

Categorial Grammars, Lexical Rules and the English Predicative

Bob Carpenter
Carnegie Mellon University

July 27, 1995

1 Introduction

When applying categorial grammars to the study of natural languages, it is traditional to assume a *universal* collection of phrase-structure schemes. Not only are these phrase-structure schemes applied cross-linguistically, but a compositional type-driven functional semantics is usually assumed to be determined by syntactic structure. It follows from these strong semantic and syntactic restrictions that all language-specific generalizations must be lexically determined in a categorial grammar; once the lexicon is established for a language, universal rules of syntactic and semantic combination take over to completely determine the set of grammatical expressions and their meanings. With such a large responsibility being assigned to the lexicon, it is not surprising that a simple list is not a sufficiently structured way of organizing lexical information. To achieve any degree of empirical coverage, lexical mechanisms must be provided to account for natural generalizations that exist within the lexicon of a single language.

In this paper, we will study the possibilities for applying *lexical rules* to the analysis of English syntax, and in particular the structure of the verb phrase. We will develop a lexicon whose empirical coverage extends to the full range of verb subcategories, complex adverbial phrases, auxiliaries, the passive construction, yes/no questions and the particularly troublesome case of predicatives. The effect of a lexical rule, in our system, will be to produce new lexical entries from old lexical entries. The similarity between our system and the metarule system of generalized phrase-structure grammar (GPSG, as presented in Gazdar, *et al.* 1985) is not coincidental. Our lexical rules serve much the same purpose as metarules in GPSG, which were restricted to lexical phrase structure rules. The similarity is in a large part due to the fact that with the universal phrase-structure schemes being fixed, the role of a lexical category assignment in effect determines phrase-structure in much the same way as a lexical category entry and lexical phrase-structure rule determines lexical phrase-structure in GPSG. Our lexical rules will also bear a relationship to the lexical rules found in lexical-functional grammar (LFG, see Bresnan 1982), as LFG rules are driven by the grammatical role assigned to arguments. Many of our analyses were first applied to either LFG or GPSG, as these were the first serious linguistic theories based on a notion of unification. In the process of explaining the basic principles behind categorial grammar and developing our lexical rule system, we will establish a categorial grammar lexicon with coverage of English syntactic constructions comparable to that achieved within published accounts of the GPSG or LFG frameworks.

Language, at its most abstract level, is simply a relation between expressions and meanings. Syntacticians are primarily interested in the structure of the set of grammatical or well-formed expressions, while semanticists concentrate on the range of possible meanings that can be expressed by a language and the relation of these meanings to the world, to cognitive agents or to both. In categorial grammars, lexical assignments to basic expressions and potential syntactic constructions

are restricted in such a way that only semantically meaningful combinations of syntactic categories are allowed. The combinatorial semantics can, in a strong sense, be read directly from the syntactic categories involved and their analysis, much as in the type-driven translation scheme of Klein and Sag (1985). We will not be interested in particular semantic claims, either to do with the nature of meaning or the particular meanings associated with individual lexical items, but will instead be concerned with the purely compositional aspects of semantic interpretation having to do with the way complex phrase meanings are built up from meanings and categories assigned to lexical entries.

The paper is organized roughly as follows. We start by presenting the basics of categorial grammar, including the directional functional category system, the nature of the lexicon, the semantic domain structure and the basic applicative universal phrase-structure schemes. We then turn to English and develop the simple noun phrase and verb phrase syntax, with a brief diversion to discuss the nature of modification in categorial grammar. We then develop a detailed account of the core verbal lexicon, concentrating especially on the auxiliary system, control verbs and complex adverbials. We then turn our attention to the development of a lexical rule system and apply it to the core lexicon that we have created, concentrating on verbal lexical rules and the predicative constructions in particular.

We will not be concerned, in this paper, with how unbounded dependency and coordinate constructions can be treated within categorial grammars. Our attention in the base lexicon is focused on basic complement selection and in the lexical rules on bounded phenomena. The addition of type-raising and composition rule schemes, abstraction metarules and other devices allow unbounded dependency and coordination constructions to be handled in a semantically appropriate fashion. In particular, the approaches of Steedman (1985, 1987, 1988), Morrill (1987b, 1988), and Moortgat (1987c, 1988, 1988b) to unbounded dependency and coordination constructions extend and are compatible with the applicative phrase structure schemes presented in this paper.

2 Applicative Categorial Grammar

The study of categorial grammars originated with Ajdukiewicz (1935), who developed a logical language in which the combinatory properties of expressions could be expressed functionally. As in most linguistic analyses, every basic or lexical expression is assigned to one or more categories, in categorial grammar. These lexical category assignments fully determine the combinatorial properties of expressions, because the set of phrase structure rules is assumed to be universal. Thus, any language specific details and generalizations must be expressed in the lexicon. Before introducing our system of lexical rules, we will pause to lay out the basics of Bar-Hillel's (1953) pure directed categorial grammar system, which will serve as the basis for our analysis of the English core lexicon.

