2

A theory of syntactic features

1 Features in earlier generative syntax

In the theory of phrase structure grammars as standardly presented, category labels like 'S', 'NP', 'VP', 'N', etc. are monadic, which is to say that they have no internal structure and are not reducible to anything else.[1] Likewise, in pre-Jakobsonian phonology, phonemes like /g/, /k/, /p/, /s/, etc. were taken to be monadic and irreducible. In phonology, distinctive feature theory (Jakobson, Fant, and Halle 1951) replaced this view of the phoneme with one in which each phoneme was defined by reference to a set of features that might be specified positively or negatively. Under this conception, /g/, for example, comes to be understood as merely an abbreviation for, say,

$$\begin{bmatrix} +\text{segment} \\ +\text{consonantal} \\ -\text{sonorant} \\ -\text{syllabic} \\ +\text{high} \\ +\text{back} \\ -\text{low} \\ -\text{round} \\ -\text{anterior} \\ -\text{coronal} \\ -\text{nasal} \\ -\text{continuant} \end{bmatrix}$$

In syntax, Harris (1946, 1951) proposed that the relation between categories such as V and VP, and N and NP, was a systematic one that could be captured by breaking the monadic parts of speech labels into two components, namely a category type and a phrasal level. This insight was subsequently taken up in the 'X-bar syntax' suggested by Chomsky (1970) and most fully developed in Jackendoff 1977. The fact that there was considerable descriptive potential inherent in nonterminal symbols with internal complexity was first recognized by Harman (1963). Chomsky (1965, pp. 79 ff.) developed Harman's insight further by introducing into syntax the same kind of notation for features that had already been found useful in phonology.

But at this point, development in the theory of syntactic features basically stopped. Although generative grammarians continued to assume features in their descriptive apparatus, hardly any generative grammarians attempted to give syntactic features the kind of well-defined formal underpinnings that, say, the theory of phrase structure rewriting rules had. George Lakoff's 1965 dissertation (published as Lakoff 1970) was an honorable exception, but it influenced the field more toward the development of abstract deep structures and complex transformational derivations than toward appropriate exploitation of features in phrase structure description, despite the rich proposals for feature analysis that it presented.

During the seventies, the theory of features fell gradually into a state of chaos. This is clearly illustrated in the introductory texts on transformational grammar. One might expect a theory of features to deal with, among other things, the distribution of features and feature values in rules and trees; the possibilities of co-occurrence for feature specifications within a category; and the assignment of feature values on the basis of default statements in cases where no rule or general convention has assigned a value. Introductory texts on generative grammar occasionally touch on all three of these matters, but present no coherent proposals. Thus we find Akmajian and Heny (1975, p. 163) saying:

> The Reflexivization rule has the effect of placing the feature [+REFLEXIVE] on the NP node; we shall assume that by general convention this feature is then assigned to the head N of the NP . . .

This suggests a default assumption that NPs are not [+REFLEXIVE], and a principle of downward copying of feature specifications from NP nodes onto their lexical N heads. But then we find on p. 199, in the context of a discussion of number agreement features, which also appear on NP, the following statement:

> We assume that these features will have been 'spread' onto the whole subject NP from the head N.

This suggests upward copying of feature specifications from the lexical head N onto other members of the NP. Which do Akmajian and Heny actually postulate: downward copying, upward copying, both, or some nondirectional principle that has the effects of both in different cases?

Akmajian and Heny are not untypical in never even raising, let alone answering, such questions.

It seems clear enough that much of what features were doing in transformational derivations was never understood by those who made reference to them, so it is not surprising that it is not explained clearly in introductory books. Consider this passage from a discussion by Stockwell et al. (1973, pp. 47–8) of a rule that is triggered by, and then erases, a feature specification involving what we would call a terminal symbol feature (see section 4):

> The ... part of the structure change which erases the exception feature that governs the rule in the first place has no purpose except to unclutter the tree somewhat. It can probably be stated in some much more general way: e.g. a convention imposed on all rules that an exception feature is erased after doing its work – i.e. after governing some rule. The difficulty with such a convention is that one would have to take care to provide that the feature was relevant in *only one* rule; in the face of that hazard, we have erased features within each rule when we were sure they were no longer needed – and we have not been consistent in erasing them even under those circumstances.

There are some evident signs of desperation here. And it is not surprising. The business of determining feature specifications in deep structure trees and then further controlling the effect of each transformational operation on feature values leads to problems in the computation of derived structure that look quite substantial. For example, the reader may like to reflect upon how an algorithm might be devised to keep track of the legality of feature manipulations in the light of the following proposed constraint due to Emonds (1976, p. 212):

> A node B introduced into a tree by a transformational insertion T (rather than by movement or copying) can never during the derivation dominate a feature that is not defined as syntactic independent of all transformations that move into B (including T).

The initially baffling aspect of this global constraint on derivations does not dissipate as one searches the literature of transformational grammar for the formal machinery in terms of which it is supposed to be interpreted. It is an inhabitant of a twilight world in which lip-service is paid to formal precision but primitives are never supplied and definitions are never stated.

It is our intention in this chapter, and in chapter 5, to lay the foundations of a formally respectable theory of syntactic features, their values, their distribution in syntactic representations, and the universal and parochial principles that govern them. This is work that any imaginable theory of syntax for human languages will ultimately have to do, so the material of this chapter should not be seen as an exercise in generalized phrase structure grammar alone. We stress in particular that most syntactic theories, including past and present transformational theory, presuppose a theory of features covering much the same ground as the one we develop here, but have left it entirely implicit and undeveloped,[2] despite the occasionally glimpsed necessity of making appeal to it as seen in the quotations above.

2 Syntactic categories

The use of a finite set of nonterminal symbols that are composed of syntactic feature specifications does not of itself increase the expressive power of the theory of phrase structure. For reasons put very nicely by Halle (1969) with respect to phonology, there is an exact equivalence between generative systems that use complex symbols (matrices of distinctive feature specifications) and those that do not. The proof is trivial. Basically, only the way the symbols are interpreted is at issue. A nonterminal symbol $[x_1, x_2, \ldots, x_n]$, where each x_i is some feature specification, can be treated as having internal structure to which statements in the grammar can refer to capture generalizations, or it can be regarded as a calligraphically ornate representation of an atomic symbol distinct from all other symbols. Moreover, anything done by a rule referring to, say, $[x_2]$ could also be done by a rule which referred to the complete list of all complex symbols in which $[x_2]$ appeared.

Illustration in more concrete terms will make this clearer. Throughout this book we shall use the traditional categories Noun (N), Verb (V), Adjective (A), and Preposition/Postposition (P), but formally we shall treat them (following Chomsky 1970) as decomposable by means of a feature system that postulates a feature specification [+N] which only N and A have, and a feature specification [+V] which only V and A have.[3] Thus nouns are nominal but not verbal; adjectives are nominal and verbal (capturing a traditional notion of 'substantive'); verbs are verbal but not nominal and prepositions are neither verbal nor nominal. In other words, the feature specifications [±N] and [±V] group the categories informally notated, N, V, A, and P into natural classes as shown in (1).

