will ignore other possible expansions of $\overline{P}$ and $\overline{F}$ here.

12. I am grateful to John Lyons for reminding me of the relevance of these facts.

13. I am indebted to Ewan Klein and Barbara Partee for some painstaking advice on the semantic component of the rules given.

14. We will adopt the convention of omitting feature specifications in the semantic part of a rule.

15. Note that lexical items can be multiply listed, thus eat is a V[7] as well as a V[8]. But $(i_\nu \text{ eat})'$ is a one-place predicate, whereas $(i_\nu \text{ eat})'''$ is a two-place predicate.

16. Following Oehrle (1975) and Dowty (1978), we treat the dative alternation in English as a purely lexical matter.

17. This rule only deals with the construction where seem and appear have full lexical subject rather than dummy it or there. For the definitive treatment of dummy subject dependencies, see Sag (this volume).

18. This rule is included only for the sake of completeness, see Gardar, Pullum and Sag (1980) for a more general analysis of the complements of copular be.

19. We will ignore other possible expansions of $\overline{A}$ here.

20. Following Wasow (1977), we assume that passive adjectives like closed are arrived at by lexical rule. Non-adjectival passives are discussed in section 8, below.

21. On regular expressions, and the notation used to characterize them, see, e.g., Hopcroft and Ullman 1979: 28-29.

22. This is trivial to prove, and it follows in any case as a corollary of a more general theorem proved by Langendoen (1976).

23. Although the strings generated by these regular CF-PSG rules can be generated by ordinary CF-PSG rules, the trees that they induce cannot. So one can reasonably ask whether employing rules like those in (6.4) is going to make parsing of CFLs hard (assuming, that is, that the parse tree we want is the one that would be assigned by (6.4)). The answer is no: Earley points out that his algorithm 'has the useful property that it can be modified to handle an extension of context-free grammars which makes use of the Kleene star notation' (1970: 101) (Note that $\alpha^* = \alpha\alpha^*$).

25. This terminology is due to Langendoen (1976). The idea of using a grammar to generate one's grammar originates with van Wijngaarden (1969) who used the technique to give a perspicuous syntax for the computer language ALGOL68. A good introduction to his work can be found in Clevseland and Uzgalis (1975). Janssen (1979) employs a van Wijngaarden-style two-level grammar to define a generalization of Montague's FTQ syntax.

26. This metarule, and the auxiliary $\bar{V}$ rules assumed below, are taken from Gazdar, Pullum and Sag (1980) from which further details should be sought.

27. A desirable consequence of this requirement is that coordination rules can never feed metarules.

28. I am grateful to Tom Wasow for making me aware that some such restriction on grammar and metarule systems was required. Aravind Joshi has pointed out to me that allowing a single non-abbreviatory variable in the structural analysis of metarules would open the way to PS grammars with infinite sets of rules, but would not result in any of those grammars inducing non-context-free languages. Non-context-free languages can only result when two or more non-abbreviatory variables are permitted.


30. Keenan's paper mysteriously ascribes various properties to the analysis of passives proposed in Gazdar (forthcoming). This is mysterious since the closest thing to an analysis of passives in Gazdar (forthcoming) is the following tautology: 'if there are no transformations then there is no passive transformation'. Pace Keenan, it is does not follow from this, or from anything else in that paper, that 'there are two passive operations in English' (ibid: 212) or that 'adjective phrases are derived from TVPs and the passive VPs are formed by an independent rule which adjoins copular verbs like be with adjective phrases' (ibid: 213).

31. The phrasal treatment of passive given above is similar in certain respects to the lexical analysis, given by Bresnan (1978). However, as Keenan (1979b) has pointed out, her analysis incorrectly requires the presence of the copula before the passive V, her semantics becomes incoherent when there is a quantified NP in the by-PP, and she does not allow for coordinate structures like those shown in (8.16b) and (8.17b).

32. Thus, for example, it is trivial to arrange for all the sentences generated by some finite state grammar to end with the same terminal symbol that they began with, even when there is no upper bound on the length of sentence generable by the grammar.

33. At least, not in general. In the case of the rule for topicalization at (9.7) below, it happens that S/α combines with an α to form an S, exactly as they would on a categorial interpretation of the notation.

34. Aravind Joshi has pointed out to me that a notation having the same interpretation as derived category symbols is to be found in Harris's work, thus: 'S-N or "S excising N" indicates an S string from which one N has been cut out' (1962: 11n). However, Harris appears to use such labels simply as a descriptive device for the analysis of texts, rather than as an essential component of an explicit generative grammar for English. More recently, c.f. Baker (1978: 113), and Hellan (1977: 128).

35. The definition in (9.2) is a metarule, albeit a very general one, and could just as well be stated in the abbreviatory notation for metarules introduced in section 7 above. Thus:

$$\langle a, x \sigma y \rangle, \underline{3} \Rightarrow$$

$$\langle a/\beta, x \sigma/\beta y \rangle, \underline{3} \rangle$$

where $a, \sigma \epsilon V_\beta \subset V_N^\prime$. 
36. See Gazdar (1980b) for comparative "deletion" and "sub-deletion" dependencies, and Gazdar (forthcoming) on relative clauses and constituent questions.

37. I am grateful to Geoff Pullum for drawing this passage to my attention.

38. The generality of a here is slightly spurious: as Bresnan and Grimshaw demonstrate, when a = PP we have to restrict it to locative and temporal PPs. Note also that the use of the ever feature on the head is simply to get the morphology of free relatives right (see Bresnan and Grimshaw 1978: 338-339). Its presence does not amount to the obviously false claim that the -ever suffix has to appear in free relative heads.

39. Believe is introduced by the V rules [\(V\ V\ N\ V\ ) (see (5.8) and (5.15), above). Only [\(V\ V\ N\ ) the latter can be input to the passive metarule (8.1) since the former does not mention a post-verbal N. Consequently, the only passive rule for believe will be [\(V\ V\ N\ ) This will admit (10.2b) but not (10.2a). (INF) (by]

40. The question of how one formalizes island constraints is, of course, distinct from the question of how one motivates them. The remarks made here only address the question of formalization, a question that is rarely addressed in the literature.

41. Remember that [\(a/m\ t\) is not itself a derived rule and so it will not be forbidden by the addition of a a≠β clause to (10.4).

42. See Gazdar (forthcoming) for detailed discussion of a generalization of (10.6) referred to there as Generalized Left Branch Condition.

43. Definition is not quite this simple, however, since we need to ignore pseudo-paths caused by the contiguity of two genuine projection paths. For example, in a case like that shown in (i):

(i)

```
\[\begin{array}{c}
Q \\
N \\
V/N \\
who \\
N \\
N/N \\
V/N \\
im \\
V \\
V/N \\
\end{array}\]
```

It is straightforward, though tedious, to define the notion of projection path so as to exclude the path from the highest V/N to the lowest N/N in (i).


Gazdar, G.J.M. (forthcoming) Unbounded dependencies and coordinate structure. To appear in LI.


Ladusaw, W. (1979) Polarity sensitivity as inherent scope relations. PhD dissertation, University of Texas at Austin. Distributed by UPLC.


Maling, J. & A. Zaenen (1980b) A base-generated account of Scandinavian extraction phenomena. This volume.

Marslen-Wilson, W.D. (1975) Speech shadowing and speech perception, unpublished PhD dissertation, MIT.