2.1 Category System

The basic assumption of categorial grammar is that there is some fixed finite set *BasCat* of *basic categories* from which other categories are constructed. For our purposes here, it will be sufficient to assume that *BasCat* contains the following basic categories:

- | | | |
|-----|---------------|--------------------|
| (1) | CATEGORY | DESCRIPTION |
| | $np(P, N, C)$ | <i>noun phrase</i> |
| | $n(N)$ | <i>noun</i> |
| | $s(V)$ | <i>sentence</i> |

where P, N, C and V range over *features* drawn from the following *finite* lists:¹

(2)	VARIABLE	FEATURE	VALUES
	P	person	1,2,3
	N	number	<i>sing, plu</i>
	C	case	<i>subj, obj</i>
	V	verb form	<i>bse, fin, perf, pred, inf</i>

We thus have 12 np possibilities, 2 n possibilities and 5 possibilities for s , depending on the values assigned to features. We can simply assume that we have 19 basic categories and not worry about the structure of these categories in terms of features.²

The finite set of basic categories is used to generate an infinite set of functional categories, each of which specifies a (possibly complex) argument and result category. The fundamental operation is that of concatenating an expression assigned to a functional category to an expression of its argument category to form an expression of its result category, with the order of the concatenation being determined by the functional category. For example, a determiner will be specified as a functional category that takes a noun complement to its right to form a noun phrase result, with agreement being handled by identity of simple features. More formally, we take the complete set $\text{Cat}(\text{BasCat})$ of *categories* generated from the basic categories BasCat to be the least such that

- (3)
- $\alpha \in \text{Cat}(\text{BasCat})$ if $\alpha \in \text{BasCat}$
 - $\alpha / \beta, \alpha \setminus \beta \in \text{Cat}(\text{BasCat})$ if $\alpha, \beta \in \text{Cat}(\text{BasCat})$.

A category $\alpha \setminus \beta$ or α / β is said to be a *functor* category and to have a *domain* or *argument* category of β and a *range* or *result* category of α . A functional category of the form α / β is said to be a *forward functor* and looks for its β argument to the right, while the *backward functor* $\alpha \setminus \beta$ looks for its argument to the left. With our choice BasCat of basic categories, the following would all be categories in $\text{Cat}(\text{BasCat})$:

- (4)
- $$\begin{aligned} & n(\textit{sing}), \quad (n(\textit{plu}) \setminus n(\textit{plu})) / np(2, \textit{plu}, \textit{obj}), \\ & (s(\textit{fin}) \setminus np(3, \textit{sing}, \textit{subj})) / (s(\textit{fin}) \setminus np(3, \textit{sing}, \textit{subj})) \\ & ((s(\textit{fin}) / np(3, \textit{sing}, \textit{subj})) / np(2, \textit{plu}, \textit{obj})) / np(1, \textit{sing}, \textit{obj}) \end{aligned}$$

From now on, we will omit disambiguating parentheses within categories, taking the slashes to be left-associative operators, so that, for instance:

- (5)
- $$\begin{aligned} \alpha \setminus \beta / \beta &= (\alpha \setminus \beta) / \beta \\ \alpha \setminus \beta / \beta \setminus \alpha &= ((\alpha \setminus \beta) / \beta) \setminus \alpha \\ \alpha \setminus \beta / (\alpha \setminus \beta) / \gamma &= ((\alpha \setminus \beta) / (\alpha \setminus \beta)) / \gamma. \end{aligned}$$

¹We will later extend the possible verb forms and cases to account for complementized sentences, expletives and prepositional arguments. More fine-grained linguistic analyses would presumably assume a finite number of additional features and corresponding values.

²Our notation is intentionally suggestive of logic grammars, which are a particular kind of unification grammar employing first-order terms for data structures. The entire system presented here, including the lexical rules, has been implemented straightforwardly in Prolog by simply using first-order terms for basic categories and taking the slashes to be infix binary functions. More sophisticated categorial logic grammars have been investigated for syntactic analysis by Moortgat (1987) and Morrill (1988) and for semantic analysis by Carpenter (1989).

2.2 Semantic Domains

One of the basic tenets of categorial grammar is that the category assigned to an expression should express its semantic functionality directly (this idea can be traced back at least to Montague (1970)). The semantics of an expression is usually encoded using Church’s (1940) simple theory of functional types, where arbitrary semantic domains are assigned to basic categories and simple functional domains are assigned to functor categories. For the sake of simplicity, we assume such a simply typed semantics, though we note that other semantics for the λ -calculus could be adapted just as easily (for instance, see Chierchia and Turner, forthcoming, who use an untyped (or mono-typed) λ -calculus semantics for a categorial grammar analysis of properties and nominalization). The set of types Typ is defined following Church (1940) by taking the minimal set such that:

- (6) • $\alpha \in \text{Typ}$ if $\alpha \in \text{BasCat}$
 • $\langle t_1, t_2 \rangle \in \text{Typ}$ if $t_1, t_2 \in \text{Typ}$

A type $\langle \alpha, \beta \rangle$ is intuitively meant to denote the type of functions from objects of type α to objects of type β . We can define a simple typing function $\tau : \text{Cat}(\text{BasCat}) \Rightarrow \text{Typ}$ that assigns a type to every category by setting:

- (7) • $\tau(\alpha) = \alpha$ if $\alpha \in \text{BasCat}$
 • $\tau(\alpha / \beta) = \tau(\alpha \setminus \beta) = \langle \beta, \alpha \rangle$