(1)

	[+N]	[−N]
[+V]	A	V
[−V]	N	P

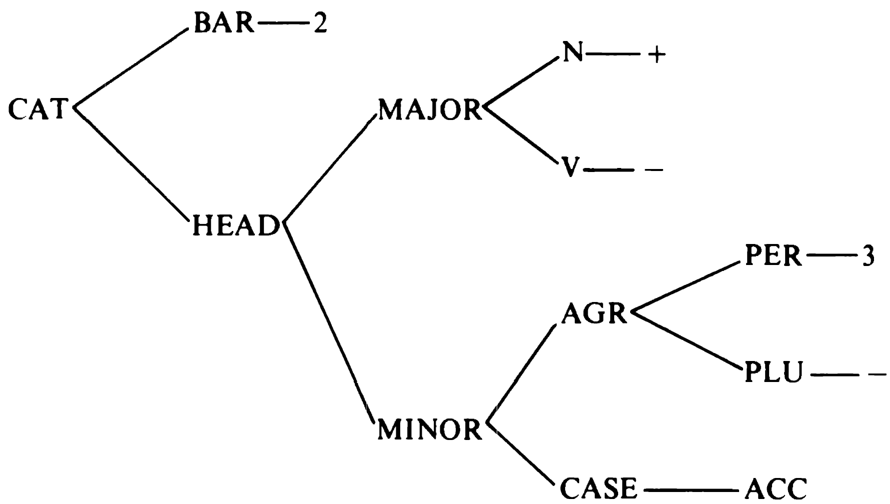
This enables us to refer to the class of, say, nouns and prepositions/postpositions simply by writing [−V]. But any statement about [−V] items could have been made without reference to features; what is claimed about [−V] would simply be stated to true of N and also true of P.

In phonology, complex symbols for phonological units are standardly taken to be sets of ⟨*feature, feature-value*⟩ pairs. (Such a pair is what we refer to as a *feature specification*.) However, a number of phonologists have proposed that phonological units should have more internal structure than this, and some syntacticians have proposed similar enrichment for the structure of syntactic categories. Thus Chomsky (1965, p. 171) implicitly assumes a hierarchical internal structure for lexical categories, and Bresnan (1976) defines categories as ordered pairs of an integer (representing bar level) and a bundle of feature specifications. More recently, Stucky (1981c) and subsequently Horrocks (1983) have proposed that languages with object agreement be analyzed by reference to complex representations involving ordered pairs of sets of agreement feature specifications. We shall likewise be assuming that categories have a significant amount of internal structure.

One particularly useful elaboration of the theory of features, of which we adopt a special version below, involves letting features take other feature specifications, or whole categories, as their values, an idea suggested, in effect, by Anderson (1977b, 1981), Kay (1979), Evans (1982), and Pollard (1982).

In Gazdar and Pullum (1982), a theory of features was developed along these lines, using graph theory to provide the basic concepts, and reconstructing feature specifications and categories as the same sort of objects, namely directed acyclic graphs obeying the single mother and unique root conditions. The advantage Gazdar and Pullum derived from the graph-theoretic approach was that whole clusters of feature specifications could be picked out in a natural way if they shared a mother node in the graph. So, for example, suppose we take a category looking like this:

(2)



This represents (partially) the label of a 3rd person singular accusative noun phrase (NP), where an NP is theoretically reconstructed as a phrase that has the category type characterized by [+N] and [−V] (like a noun) and the bar level 2. The 'is a value of' relation is read from the leaves on the right toward the root on the left. Thus 3 is a value of PER; PER is a value of AGR; AGR is a value of MINOR; MINOR is a value of HEAD; and HEAD is a value of CAT. If we want to say that the lexical head of an NP must have the same values for person, number, case, and the features N and V as the NP itself does (part of an important generalization we shall discuss in more detail as we proceed), we simply require that HEAD in the lexical category in question have the same graph-theoretic structure, node for node, as the NP has. HEAD, by virtue of its position in the graph, provides a way of pulling out a cluster of related feature specifications that must be referred to for a particular purpose.

We adopt a slightly different conception of feature and category in the present work.[4] We take feature specifications to be ordered pairs of the form ⟨*feature, feature-value*⟩, where a feature is an atomic symbol, and a feature-value is either an atomic symbol or a category. We take a category to be a set of feature specifications. In these terms, the category represented above would be just the set of feature specifications in (3).

(3) {⟨BAR, 2⟩,
⟨N, +⟩,
⟨V, −⟩,
⟨PER, 3⟩,
⟨PLU, −⟩,
⟨CASE, ACC⟩}

We will adopt various conventions for representing categories and feature specifications, and it will be useful to mention some of them now.

Following traditional practice, feature specifications will usually be enclosed in square brackets. Thus we will use [CASE ACC], or just [ACC] (since no other feature can have ACC as a value), to stand for ⟨CASE, ACC⟩, and, in the case of Boolean features only, [+N] and [−PLU] to stand for ⟨N, +⟩ and ⟨PLU, −⟩, respectively. C^i stands for a category of type C having i bars, so N^2 means {⟨N, +⟩, ⟨V, −⟩, ⟨BAR, 2⟩}, which we shall also write as 'NP'. We add further feature detail in following square brackets, often giving just values where this makes clear what the whole feature specification must be; thus we can write NP[ACC] to stand for {⟨N, +⟩, ⟨V, −⟩, ⟨BAR, 2⟩, ⟨CASE, ACC⟩}. We will nearly always omit whatever part of the feature structure in a category has no bearing on the point at hand.

Let us return now to (3). Despite our restriction of feature values to atoms or categories, we still want to be able to refer to collections of feature specifications such as those dominated by HEAD and MAJOR in (3). The way we shall do this is by associating names such as **HEAD** with designated subsets of the set of features.

In fact, we define **HEAD** as follows:[5]

(4) **HEAD** = {N, V, PLU, PER, VFORM, SUBJ, PFORM, AUX, INV, PAST, PRD, ADV, SLASH, AGR, SUBCAT, BAR, LOC}

Several of these features have not been introduced so far, and it may be useful if we give a preliminary introduction to some of them now. N and V distinguish major parts of speech as detailed earlier. PLU is a binary feature for representing the singular/plural distinction, while PER takes the values 1, 2, and 3 to represent person distinctions.[6]

VFORM distinguishes parts of the verb paradigm (FIN, finite; INF, infinitival; BSE, base-form, i.e. bare infinitive; PAS, passive participle; etc.), and we will normally abbreviate ⟨VFORM, FIN⟩, etc., simply as [FIN]. SUBJ distinguishes sentences from verb phrases (they have the same bar level in the present theory). PFORM is used to encode subcategorizational requirements for particular prepositions (e.g. *rely on*, and it takes *to*, *for*, *of*, *on*, etc., as values. Thus *rely* subcategorizes for a PP[PFORM *on*]. Throughout the book, we abbreviate PFORM specifications in an obvious way, by writing 'PP[*on*]' for 'PP[PFORM *on*]', etc. There is no ambiguity, since the appearance of a terminal symbol like *on* or *of* in a feature bundle attached to P or PP will always mean that it is a value of PFORM. AUX is a Boolean (binary) feature which identifies auxiliary verbs. Likewise, INV picks out verb-initial sentences such as direct polar interrogatives in English (see chapter 4). Another Boolean feature, PRD, distinguishes predicative complement phrases (occurring after the copula) from non-predicative phrases of similar category, and the

Boolean ADV distinguishes adverbial instances of the A^2 category from adjectival ones (see chapter 6). SLASH has to do with the theory of unbounded dependencies and is discussed extensively below, especially in chapters 5, 7 and 8. AGR, which, like SLASH, takes categories as its values, marks properties that a predicate element has by virtue of its syntactic obligation to show concord with some other element, in a way clarified in chapters 5 and 6.