Finally, we will define a *semantic domain* \mathcal{D}_t for every type $t \in \text{Typ}$ in the same way as Church, by assuming that every basic type $\alpha \in \text{Typ}$ is assigned to some set \mathcal{D}_α and that every complex type $\langle t_1, t_2 \rangle$ is assigned a type according to the scheme:

$$(8) \quad \mathcal{D}_{\langle t_1, t_2 \rangle} = \mathcal{D}_{t_2}^{\mathcal{D}_{t_1}}$$

where we take A^B to be the set of total functions from the set B to the set A . The result is that every category is assigned to exactly one semantic domain. It is important to note that there is no requirement that the domains assigned to different basic categories be disjoint, so that for instance, the domains $\mathcal{D}_{np(3,sing,subj)}$ and $\mathcal{D}_{np(2,sing,obj)}$ could be identical. Similarly, there is no restriction preventing the domains assigned to basic categories from being equivalent to other functional domains that do not involve them; in Montague’s (1970) analysis, the semantic domain assigned to the complex verb phrase category was the same as that assigned to the basic noun category. We will write $\alpha : \mathbf{f}$ for the pair consisting of the syntactic category α and a semantic object \mathbf{f} drawn from the domain $\mathcal{D}_{\tau(\alpha)}$.³

We will not be interested in the fine grained analysis of any particular semantic domain \mathcal{D}_t , since it will not be relevant to the task at hand. A number of categorial semantics have been worked out which are more or less compatible with the categorial grammar presented here (see Montague 1970, Dowty 1979, Carpenter 1989, Chierchia and Turner forthcoming).

³To be more precise, we follow Montague (1970) in using a λ -calculus expression as a stand-in for a semantic object.

2.3 Lexical Assignments

As is usual in formal grammars, we will suppose that we have some finite set BasExp of *basic expressions*. For our purposes, BasExp can be taken to be a finite subset of English words.⁴ A *lexicon* is then simply taken to be a relation between basic expressions and pairs consisting of syntactic categories and semantic objects of the appropriate types. For instance, we could have a lexical entry $np(3, \text{sing}, \text{subj}) : \mathbf{opus}$ for the word *opus* if the constant \mathbf{opus} were an element of the domain $\mathcal{D}_{np(3, \text{sing}, \text{subj})}$. More formally, a *lexicon* is a relation

$$(9) \quad \text{Lex} \subseteq \text{BasExp} \times (\text{Cat}(\text{BasCat}) \times \mathcal{D}),$$

subject to the semantic well-typing constraint that

$$(10) \quad \text{if } \langle e, \langle \alpha, \mathbf{f} \rangle \rangle \in \text{Lex} \text{ is a } \textit{lexical entry} \text{ then } \mathbf{f} \text{ is an element of the domain } \mathcal{D}_{\tau(\alpha)}.$$

Note that nothing in this definition rules out lexical syntactic or semantic ambiguity, which would correspond to lexical entries of the form $\langle e, \langle \alpha, \mathbf{f} \rangle \rangle$ and $\langle e, \langle \alpha', \mathbf{f}' \rangle \rangle$ where $\alpha \neq \alpha'$ and/or $\mathbf{f} \neq \mathbf{f}'$.

Most of the work in this paper will be directed toward the construction of a suitably rich lexicon for dealing with the basic syntactic constructions of English.

2.4 Application Schemes

Categorial grammar is essentially a phrase-structure formalism. What we mean by this is that there are lexical assignments to basic expressions and a set of phrase-structure rules that combine expressions to produce phrases purely on the basis of syntactic categorization. Categorial grammars are rather unique among linguistic theories in postulating an *infinite* set of categories and phrase-structure rules rather than the finite set found in context-free phrase structure grammars. Luckily, the set of rules used in categorial grammar is quite well-behaved, and in the case of the pure applicative categorial grammar, can be derived as instances of two schemes. The reliance on phrase-structure rules guarantees a resulting theory that is purely concatenative in that grammar rules have locally determined behavior.⁵

Following our intuitive explanation of α / β as a forward-looking functor and $\alpha \setminus \beta$ as a backward-looking functor, we postulate the following two application schemes:

$$(11) \quad \begin{aligned} &\bullet \alpha / \beta : \mathbf{f} \quad \beta : \mathbf{g} \quad \rightarrow \quad \alpha : \mathbf{f}(\mathbf{g}) \quad (\textit{forward application}) \\ &\bullet \beta : \mathbf{g} \quad \alpha \setminus \beta : \mathbf{f} \quad \rightarrow \quad \alpha : \mathbf{f}(\mathbf{g}) \quad (\textit{backward application}) \end{aligned}$$

where $\alpha, \beta \in \text{Cat}(\text{BasCat})$ range over all syntactic categories and where $\mathbf{f} \in \mathcal{D}_{\tau(\alpha/\beta)} = \mathcal{D}_{\tau(\alpha \setminus \beta)}$ and $\mathbf{g} \in \mathcal{D}_{\tau(\beta)}$. We have written our phrase-structure schemes with a bottom-up orientation as is common in categorial grammars. Note that because we have $\mathbf{f} \in \mathcal{D}_{\langle \tau(\beta), \tau(\alpha) \rangle}$ and $\mathbf{g} \in \mathcal{D}_{\tau(\beta)}$ we will have $\mathbf{f}(\mathbf{g}) \in \mathcal{D}_{\tau(\alpha)}$, thus ensuring that the semantic type of the result is appropriate for the

⁴We will not consider phrasal lexical entries, though the formalism can be easily extended to handle them by simply assuming that the lexicon can assign categories to arbitrary expressions. We will also not consider building words out of morphemes categorially, though see Dowty (1979), Keenan and Timberlake (1988) and Moortgat (1987c).

⁵Gazdar (1981) contains an interesting and relevant discussion of the role of syntactic categories in phrase-structure grammars as fully determining distributional behavior locally. Gazdar contrasts the strict phrase-structure approach with transformational analyses, and the paper contains a reply from Chomsky.

resulting syntactic category α . This last property is referred to as *strong typing* in programming language theory, where a strongly typed program can never give rise to semantic type conflicts.

It should be obvious that with a finite lexicon only a finite number of instances of the application phrase structure schemes will ever be necessary. Instances with more slashes than lexical categories will never be invoked, since the rules strictly reduce the number of slashes in categories. This means that any finite categorial lexicon, together with the application schemes, will determine a grammar structurally equivalent to a context-free grammar. Somewhat surprisingly, the converse to this result also holds in the weak generative case, as was proved by Gaifman (Bar-Hillel, Gaifman and Shamir 1960). That is, every context-free language (set of expressions generated by a finite context-free grammar) can be generated by a categorial grammar applying the application schemes to a finite lexicon. Consequently, evidence beyond simple acceptability of sentences must be employed to distinguish between categorial and context-free grammars. The strongest motivation for using categorial grammars is the ease with which they can be extended to provide adequate semantic analyses of unbounded dependency and coordination constructions.

3 Basic English Lexicon

In this section, we will provide a lexical characterization of the core syntactic constructions available in English. We begin with the simple noun and verb phrases, then consider the nature of modifiers and finally conclude with a more detailed analysis of the verb phrase. The point here is to create a sufficiently rich base lexicon from which to begin to study lexical rules and other extensions to the basic categorial framework.

3.1 Simple Noun Phrases

We begin our study of English with the simplest kinds of noun phrases, including proper names, pronouns and simple determiner-noun constructions. We will use the determiner-noun analysis to illustrate the way in which agreement can be handled in a simple way through schematic lexical entries expressed in terms of features.

Proper Names and Pronouns

Proper names are the simplest kind of noun phrase, taking third person singular agreement and occurring in either subject or object position. We will use the following notation to express lexical category assignments:

$$(12) \quad np(3, sing, C) \rightarrow opus, bill, milo$$

We follow the convention of assuming that the variables can take on any of the possible values of the feature over which they range. Thus, the above lexical entry is really schematically representing two lexical entries, one where $C = subj$ and another where $C = obj$. We will also assume that there is a constant of the appropriate sort attached to each lexical entry for each word. Thus the complete lexical entries for *Opus* would be $np(3, sing, subj) : \mathbf{opus}$ and $np(3, sing, obj) : \mathbf{opus}$. Even though we will only provide a single lexical entry for many categories, the reader is urged to use his or her imagination to fill in the lexical categories of related expressions.

The next simplest category is that of pronouns, which evidence a large degree of variation in feature assignment. Pronoun distribution can be accounted for with the following lexical entries:

- (13)
- | | | |
|---------------------|---------------|------------|
| $np(1, sing, subj)$ | \rightarrow | i |
| $np(1, sing, obj)$ | \rightarrow | me |
| $np(1, plu, subj)$ | \rightarrow | we |
| $np(1, plu, obj)$ | \rightarrow | us |
| $np(2, N, C)$ | \rightarrow | you |
| $np(3, sing, subj)$ | \rightarrow | he, she |
| $np(3, sing, C)$ | \rightarrow | it |
| $np(3, sing, obj)$ | \rightarrow | him, her |
| $np(3, plu, subj)$ | \rightarrow | $they$ |
| $np(3, plu, obj)$ | \rightarrow | $them$ |

It can be seen from this list that pronouns, unlike proper names, can take a wide variety of person, number and case agreement features. This will allow us to account for their pattern of distribution in verb phrases and modifiers when we come to discuss the uses of noun phrases as complements.

Nouns and Determiners

Besides being composed of a basic expression, a noun phrase can consist of a determiner followed by a noun that agrees with the determiner in number. The result will be a third person noun phrase which can show up in subject or object position. For nouns, we take the following lexical entries:

- (14)
- | | | |
|-----------|---------------|-----------------------|
| $n(sing)$ | \rightarrow | $kid, man, penguin$ |
| $n(plu)$ | \rightarrow | $kids, men, penguins$ |
| $n(N)$ | \rightarrow | $sheep, fish$ |

Thus we can see that there are nouns which are singular, those which are plural and those which can be both. Notice that since the category $n(N)$ really has two possible instantiations, we are not committed to providing the same semantic constant for both the $n(sing)$ and $n(plu)$ entries of a noun like *sheep*.

Determiners are our first example of a functional category. We classify determiners as functors which take a noun argument to their right to produce a noun phrase. This gives us the following lexical entries:

- (15)
- | | | |
|----------------------------|---------------|---------------|
| $np(3, sing, C) / n(sing)$ | \rightarrow | $every, a$ |
| $np(3, plu, C) / n(plu)$ | \rightarrow | $most, three$ |
| $np(3, N, C) / n(N)$ | \rightarrow | the |

In the entry for *the* we must pick one value for the number feature N and use it in both places, thus getting an entry for *the* which is of the same syntactic category as *most* and another which is of the same category as *every*.