Note that we have not required that the subsets we designate impose a partition on the set of features. Hence two subsets could have a non-null intersection. We shall see elsewhere in this book that this possibility is realized. **HEAD** intersects with another set called **FOOT**. For instance, we shall propose in chapter 5, following Flickinger (1983), that the feature SLASH is in the intersection of **HEAD** and **FOOT**, having the properties of both head features and foot features.

We have already seen that, on the present approach, a syntactic category C is a set of feature specifications, where each feature specification is an ordered pair consisting of a feature and a feature-value. Moreover, it should be intuitively obvious that if C has as an element a pair of the form $\langle f, v \rangle$, for some feature f, then it should not also have as an element a pair $\langle f, v' \rangle$, where v' is distinct from v. (For example, a category could not have as elements both ⟨CASE, NOM⟩ and ⟨CASE, ACC⟩.) Now it is usual in mathematics to represent functions as sets of ordered pairs which meet just the condition we mentioned above. In other words, we might well regard syntactic categories as functions. They take features as arguments, and yield what we have called feature-values as values.[7] Thus instead of saying that a pair ⟨CASE, ACC⟩ is a member of category C, we can equally well say that C(CASE) = ACC, i.e. C yields the value ACC for the argument CASE. Ignoring feature specifications that include categories as values for the moment, suppose we assume two finite, nonempty sets: F, the set of features, and V, the set of feature-values. Then a syntactic category is a partial function from F into V.[8] We say 'partial' because we often want to allow the possibility that a category C is undefined for some particular feature $f \in F$. A total function from F would assign a value to every element of F, but a partial function need not do so. For example, it seems sensible to suppose that prepositions do not take a value for the feature VFORM. So a preposition category which we might write as P should be such that P(VFORM) is always undefined. We will use the symbol '$\sim$' to show that a feature is undefined, thus 'prepositions are $\sim$[VFORM]' should be read as saying that prepositions do not take a value for the feature VFORM. Note that, in the case of Boolean features such as INV, the two expressions '[$-$INV]' and '$\sim$[INV]' mean different things: the former describes a category C such that C(INV) = $-$, whereas the latter means that C(INV) is undefined.

When a partial function C on a set F is undefined for some element $x \in F$, we say that x is not in the *domain* of C; in notation $x \notin DOM(C)$. Another way of putting it is to say that $DOM(C)$ is that subset F' of F such that C is a total function on F'. Reverting to the set theoretical representation of functions, C will be undefined for $f \in F$ if and only if no pair of the form $\langle f, v\rangle$ belongs to C. Since it is possible for a partial function to be undefined for all arguments, it follows that we admit the empty set $\{\}$ as a possible category.

The possibility of categories that are partial functions allows for a rather natural definition of the notion 'minor category'. Minor category expressions (e.g. determiners, complementizers, conjunctions, etc.) are 'minor' in that they lack a BAR specification, and thus fall outside the X-bar system altogether:

(5) An instantiated category C is a minor category if and only if C(BAR) is undefined.

As this definition suggests, the analysis of categories that we are adopting gives us a convenient way of specifying the value that is associated with a given feature in some category. Where C is a category, '$C(f)$' can be read as 'the value which the category C assigns to the feature f' or, more simply, as 'C's value for f'. Thus 'A(PLU) = B(PLU)' could be read as 'A's value for number is the same as B's value for number'.

The elements of V, the set of feature-values, will either be elements like '+', '−', or integers, or names like FIN (finite). Features that have values of this sort we will call *atom-valued features*. But a feature-value is not necessarily atomic. The other possibility is for it to be a category. Some examples will make clear the kind of roles played by the two kinds of features.

Atom-valued features include Boolean features like N, with value-set $\{+, -\}$; integer-valued features like BAR (for bar-level in the sense of X-bar theory), with value-set $\{0, 1, 2\}$; and features like VFORM, which classify verb forms in terms of values like FIN (finite), INF (infinitival), etc. The atom-valued features thus often correspond to quite traditional classifications of linguistic elements according to their grammatical properties.

Category-valued features are rather less familiar. The feature SLASH is one example. It marks constituents that would be treated in transformational terms as having had a phrase extracted by a movement rule or deleted across an unbounded context.[9] Consider the following feature specification:

(6) $\langle \text{SLASH}, \{\langle \text{N}, +\rangle, \langle \text{V}, -\rangle, \langle \text{BAR}, 2\rangle, \langle \text{PER}, 3\rangle, \langle \text{PLU}, -\rangle\}\rangle$

SLASH will be employed extensively in chapters 7 and 8 in connection with unbounded dependencies, and its role in the theory can only be properly understood in terms of the discussion in those chapters, but it will suffice here to say that the intuitive interpretation of the feature specification in (6) is 'lacking a third person singular noun phrase subconstituent'.

It appears to be generally assumed that grammars employing features must adopt the following stipulation:

(7) Only a fully specified category may label a node.

However, we do not make this stipulation. In the present framework, fully specified categories have no privileged status, and something barely specified like (8)

(8) $\{\langle \text{BAR}, 2\rangle\}$

is just as much a category as any other. It is something like a Trubetzkoyan 'archicategory', having 2 as its bar level but being unspecified for all other features, even the major category features N and V. It can be read as simply 'phrase', and informally notated X^2. The availability of such underspecified categories is crucial to some of the analyses we develop in terms of our theory, as will be seen in later chapters. And although it might seem to the reader that allowing such massive underspecification in categories would lead to disastrous overgeneration of structures with vaguely labeled constituents, in fact this is not so. In the lexicon, most items are fully specified for syntactic features. This assumption interacts with the Head Feature Convention (discussed in chapter 5) to ensure that trees mostly have fully specified node labels. (What the Head Feature Convention says is basically that the head features on a mother category are the same as the head features on any daughter which is a head.)

The present theory of features makes heavy use of notions of *extension* and *unification*. These notions are defined on categories, not features. They both have relatively clear but somewhat technical definitions, which we will now present informally.

An *extension* of a category is like a superset of it, except for two details. First, the extension of a category must also be category, that is, it must be a superset which is also a function. Second, something has to be said about what it means to be a superset of a category that includes category-valued features.