We are now in a position to provide some phrase structure analyses. We make the standard assumptions about admissible trees under a given lexicon and set of phrase-structure rules. We have the following analysis of the noun phrase *every kid*:

- (16)
- | | |
|---|--|
| $every$ | kid |
| $\frac{np(3, sing, C) / n(sing) : \mathbf{every}}{\quad}$ | $\frac{n(sing) : \mathbf{kid}}{\quad}$ |
| $\frac{\quad}{np(3, sing, C) : \mathbf{every(kid)}}$ | |

We have used a categorial grammar notation for phrase-structure trees due to Steedman, with the root of the tree at the bottom and the leaves along the top, with lines spanning the branches. We will include feature variables in trees if the tree is admissible for *any* possible substitution of features.⁶ Using the grammar as it stands, it is not possible to derive a category for strings such as ** three kid* or ** every men* (we employ the standard notation of marking ungrammatical strings with an asterisk). Similarly, *the kid* could only be analyzed as belonging to the category $np(3, sing, C) : \mathbf{the(kid)}$, and not to the category $np(3, plu, C) : \mathbf{the(kid)}$, since the determiner *the* can not be instantiated to the category $np(3, plu, C) / n(sing)$ since the same variable occurs in the noun phrase and noun in the lexical entry for *the*.

It is interesting to note that this is not the only possible categorial analysis of noun phrase structure. Another possibility which immediately presents itself is illustrated by the following two potential lexical entries:

$$(17) \quad \begin{array}{ll} det(sing) & \rightarrow \textit{every} \\ np(3, N, C) \setminus det(N) & \rightarrow \textit{sheep} \end{array}$$

Note that (17) assumes that determiners are assigned to a basic category and that nouns are analyzed functionally. This assumption will lead to unsightly categories both syntactically and semantically even in the simple cases of nominal modifiers such as prepositional phrases and adjectives. While (17) would provide the same distributional analysis as the one presented so far, it turns out to be much simpler from the semantic point of view to interpret nouns as basic categories and treat determiners functionally. In extended categorial grammars, type-lifted analyses are often automatically generated for nouns such as *kid* in which they are assigned the category:

$$(18) \quad np(P, N, C) \setminus (np(P, N, C) / n(N)) : \lambda D^{\langle n, np \rangle}.D(\mathbf{kid}).$$

In λ -abstracts such as this, we will write the type of a variable as a superscript in the abstraction, usually omitting the features for the sake of readability.

3.2 Simple Verb Phrases

We will classify verb phrases functionally as categories which form sentences when they are combined with a noun phrase to their left. Note that the verbal agreement properties of a verb phrase are marked on its sentential result and its nominal agreement properties are marked on its noun phrase argument. Typically, unification categorial grammars based on feature structures allow features to be marked directly on a functional category and not require them to be reduced to a pattern of markings on basic categories (Karttunen 1986, Uszkoreit 1986, Zeevat, Klein and Calder 1987). This liberal distribution of features on functional categories is also found in the head-driven phrases structure grammars (HPSG) of Pollard and Sag (1987). We take the more conservative approach in this paper of only allowing features to be marked on basic categories, assuming that the more general approach could be adopted later if it were found to be necessary to express certain types of syntactic distinctions. Explicit conventions such as the head feature principles of GPSG and HPSG will be implicitly modeled by the distribution of features in functional categories such as transitive verbs, relative pronouns and modifiers.

⁶Logically, this means that we give the variables universal readings.

Intransitive Verbs

The simplest kind of verb phrase consists of a single intransitive verb. The categorization for a simple base form intransitive verb is as follows:

$$(19) \quad s(bse) \setminus np(P, N, subj) \rightarrow \text{sneeze, run, sing}$$

Finite form verb phrases show the following agreement classes:

$$(20) \quad \begin{aligned} s(fin) \setminus np(3, sing, subj) &\rightarrow \text{sneezes, runs, sings} \\ s(fin) \setminus np(2, N, subj) &\rightarrow \text{sneeze, run, sing} \\ s(fin) \setminus np(1, N, subj) &\rightarrow \text{sneeze, run, sing} \\ s(fin) \setminus np(P, plu, subj) &\rightarrow \text{sneeze, run, sing} \end{aligned}$$

Finally, there are predicative and perfective entries for simple verbs:

$$(21) \quad \begin{aligned} s(pred) \setminus np(P, N, subj) &\rightarrow \text{sneezing, running, singing} \\ s(perf) \setminus np(P, N, subj) &\rightarrow \text{sneezed, run, sung} \end{aligned}$$

There are no basic lexical entries with the verb form *inf*; we make the assumption common in unification grammars that *to* is categorized as an auxiliary that takes a *bse* form verb phrase argument to produce an *inf* form result. Note that three separate listings are necessary for the non-third-singular finite verbs. Grammars that actually allow features to be manipulated in a sensible way, such as HPSG, will not have this problem, which really only arises due to our simplified treatment of categories as atomic objects. A common way to express the lexical entry for *sneeze* would be using logical descriptions as in:

$$(22) \quad \begin{aligned} s(V) \setminus np(P, N, subj) &\rightarrow \text{sneeze} \\ \text{where} & \\ V = bse \text{ or} & \\ V = fin \text{ and } (P = 1 \text{ or } P = 2 \text{ or } N = plu) & \end{aligned}$$

For instance, see Pollard and Sag (1987) for this kind of logical treatment of lexical entries in HPSG. In any case, as far as the lexicon is concerned, there are really just 13 fully specified lexical entries for *sneeze*, corresponding to the assignments of values to variables in the lexical entry that satisfy the logical description.