In the case of atom-valued features, things are fairly clear. For example, we want $\{\langle \text{N}, +\rangle, \langle \text{V}, +\rangle, \langle \text{PRD}, +\rangle\}$ to be an extension of $\{\langle \text{N}, +\rangle, \langle \text{V}, +\rangle\}$. It simply has an additional feature specification. However, we do not want $\{\langle \text{N}, +\rangle, \langle \text{V}, +\rangle, \langle \text{PRD}, +\rangle, \langle \text{PRD} \; -\rangle\}$ to be an extension of the latter, because it is not a category. Since $\langle \text{PRD}, + \rangle$ and $\langle \text{PRD}, - \rangle$ contradict each other about the value for PRD, this set does not constitute a function (even a partial one) from features to feature-values.

In the case that a category-valued feature is present in a category C, we must ensure that extensions of C preserve the details inside the value of that feature. We want extensions of a feature containing [SLASH N2[PER 3, PLU −]], for example, to preserve the third person singular information locked up inside the value of SLASH. This means that our definition has to be recursive. Informally, it goes as follows:

(9) A category A is an *extension* of a category B ($B \sqsubseteq A$) if and only if
(i) the atom-valued feature specifications in B are all in A, and
(ii) for any category-valued feature f, the value of f in A is an extension of the value of f in B.

This definition is given more precisely in section 6.

Our concept of *unification* is essentially identical to that of Kay (1979), and is closely analogous to the operation of union on sets except that, as in the case of extension, the resulting set must be a function. Unification is undefined for sets containing feature specifications that contradict each other.

(10) Let K be a set of categories. The *unification* of K($\bigsqcup$K) is the smallest category which is an extension of every member of K, if such a category exists, otherwise, the unification of K is undefined.

For example, let $A = \{\langle \text{N}, + \rangle, \langle \text{PRD}, + \rangle\}$ and $B = \{\langle \text{N}, + \rangle, \langle \text{V}, - \rangle\}$. Then $\{\langle \text{N}, + \rangle, \langle \text{V}, - \rangle, \langle \text{PRD}, + \rangle\}$ is the unification of A and B. However, $\{\langle \text{N}, + \rangle, \langle \text{V}, - \rangle, \langle \text{PRD}, + \rangle, \langle \text{PLU}, + \rangle\}$ is not, because the gratuitous extra specification $\langle \text{PLU}, + \rangle$ is not in either A or B, and so is not in their union. Moreover, the unification of $\{\langle \text{N}, + \rangle, \langle \text{V}, + \rangle\}$ and $\{\langle \text{N}, + \rangle, \langle \text{V}, - \rangle\}$ is not defined, because of the clash regarding the value for V.

We conclude this section by noting the conventions we employ subsequently for referring to categories: α, β, C, C', C_i, ... will be used as metalanguage variables over categories. They will never occur in actual rules or metarules. X will be used to stand for the maximally underspecified category, namely $\{\}$, and is never a variable.

3 Feature co-occurrence restrictions

If a consonant is pharyngeal, then it is not nasal. Likewise, if a syntactic category exhibits a distinction in present versus past tense, then that category is not a preposition or a prepositional phrase. Restrictions of this sort are expressed in generative phonology by means of what Chomsky and Halle (1968, chapter 9) call marking conventions. Chomsky and Halle employ two kinds of marking conventions. One kind (1968, pp. 404–7) is illustrated in (11).

(11)
$$\begin{array}{lll} [+\text{voc}] & \rightarrow & [+\text{son}] \\ [+\text{nasal}] & \rightarrow & [+\text{son}] \\ [+\text{high}] & \rightarrow & [-\text{low}] \\ [+\text{low}] & \rightarrow & [-\text{high}] \\ [+\text{ant}] & \rightarrow & [+\text{cns}] \\ [-\text{cor}] & \rightarrow & [-\text{lateral}] \end{array}$$

These rules can be seen to constitute part of the definition of *possible phonological segment*. Each has the potential of reducing the space of possible segments by up to 25 per cent. We will refer to absolute conditions of the type shown in (11) as *Feature Co-occurrence Restrictions* (FCRs, hereafter).

FCRs, for us, constitute part of the definition of the notion 'legal extension of a category', and thus of the notion 'fully specified syntactic category'. Some FCRs will be universal and thus be part of the definition of 'possible syntactic category in a natural language', and some will be parochial and thus be part of the definition of, for example, 'possible syntactic category in English'. Consider an analogy with phonotactics. Particular combinations of syntactic features may constitute a possible category for a language, even though the language never happens to employ the category, just as */blik/* is a possible English word, though not an actual one. Likewise, particular combinations of syntactic features may not constitute a possible category for a given language, even though the same combination would be possible for another language. Thus */dnip/* is not a possible English word, despite the fact that nothing in 'universal phonotactics' prohibits it from being a word in some other language.

FCRs are typically stated as material conditionals or biconditionals. For an example, consider the restriction in English that the feature specification [+INV] implies the specifications [+AUX] and [FIN]. [+INV] is a specification that appears on sentences which include a subject but begin with a verb, and also appears on that sentence-initial verb. The FCR has as a consequence that such a verb will always be a finite auxiliary verb. It is formulated as follows:

(12) FCR 1: $[+\text{INV}] \supset [+\text{AUX}, \text{FIN}]$

For a second example, consider the statement that the major category P cannot be associated with tense. In fact, we can make this follow as a consequence from a stronger statement: VFORM values are only associated with the major category V. This is formulated as follows:

(13) FCR 2: [VFORM] $\supset$ [+V, −N]

(13) may be viewed as saying that the distinctions coded by the feature VFORM are only relevant to categories of type V. Unsurprisingly, our grammar also needs to contain the following FCRs in addition to (13):

(14) a. FCR 3: [NFORM] $\supset$ [−V, +N]
b. FCR 4: [PFORM] $\supset$ [−V, −N]

These three FCRs are, of course, universal in a rather uninteresting way: they amount to 'meaning postulates' which constrain the possible uses of the features VFORM, NFORM and PFORM.

There will be a substantial number of cases in the grammar where the utility of drawing some featural distinction rests on the presupposition that some other feature specification is also in force, but we shall not attempt to state all the requisite FCRs, though it would be straightforward to do so. Here are just two further examples: (i) it is only relevant to ask whether a category contains a specification for the Boolean feature PAST if it also specifies FIN as the value of VFORM. This constraint on PAST is ensured by the FCR below (see chapter 10 for some further discussion of this FCR):

(15) FCR 5: [PAST] $\supset$ [FIN, −SUBJ]

And (ii), the only constituents that can have a phrase omitted from them are themselves phrasal constituents. It makes no sense to think of a *word* with an NP gap in it, for example. Words all have the feature SUBCAT (see section 5) and constituents with something missing are marked with the feature SLASH (see chapters 5 and 7), thus we can state the following FCR to express the fact just noted:

(16) FCR 6: [SUBCAT] $\supset$ ~[SLASH]

4 Feature specification defaults

In English, typical consonants are not [+low], though /h/ is [+low] in the Chomsky and Halle feature system. And typical occurrences of NPs are not possessive, though possessive NPs do of course occur in English.

Chomsky and Halle (1968, pp. 405–7) introduce a notation for expressing this kind of fact in phonology, a notation which is illustrated in (17).

(17) [*u* high] → [+high]
[*u* nasal] → [−nasal]
[*u* low] → [−low]
[*u* ant] → [+ant]
[*u* cor] → [+cor]
[*u* cont] → [+cont]

These rules tell us what the default value is for the feature in question. If nothing is said about it, then we are to assume the value specified on the right of the arrow. We will refer to markedness conventions of this kind as *Feature Specification Defaults* (FSDs, hereafter).

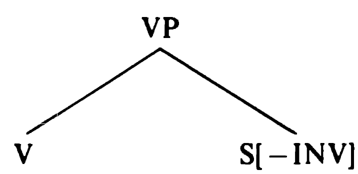
Our theory of syntactic categories will employ both FCRs and FSDs. In this section, we deal with the second type of marking convention, FSDs. These form an important part of the link between the highly schematic rules listed in the grammar and the fully specified structural descriptions they induce.

As the first illustration, consider rule (18).

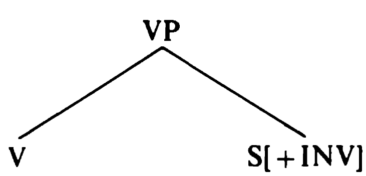
(18) VP → V S

Consider the rule in relation to the feature INV (mnemonic for INVerted sentence). Two of the tree fragments that we might expect to get from (18) are displayed in (19).

(19)a.



b.



That is, there is no particular reason for having [−INV] rather than [+INV] on the S daughter. But if we allow the grammar to admit tree fragments like (19b), then we will end up generating examples like **Lee believes will the children be late.* So INV needs to have a default specification, namely [−INV]. We can state this default as follows:

(20) FSD 1: [−INV]

Since there is no reason for INV not to have that specification on S, it must have it, according to the approach to defaults that we adopt in this book. There is, of course, one class of structures in which [+INV] is obliged to be present, namely those arising from rules such as (21).

(21) S[+INV] → V[AUX] NP VP[BSE]

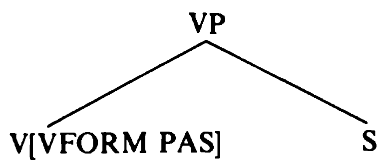
But, since this rule stipulates the presence of [+INV], the default will not be invoked.

For another example of the same point, let us look at the conditional FSD shown in (22):[10]

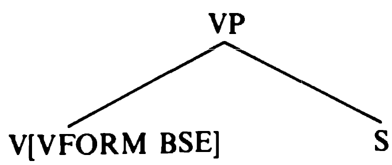
(22) FSD 7: [BAR 0] ⊃ ~[VFORM PAS]

This says that, other things being equal, a lexical category will not get instantiated as [VFORM PAS]. Now consider the two tree fragments in (23):

(23)a.



b.



If VFORM did not have a default excluding PAS, strings like **We were believed that the earth is flat* would be permitted by the grammar. Nevertheless, there are rules which require a [VFORM PAS] specification – for example, the rules which result from the passive metarule discussed in chapter 4. For example, suppose the grammar contains rule (24):

(24) VP[VFORM PAS] → V PP[*to*]

In this case, the [VFORM PAS] specification will be required to appear on the lexical head of the constituent admitted by this rule, and VFORM is thus relieved of the obligation to take some other value.

5 Lexical subcategorization

Any grammar of English has to provide for the ungrammaticality of such examples as (25).

(25) *I devoured to him that the grass is green.

One approach would be to let the syntax generate strings in which (for example) verbs occur with the wrong number and type of other constituents in the VP, and use the mechanisms that associate syntactic structures with meanings to eliminate them.[11] In other words, treat strings like (25) as grammatical but not semantically interpretable. This approach, which we might call 'semantic filtering', is not appropriate to the domain under consideration. The reason is that there is fairly clear evidence that the meaning of a verb does not completely determine its subcategorization. Consider the following sets of data, in which the first two sentences in each set illustrate the (near) synonymy of two verbs, and the second two examples demonstrate a dissimilarity between them as regards subcategorization.

(26) a. The beast ate the meat (ravenously)
b. The beast devoured the meat.
c. The beast ate (ravenously)
d. *The beast devoured.

(27) a. The ground sometimes shakes under your feet.
b. The ground sometimes quakes under your feet.
c. What is shaking the ground?
d. *What is quaking the ground?

(28) a. He gave this to me.
b. He donated this to me.
c. He gave me this.
d. *He donated me this.[12]

(29) a. It is likely that Alex will leave.
b. It is probable that Alex will leave.
c. Alex is likely to leave.
d. *Alex is probable to leave.

(30) a. Aren't you even going to try to solve it?
b. Aren't you even going to attempt to solve it?
c. Aren't you even going to try?
d. *Aren't you even going to attempt?

Further examples of the same sort could be given. What they suggest is that there are restrictions on contexts of occurrence for lexical items which the grammar must specify, and which cannot be reduced to facts about meaning.

Consider the following phrase structure rule:

(31) VP → V (NP) (NP) (PP) (PP) (S)

Our problem is, for example, to ensure that V expands as *devour* only when V is immediately adjacent to NP, and not, say, PP or S as (31) would permit. Viewed in this light, one obvious strategy for coping with the facts of subcategorization is to enrich the theory of grammar by introducing context-sensitive phrase structure rules and using rules of this type for lexical insertion, along the lines exemplified in (32).

(32) V → *bring*/____NP
V → *persuade*/____NP S
V → *decide*/____PP
V → *grow*/____AP
V → *save*/____NP PP
V → *trade*/____NP PP PP

However, an approach along these lines, though initially plausible, runs into numerous difficulties. One is that these rules include a mass of statements which redundantly repeat things that the phrase structure rules have already said, e.g. that NP PP sequences are found in English VPs but PP NP sequences are not.[13] This point is made and illustrated in detail in Heny (1979). Another problem is that the rule for expanding VP will have to introduce all the different sorts of constituent that verbs can demand as their complements in some instance, and since no verbs take *all* of these (for example, *devour* takes an NP but cannot also have an indirect object or a subordinate clause, witness, **I devoured the meat to him that grass is green*) there also have to be numerous negative specifications associated with the rules in (32).[14]

In response to these problems, we adopt an alternative approach to subcategorization, one which only employs context-free phrase structure rules. Consider the set of rules in the grammar which introduce preterminal symbols (such as A, COMP, Det, N, P and V). It is precisely these rules, and no others, which determine the possibilities of subcategorization for words. There is a clear sense in which these rules *are* the subcategorization frames for the language.