We can now analyze simple finite sentences as follows:

$$(23) \quad \frac{\frac{\text{opus}}{np(3, sing, subj) : \mathbf{opus}} \quad \frac{\text{sneezed}}{s(fin) \setminus np(3, sing, subj) : \mathbf{sneezed}}}{s(fin) : \mathbf{sneezed}(\mathbf{opus})}$$

Note that we will also be able to analyze non-finite sentences such as *Opus running* (which are sometimes referred to as *small clauses*) as being of category $s(pred)$ using our lexical entries. We make the contentious, but non-problematic, assumption that all verbs mark their subjects for case. This means that we will be able to analyze the string *he running*, but not the string **him running* as being of category $s(pred)$. Particular claims about these so-called *small clause* analyses may differ with respect to the case assigned to the complement noun phrases in predicative verbal lexical entries. The reason that we will not suffer any problems on account of this decision is that we assume that control verbs such as *persuade* independently take arguments corresponding to the object noun phrase and infinitive verb phrase in verb phrases such as *believe him to be running*. Thus, the main verb will be assigning case to the *him* which semantically plays the role of the subject of the running.

3.3 Basic Modifiers

In categorial grammars, a modifier will be any category which eventually combines with arguments to produce a category of the form α / α or $\alpha \setminus \alpha$, which are called *saturated modifier* categories. We can formalize the notion of eventually producing a saturated modifier in terms of categories that have applicative results which are saturated modifiers. We define the notion of *applicative result* by the following clauses:

- (24) • α is (trivially) an *applicative result* of α
- γ is an *applicative result* of α / β or $\alpha \setminus \beta$ if γ is an applicative result of α .

A *modifier* will then be any category with an applicative result of the form α / α or $\alpha \setminus \alpha$.

An important fact to keep in mind concerning the behavior of modifiers is that they iterate. The iteration occurs in the following two schemes instantiating the application schemes:

- (25) • $\alpha / \alpha : \mathbf{f} \quad \alpha : \mathbf{g} \quad \rightarrow \quad \alpha : \mathbf{f}(\mathbf{g})$
- $\alpha : \mathbf{g} \quad \alpha \setminus \alpha : \mathbf{f} \quad \rightarrow \quad \alpha : \mathbf{f}(\mathbf{g})$

Thus any number of expressions of category α / α may precede an expression of category α , and similarly any number of expressions of category $\alpha \setminus \alpha$ may follow an expression of category α .

Basic Adjectives and Intensifiers

The simplest kind of modifier is the basic adjective. In English, adjectives combine with nouns to their right to produce a noun result. We thus have the following categorization:

- (26) $n(N) / n(N) \rightarrow red, tall, fake.$

This gives us the following analysis of a simple adjective-noun construction:

- (27)
$$\frac{\frac{red}{n(sing) / n(sing) : \mathbf{red}} \quad \frac{herring}{n(sing) : \mathbf{herring}}}{n(sing) : \mathbf{red}(\mathbf{herring})}$$

Note that the number feature on both the argument and result categories in the adjectives is identical. This will insure that an adjective-noun construction will be given the same number feature as its noun. In other languages where adjectives are marked for agreement, an adjective might have restrictions placed on its number value. But it is important to note that there is no separate marking for number on adjectives other than those that occur in its argument and result categories.

Intensifiers can be thought of as modifiers of adjectives, so that they take adjectives as arguments to produce adjectives as results. This gives us the lexical entries:

- (28) $n(N) / n(N) / (n(N) / n(N)) \rightarrow very, quite.$

We can use this lexical assignment to provide the following analysis of nominal intensifiers:

- (29)
$$\frac{\frac{\frac{very}{n(sing) / n(sing)} : \mathbf{very}}{/ (n(sing) / n(sing))} \quad \frac{red}{n(sing) / n(sing) : \mathbf{red}} \quad \frac{herring}{n(sing) : \mathbf{herring}}}{n(sing) / n(sing) : \mathbf{very}(\mathbf{red})} \\ \frac{}{n(sing) : \mathbf{very}(\mathbf{red})(\mathbf{herring})}$$

In this example, we have stacked complements to conserve space. We will not otherwise change our bracketing conventions.

Basic Adverbs

Adverbs are only slightly more complicated than adjectives in that they modify verb phrase categories matching the scheme $s(V) \setminus np(P, N, C)$. Adverbs also show up in both the pre-verbal and post-verbal positions, with some restrictions on their distributions. We can account for these facts with the following lexical entries:

$$(30) \quad \begin{array}{l} s(V) \setminus np(P, N, C) / (s(V) \setminus np(P, N, C)) \rightarrow \textit{probably, willingly, slowly} \\ s(V) \setminus np(P, N, C) \setminus (s(V) \setminus np(P, N, C)) \rightarrow \textit{yesterday, willingly, slowly} \end{array}$$

These entries will insure that modal adverbs like *probably* only occur before verb phrases, temporal adverbials like *yesterday* show up only after verb phrases and that manner adverbs such as *slowly* and *willingly* can show up in either position. Again, rather than stating two entries for *willingly* and *slowly*, logical unification mechanisms can be employed to capture the generalization of directedness by employing a feature for the direction of the complement (this approach is taken in Zeevat, Klein and Calder (1987)). The following is an instance of the way in which adverbs function:

$$(31) \quad \frac{\frac{\textit{probably}}{s(pred) \setminus np(P, N, C) / (s(pred) \setminus np(P, N, C))} \quad \frac{\textit{cheating}}{s(pred) \setminus np(P, N, C)}}{\textit{probably} \quad \textit{cheating}}}{s(pred) \setminus np(P, N, C) : \textit{probably(cheating)}}$$

Remember that the variables occurring in the trees are interpreted in such a way that any substitution will yield an admissible tree. Just as with the adjectives, the features on the verb phrase are percolated up to the result due to the identity between the features in the argument and result categories of adverbials. Backward looking adverbs will behave in the same way.

Intensifiers for adverbs work the same way as intensifiers for adjectives, but we will refrain from listing their category as it is of the form:

$$(32) \quad s \setminus np / (s \setminus np) / (s \setminus np / (s \setminus np))$$

with the additional complications of feature equivalences and the fact that verbal intensifiers can also modify post-verbal adverbs.

Prepositional Phrases

Prepositional phrases provide our first example of modifiers which take arguments. We will first consider prepositional phrases in their nominal modifier capacity and then look at their similar role in the verb phrase. To allow prepositional phrases to act as post-nominal modifiers of nouns, we give them the following lexical entry:

$$(33) \quad n(N) \setminus n(N) / np(P, N2, obj) \rightarrow \textit{in, with, beside}$$

Thus, an expression categorized as an object position noun phrase following a preposition will create a nominal modifier. A simple prepositional phrase will be analyzed as in:

$$(34) \quad \frac{\frac{\textit{beside}}{n(N) \setminus n(N) / np(3, sing, obj) : \textit{beside}} \quad \frac{\textit{opus}}{np(3, sing, obj) : \textit{opus}}}{n(N) \setminus n(N) : \textit{beside(opus)}}$$

There are convincing semantic arguments for treating prepositional phrases as attaching to nouns rather than to noun phrases (which would also be a possible categorization), since they fall within the scope of quantificational determiners in examples such as:⁷

(35) Every [student in the class] is bored.

In this case, the universal quantification introduced by *every* is restricted to students in the class.

Since prepositional phrases occur after nouns, they will create well-known structural ambiguities when used in conjunction with adjectives. This is evidenced by the following parse trees:

$$(36) \quad \frac{\frac{\text{tall}}{n(\textit{sing}) / n(\textit{sing}) : \mathbf{tall}} \quad \frac{\textit{penguin}}{n(\textit{sing}) : \mathbf{penguin}} \quad \frac{\textit{beside opus}}{n(\textit{sing}) \setminus n(\textit{sing}) : \mathbf{beside}(\mathbf{opus})}}{n(\textit{sing}) : \mathbf{tall}(\mathbf{penguin})}}{n(\textit{sing}) : \mathbf{beside}(\mathbf{opus})(\mathbf{tall}(\mathbf{penguin}))}$$

$$(37) \quad \frac{\frac{\text{tall}}{n(\textit{sing}) / n(\textit{sing}) : \mathbf{tall}} \quad \frac{\textit{penguin}}{n(\textit{sing}) : \mathbf{penguin}} \quad \frac{\textit{beside opus}}{n(\textit{sing}) \setminus n(\textit{sing}) : \mathbf{beside}(\mathbf{opus})}}{n(\textit{sing}) : \mathbf{beside}(\mathbf{opus})(\mathbf{penguin})}}{n(\textit{sing}) : \mathbf{tall}(\mathbf{beside}(\mathbf{opus})(\mathbf{penguin}))}$$

The structural ambiguity is reflected in the semantics at the root of the trees. In these cases of structural ambiguity, the lexical semantics assigned to the modifiers might be such that there is no resulting semantic ambiguity.

Prepositions are unlike simple adverbs and adjectives in that they apply to both nouns and verb phrases. Since we have categorized nouns as category n and assigned verb phrases to the major category $s \setminus np$, we will have to assign prepositional phrases to two distinct categories, one of which is the noun modifying category which we have already seen, and the second of which is the following verb modifying categorization:⁸

$$(38) \quad s(V) \setminus np(P, N, C) \setminus (s(V) \setminus np(P, N, C)) / np(P2, N2, obj) \rightarrow \textit{in, with, beside}$$

This lexical entry will allow prepositions to occur as post-verbal modifiers, thus allowing a prepositional phrase such as *in Chicago* to show up in all of the places that any other post-verbal modifier such as *yesterday* would. Note that there is a different person and number assigned to the object of the prepositional phrase than the subject of the sentence through the modifying category. This does not require the person and number of the prepositional object and modified verb phrase to be different, but simply states that they do not have to be identical.

It is also possible to have prepositional phrases that do not take noun phrase complements. These prepositions have the following simplified lexical entries:

$$(39) \quad \begin{array}{ll} s(V) \setminus np(P, N, C) \setminus (s(V) \setminus np(P, N, C)) & \rightarrow \textit{inside, outside} \\ n(N) \setminus n(N) & \rightarrow \textit{inside, outside} \end{array}$$

⁷See Partee (1975) for an argument in favor of the analysis presented here. Bach and Cooper (1978) later showed how the underlying semantic scoping could be recovered with an *np*-modifier categorization of prepositional phrases.

⁸The use of distinct variables P and $P2$ indicates possibly different values of the person feature. But keep in mind that all occurrences of the same variable must be identical.