It is both necessary and sufficient for achieving correct subcategorization that the grammar provide a mechanism whereby the relevant subclasses of a preterminal symbol can be matched with the rules that introduce it. This can be done in a manner which is at once simple and transparent, by using integers as values for a feature SUBCAT which only appears on preterminal symbols.[15] As far as the lexicon is concerned, these integers are best seen as pointers to the rules which will allow the relevant item to be introduced.[16] We need to ensure that SUBCAT will not be defined for a category that assigns BAR a value greater than 0, and that all [BAR 0] categories are defined for SUBCAT. This is accomplished by the following FCRs:

(33) FCR 7: [BAR 0] $\equiv$ [N] & [V] & [SUBCAT]

(34) FCR 8: [BAR 1] $\supset$ $\sim$[SUBCAT]

(35) FCR 9: [BAR 2] $\supset$ $\sim$[SUBCAT]

Under this proposal, our grammar will contain rules such as those shown in (36) and (37):

(36) VP → V[1]

(37) VP → V[2] NP

Here, V[1] and V[2] are just an abbreviation for V[SUBCAT 1] and V[SUBCAT 2] respectively.[17] In addition, the lexicon will contain an entry which says that *weep* is a V[1], and that *devour* is a V[2]. More precisely, let us assume that a lexical entry contains at least four kinds of information: a phonological form, a category, an indication of any irregular morphology, and a meaning. Keeping to orthographical representations for convenience, we might expect entries of the following kind:

(38) ⟨ *weep*,
[[−N], [+V], [BAR 0], [SUBCAT 1]],
{*wept*},
weep′⟩

(39) ⟨ *devour*,
[[−N], [+V], [BAR 0], [SUBCAT 2]],
{},
devour′⟩

Now we say that a category *C* can immediately dominate a lexical item *α* just in case *C* extends the category label in the lexical entry for *α*. Consequently, V[1] can immediately dominate *weep*, but not *devour*.[18] Likewise, we ensure that V[2] will immediately dominate *abandon*, *enlighten*, *castigate*, *slap*, and so on, but not *disappear*, *elapse*, *expire*, *faint*, and so on.

Note that the interpretation of categories as feature bundles enables us to avoid the charge that totally distinct categories have to be postulated for verbs of different subcategorization types, with the result that one loses generalizations about verbs (e.g. that they all take tense). These generalizations are not lost since all verbs have at least two feature specifications in common (namely [+V, −N]) and it is this fact which forms the basis for FCRs like (13) above.

Our proposal for subcategorization has one very specific consequence: it entails that only the items introduced by the same rule as a given preterminal category *C* can be directly relevant to the question of where *C*'s subcategorization environment is met. This is not something that is entailed by context-sensitive accounts of lexical insertion such as that of Chomsky (1965). But it is worth noting that Chomsky chooses to stipulate the very constraint that follows as a theorem from the system outlined above (see Chomsky 1965, pp. 96, 99).

There are other advantages of our context-free system for subcategorization besides the consequence just noted. One that we shall comment on briefly here concerns coordination. Consider sentence (40), involving the verb *hand*, which requires both direct and indirect objects.

(40) We have handed or sent a copy of this letter to every student in the school.

In this example, *hand* is not in the context defined by the contextual feature specification [+ ____NP PP]. The problems for context-sensitive accounts of lexical insertion caused by the fact that coordination of lexical categories can destroy crucial adjacency relationships in this way are considerable, but have not to our knowledge been addressed by proponents of such accounts. In our system, however, there is no reference to linear adjacency in the conditions on insertion of the terminal symbol *hand*. *Hand* belongs to a category, let us say V[3], which is introduced by a rule that also specifies an accompanying NP and PP[PFORM *to*]. By the schema for coordination presented in chapter 8, section 2, any category C can dominate one or more Cs, a conjunction such as *and* or *or*, and one further C. Thus, a V[3] can dominate two V[3]s separated by *or*. Hence a tree for (40) can be admitted. The interaction of coordination and strict subcategorization is successfully predicted by the system assumed here, in fact, whereas it has never been satisfactorily treated within the context of Chomsky's (1965) proposals.

Finally, the feature SUBCAT allows for a straightforward definition of minor lexical category: a lexical item belongs to a minor lexical category if and only if it is specified for SUBCAT, but not specified for BAR.

6 A formal theory of features

This section contains some definitions that are crucially referred to in chapter 5, but those readers who prefer to skip it on a first reading will find that much of the content of that chapter is intelligible without it.

We begin by giving a more formal account of categories. Let A and F be finite sets, of atomic feature-values and features respectively. *Atom* (the set of atomic-valued features), is a subset of F. Only a certain subset of A may be associated with any given member of *Atom*. This value-set is determined by a function ρ. Matters are straightforward as far as members of *Atom* are concerned, but become a little more complicated when we consider category-valued features. In order to define what a possible category is, we need to say what the possible values of a feature are. But, for category-valued features, this requires a definition of possible categories! To deal with this problem, we give an inductive definition of the set K of possible categories.

Definition 1
ρ^0 is a function from F to $POW(A)$ such that for all $f \in (F - Atom)$, $\rho^0(f) = \{\{\}\}$.

If f belongs to *Atom*, then $\rho^0(f)$ is some subset of A. If f does not belong to *Atom*, then it is a category-valued feature, and is assigned the empty category, $\{\}$, by ρ^0.

Example 1
Let $F = \{\mathrm{N}, \mathrm{V}, \mathrm{SLASH}, \mathrm{AGR}\}$, $Atom = \{\mathrm{N}, \mathrm{V}\}$, $A = \{+, -\}$.
$\rho^0(\mathrm{N}) = \rho^0(\mathrm{V}) = A$, and $\rho^0(\mathrm{SLASH}) = \rho^0(\mathrm{AGR}) = \{\}$.

As we mentioned earlier, categories are formalized as partial functions. We use $Y^{(X)}$ to denote the set of all partial functions from a set X to a set Y.

Definition 2
The set K^0 of 0-level categories =
$\{C \in A^{(F)} : \forall f \in DOM(C)[C(f) \in \rho^0(f)]\}$

It follows from the preceding definition that if $\rho^0(f) = \{\{\}\}$, then $C(f)$ is undefined. Consequently, if C is a 0-level category, then $DOM(C) \subseteq Atom$.

Example 2
Continuing with the assignments in example (1), we have

$$
\begin{aligned}
K^0 = \{&\{\langle \mathrm{N}, +\rangle, \langle \mathrm{V}, +\rangle\},\\
&\{\langle \mathrm{N}, +\rangle, \langle \mathrm{V}, -\rangle\},\\
&\{\langle \mathrm{N}, -\rangle, \langle \mathrm{V}, +\rangle\},\\
&\{\langle \mathrm{N}, -\rangle, \langle \mathrm{V}, -\rangle\},\\
&\{\langle \mathrm{N}, +\rangle\},\\
&\{\langle \mathrm{N}, -\rangle\},\\
&\{\langle \mathrm{V}, +\rangle\},\\
&\{\langle \mathrm{V}, -\rangle\},\\
&\{\}\}
\end{aligned}
$$

We now wish to take the induction step in the definition of ρ. We shall impose an important constraint on ρ: the value-set that ρ associates with a given category-value feature f can only contain categories in which f does not already appear. That is, C can only be in the value set of f if f is not in the domain of C, or in the domain of any C' contained in C, at any level of embedding. This requires an ancillary definition.

Definition 3
Let $\in^+$ be the transitive closure of the relation $\in$. Let R be a relation.
Then $DOM^+(R) = \{x : \exists y[\langle x,y\rangle \in^+ R]\}$

Example 3
Let R be $\{\langle a, 1\rangle, \langle b, 2\rangle, \langle c, \{\langle a, 2\rangle, \langle d, 2\rangle\}\rangle\}$.
Then $DOM(R) = \{a, b, c\}$, while $DOM^{+}(R) = \{a, b, c, d\}$.

Definition 4
Suppose K^{n-1} and ρ^{n-1} are defined. Then we extend ρ^{n-1} to ρ^{n} as follows:

(i) If $f \in Atom$, then $\rho^{n}(f) = \rho^{n-1}(f)$,
(ii) otherwise, $\rho^{n}(f)$ is a subset of
$\{C \in K^{n-1}: \exists C' \in \rho^{n-1}(f)[C' \subseteq C] \,\&\, f \notin DOM^{+}(C)\}$.

Thus if f is a category-valued feature, then the value-set associated with f by ρ^{1} will be some subset of 0-level categories.

Example 4
Continuing with example (2), set $\rho^{1}(\text{SLASH}) = \{\{\langle \text{N}, -\rangle, \langle \text{V}, -\rangle\}\}$ and $\rho^{1}(\text{AGR}) = \{\{\langle \text{N}, +\rangle, \langle \text{V}, -\rangle\}\}$. Before illustrating the values of ρ^{2}, we have to proceed to the next definition.

We now define higher-level categories.

Definition 5
$K^{n} = \{C \in (A \cup K^{n-1})^{(F)}: \forall f \in DOM(C)[C(f) \in \rho^{n}(f)]\}$

Example 5
When we reach 1-level categories, we get all the 0-level categories together with the categories formed by assigning values to the category-valued features SLASH and AGR. Thus

$$
\begin{aligned}
\text{K}^{1} = \{&\{\langle \text{N}, +\rangle, \langle \text{V}, +\rangle\},\\
&\{\langle \text{N}, +\rangle, \langle \text{V}, -\rangle\},\\
&\{\langle \text{N}, -\rangle, \langle \text{V}, +\rangle\},\\
&\{\langle \text{N}, -\rangle, \langle \text{V}, -\rangle\},\\
&\{\langle \text{N}, +\rangle\},\\
&\{\langle \text{N}, -\rangle\},\\
&\{\langle \text{V}, +\rangle\},\\
&\{\langle \text{V}, -\rangle\},\\
&\{\}
\end{aligned}
$$

$\{\langle N, +\rangle, \langle V, +\rangle, \langle SLASH, \{\langle N, -\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle N, +\rangle, \langle V, -\rangle, \langle SLASH, \{\langle N, -\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle N, -\rangle, \langle V, +\rangle, \langle SLASH, \{\langle N, -\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle N, -\rangle, \langle V, -\rangle, \langle SLASH, \{\langle N, -\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle N, +\rangle, \langle SLASH, \{\langle N, -\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle N, -\rangle, \langle SLASH, \{\langle N, -\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle V, +\rangle, \langle SLASH, \{\langle N, -\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle V, -\rangle, \langle SLASH, \{\langle N, -\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle SLASH, \{\langle N, -\rangle, \langle V, -\rangle\}\rangle\},$

$\{\langle N, +\rangle, \langle V, +\rangle, \langle AGR, \{\langle N, +\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle N, +\rangle, \langle V, -\rangle, \langle AGR, \{\langle N, +\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle N, -\rangle, \langle V, +\rangle, \langle AGR, \{\langle N, +\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle N, -\rangle, \langle V, -\rangle, \langle AGR, \{\langle N, +\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle N, +\rangle, \langle AGR, \{\langle N, +\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle N, -\rangle, \langle AGR, \{\langle N, +\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle V, +\rangle, \langle AGR, \{\langle N, +\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle V, -\rangle, \langle AGR, \{\langle N, +\rangle, \langle V, -\rangle\}\rangle\},$
$\{\langle AGR, \{\langle N, +\rangle, \langle V, -\rangle\}\rangle\}\}$

Example 6

Let us now consider the value sets that ρ^2 can associate with SLASH. According to Definition (4.ii), the following conditions must be met in order for SLASH to receive some C in K^1 as a possible value:

a. $\exists C' \in \rho^1(\text{SLASH})[C' \subseteq C]$

b. SLASH $\notin DOM^+(C)$

In a previous example, we had $\rho^1(\text{SLASH}) = \{\{\langle N, -\rangle, \langle V, -\rangle\}\}$. Consequently, by virtue of (a) we can exclude from consideration any C which lacks the feature specifications $\langle N, -\rangle$ and $\langle V, -\rangle$. And by virtue of (b), we can exclude any C in which SLASH is already assigned a value. As a result, we have

$$\rho^2(\text{SLASH}) = \{\{\langle N, -\rangle, \langle V, -\rangle\} \\ \{\langle N, -\rangle, \langle V, -\rangle, \langle AGR, \{\langle N, +\rangle, \langle V, -\rangle\}\rangle\}\}$$

Definition 6
The set K of all categories based on A, F and ρ is K^n, where n is the cardinality of $F - Atom$.

By virtue of the constraint we placed on ρ, $\rho^n(f)$ will only be non-empty for $f \in (F - Atom)$ if there are at least $n-1$ category-valued features distinct from f. Hence, K^n will be empty for all $n >$ cardinality of $(F - Atom)$. Since F is finite, so is K.

Example 7
If we continue the recursion from the previous example, K^2 will contain all the categories in K^1, together with categories like
$\{\langle \text{N}, +\rangle, \langle \text{V}, +\rangle, \langle \text{SLASH}, \{\langle \text{N}, -\rangle, \langle \text{V}, -\rangle,$
$\langle \text{AGR}, \{\langle \text{N}, +\rangle, \langle \text{V}, -\rangle\}\rangle\}\rangle\}$ and
$\{\langle \text{N}, -\rangle, \langle \text{V}, -\rangle, \langle \text{SLASH}, \{\langle \text{N}, -\rangle, \langle \text{V}, -\rangle,$
$\langle \text{AGR}, \{\langle \text{N}, +\rangle, \langle \text{V}, -\rangle\}\rangle\}\rangle\}$
In fact, the second of these will be the only new category in K^2 which contains the specifications $\langle \text{N}, -\rangle$, $\langle \text{V}, -\rangle$. But it cannot belong to $\rho^3(\text{SLASH})$ because it already contains a value for SLASH. Hence $\rho^3(\text{SLASH}) = \{\}$ and similarly for $\rho^3(\text{AGR})$.

Our next task is to define the notions of extension, identity, and unification.

Definition 7: Extension
Given two syntactic categories A and B, B is an extension of A ($A \sqsubseteq B$) if and only if
(i) for all $f \in Atom$, if $f \in DOM(A)$ then $B(f) = A(f)$, and
(ii) for all $f \in (F - Atom)$, if $f \in DOM(A)$ then $B(f)$ is an extension of $A(f)$.

Definition 8
Given two categories A and B, $A = B$ if and only if A is an extension of B and B is an extension of A.

The relation 'is an extension of' defines a partial order on S. Our definition of unification, like that of Gazdar and Pullum (1982), is equivalent to the standard notion of least upper bound in lattice theory.

Definition 9: Unification
(i) Let $S \subseteq K$ be a set of categories, and let $C \in K$. Then C is an *upper bound* for K if and only if for all $C' \in S$, C is an extension of C'.
(ii) C is the *unification* of S if and only if C is an upper bound for S and for all C', C' is an upper bound for S, then C' is an extension of C.

If $S = \{C_1, \ldots, C_n\}$ and C is the unification of S, then we sometimes write $\bigsqcup(C_1, \ldots, C_n) = C$.

We conclude this section by providing a formal reconstruction of our informal notation for FCRs and FSDs. These are predicates which can be either true or false of categories. Our reconstruction takes the form of a set of rules that map our FCR/FSD representations into lambda expressions of a familiar kind.

Definition 10: FCR and FSD translation

(i) An expression of the form $[f\ v]$ translates into $C(f) = v$.

(ii) An expression of the form $[f]$ translates into $\exists x\,[C(f) = x]$.

(iii) An expression of the form $\sim\alpha$ translates into $\sim\alpha'$, where α' is the translation of α.

(iv) Expressions of the form $\alpha \supset \beta$, $\alpha \equiv \beta$, $\alpha \vee \beta$, and $\alpha \,\&\, \beta$ translate into $\alpha' \supset \beta'$, $\alpha' \equiv \beta'$, $\alpha' \vee \beta'$, and $\alpha' \,\&\, \beta'$, respectively, where α' and β' are the translations of α and β respectively.

(v) An FCR or FSD consisting of an expression α translates as $\lambda C[\alpha']$ where α' is the translation of α.

Here are a couple of examples to show how this translation works:

Example 8

(i) FCR 1: [+ INV] $\supset$ [+AUX, FIN]
Our abbreviating conventions, discussed above, make this equivalent to:

(ii) FCR 1: [INV +] $\supset$ ([AUX +] & [VFORM FIN])
The translation rules given in Definition 10 convert this into the following lambda expression:

(iii) FCR 1: $\lambda C[(C(\text{INV}) = +) \supset ((C(\text{AUX}) = +) \,\&\, (C(\text{VFORM}) = \text{FIN}))]$

Example 9

(i) FSD 7: [BAR 0] $\supset$ $\sim$[VFORM PAS]
translates into:

(ii) FSD 7: $\lambda C[(C(\text{BAR}) = 0) \supset \sim(C(\text{VFORM}) = \text{PAS})]$

As can be seen, FCRs and FSDs are exactly the same kind of formal objects: functions from categories to truth values. They differ only in the role they play in the grammar, a matter which will be made explicit in chapter 5.

Notes

1 We adopt the position that, strictly speaking, a *category* is a set of expressions, while a *category label* is an object which can be used to refer to categories, and

to annotate nodes in trees. However, we will often speak of categories where strictly we mean category labels, in cases where we think no confusion will arise.

2 Hellan (1980) and Lapointe (1980) are exceptions. Note that they did not appear until about 25 years after the development of TG.

3 See Chomsky (1970) for the origins of this proposal, and Bresnan (1976) for further development. And see Jackendoff (1977) for discussion and defense of an alternative. When we need to distinguish between major class features and the lexical categories N and V, we shall simply mark the zero bar level on the latter: N^0 and V^0.

4 Although presented differently, the analysis of categories and features presented here set-theoretically has an equivalent graph-theoretic formulation. In particular, crucial graph-theoretic notions like extension and unification simply carry over into equivalent set-theoretic definitions.

5 In including SLASH among the **HEAD** features, we are adopting a proposal made by Sells (1983b) in the light of Flickinger 1983. The implications of this proposal will become apparent in chapters 7 and 8.

6 Our choice of person and number features is merely conventional here. For some serious discussion of such features, see Karttunen 1984 and Sag et al. (forthcoming).

7 In Pollard 1984 a slightly different approach is proposed: features are assigned sets of values, not just individual values. The motivation has to do with the feature representation of under-specified lexical categories (e.g. the category to which the noun *fish* belongs, which is vague rather than ambiguous on Pollard's account as regards the distinction between singular, plural, and mass nouns).

8 A more precise definition of syntactic category is presented in section 6. The definition is such that, given finite sets of features and atomic feature-values, the set of all possible categories is also finite.

9 SLASH gets its name from the informal notation used in GPSG. This notation (e.g. 'S/NP') is reminiscent of categorial grammar, which interprets the notation in a different way.

10 The need to permit conditional FSDs was first discussed explicitly by Warner in an early draft of Warner (forthcoming). As can be seen, FCRs and FSDs are notated identically and this correctly reflects their formal reconstruction (for which, see the final section of this chapter). However, they differ in the way they are invoked by the grammar, and this difference will become evident in chapter 5.

11 It might appear that Brame (1978) and Bresnan (1978) are taking this view, but in fact they postulate a level of 'functional structure', which they regard as syntactic, in the sense that linguistic rules can refer to properties of *representations* at that level. We are not sure that anyone currently espouses the 'semantic filtering' approach that we are considering (and will reject) in the text.

12 Under some transformational approaches this example and the next two could be taken to illustrate governed application of transformational rules rather than subcategorization; but in all lexicalist theories and most recent versions of

transformational grammar, subcategorization is involved.

13 Sentences like *We extend to you our sincere condolences* might look as if they refute the claim in the text, but if such structures are assigned on analysis under which the NP is in 'extracted' position outside the VP, the incompatibility dissolves.

14 Chomsky (1965, pp. 110–11, 166–7) discusses inconclusively some conventions that might supply these negative specifications.

15 Cf. Pollard (1984) for a functionally similar but technically distinct use of the feature SUBCAT.

16 As a matter of historical interest, we note that Chomsky and Halle propose a mechanism for dealing with lexical exceptions to phonological rules (1968, p. 173) which is reminiscent of our approach to subcategorization:

> Each rule of the phonology has a certain identifying number. We associate with each number n a new 'distinctive feature' $[\pm n]$. Suppose that the rule numbered n is $A \rightarrow B/C\_\_\_D$. Then we stipulate that A must be marked $[+n]$ if the rule numbered n is to apply to it.

17 Thus V^2 and V[2] stand for different things. The former abbreviates the category V[BAR 2], whereas the latter abbreviates the category V[SUBCAT 2].

18 We have not explicitly invoked PS rules to carry out lexical insertion. To do so, we could postulate a rule schema of the form

$$C \rightarrow \alpha$$

where C and α meet the conditions specified in the text. However, this extra device is not really required, given the manner in which the grammar determines structural descriptions. For more details, see chapter 5, section 5.