Note that again we must provide two lexical schemes for these prepositions, one for their role as nominal modifiers and one for their role as verbal modifiers.⁹

There has been some discussion regarding the actual category that is being modified by verbal prepositional phrases. While there is strong semantic evidence that the prepositional phrase is a noun modifier in the nominal case, there is really nothing semantically that points toward verb phrase as opposed to sentential modification. Thus, a possible lexical entry for the verbal prepositional phrase would be:

$$(40) \quad s(V) \setminus s(V) / np(P, N, obj) \quad \rightarrow \quad in, with, beside$$

One difficulty in settling this issue is the fact that in extended categorial grammars, the verb phrase modifier categorization follows from the sentential modifier categorization. Rather than assuming that the matter is settled, we simply assume the now more or less standard verb phrase modifier category for prepositional phrases at the lexical level.

3.4 Auxiliaries

In this section we present a categorial treatment of auxiliaries in English along the lines of Gazdar *et al.* (1982) and Bach (1983b). The sequencing and subcategorization requirements of auxiliaries are directly represented in our lexical entries. It is assumed that an auxiliary category will take a verb phrase argument of one verb form and produce a result of a possibly different verb form and a possibly more restricted assignment of feature values. The semantic behavior of auxiliaries can also be captured naturally within this system (see Carpenter 1989).

Modal and Temporal

The simplest auxiliaries are the *temporal* auxiliaries *do*, *does* and *did* and the *modal* auxiliaries such as *will*, *might*, *should* and *could*. These auxiliaries always produce a finite verb phrase as a result and take as arguments base form verb phrases such as *eat* or *eat yesterday in the park*.

The forms of *do* all act as temporal adverbs and can be captured by the lexical entries:

$$(41) \quad \begin{array}{ll} s(fin) \setminus np(P, N, subj) / (s(bse) \setminus np(P, N, subj)) & \rightarrow \quad did \\ s(fin) \setminus np(3, sing, subj) / (s(bse) \setminus np(3, sing, subj)) & \rightarrow \quad does \\ s(fin) \setminus np(P, N, subj) / (s(bse) \setminus np(P, N, subj)) & \rightarrow \quad do \\ \text{where } (P \neq 3 \text{ or } N \neq sing) & \end{array}$$

In the last lexical entry we have restricted the values of the person and number feature so that they can be anything but the combination of third person and singular. Notice that the argument verb phrase and result verb phrase categories share their person and number features. This will be true of all auxiliary entries, some of which may in addition restrict the person and number features as is found with *does* and *do*.

The modal adverbs are syntactically distributed in exactly the same way as the temporal adverbs in terms of verb form, but they are not marked for any nominal agreement. We thus provide the following lexical entries:

$$(42) \quad s(fin) \setminus np(P, N, subj) / (s(bse) \setminus np(P, N, subj)) \quad \rightarrow \quad will, should, might$$

⁹A uniform treatment of prepositional phrase semantics under these syntactic assignments is provided in Carpenter (1989) with an event-based sentential semantics.

Using these auxiliary categories we get the following tree:

$$(43) \quad \frac{\frac{\text{do}}{s(fin) \setminus np(3, plu, subj) : \mathbf{do}} \quad \frac{\text{swim}}{s(bse) \setminus np(3, plu, subj) : \mathbf{swim}}}{s(fin) \setminus np(3, plu, subj) : \mathbf{do(swim)}}$$

The verb phrase *do run* could then combine with a third person plural subject such as *the penguins* to form a finite sentence.

Predicative

The various forms of *be*, often referred to as the *copula*, can be used before predicative verb phrases such as *eating the herring* to produce verb phrases displaying the entire range of verb forms. We will take the following lexical entry for the base form of the copula:

$$(44) \quad s(bse) \setminus np(P, N, subj) / (s(pred) \setminus np(P, N, subj)) \rightarrow be$$

The predicative auxiliary displays the full range of nominal agreement as can be seen from the following finite lexical entries:

$$(45) \quad \begin{array}{ll} s(fin) \setminus np(3, sing, subj) / (s(pred) \setminus np(3, sing, subj)) & \rightarrow is \\ s(fin) \setminus np(1, sing, subj) / (s(pred) \setminus np(1, sing, subj)) & \rightarrow am \\ s(fin) \setminus np(P, N, subj) / (s(pred) \setminus np(P, N, subj)) & \rightarrow are \\ \text{where } N = plu \text{ or } P = 2 & \\ s(fin) \setminus np(P, sing, subj) / (s(pred) \setminus np(P, sing, subj)) & \rightarrow was \\ \text{where } P = 1 \text{ or } P = 3 & \\ s(fin) \setminus np(P, N, subj) / (s(pred) \setminus np(P, N, subj)) & \rightarrow were \\ \text{where } N = plu \text{ or } P = 2 & \end{array}$$

The auxiliary verbs are unusual in that they have irregular inflectional patterns, as is displayed by *am*, *are*, *was* and *were*.

Finally, we have predicative and perfective forms of *be*, which display the following categories:

$$(46) \quad \begin{array}{ll} s(pred) \setminus np(P, N, subj) / (s(pred) \setminus np(P, N, subj)) & \rightarrow being \\ s(perf) \setminus np(P, N, subj) / (s(pred) \setminus np(P, N, subj)) & \rightarrow been \end{array}$$

Consider the example parse trees:

$$(47) \quad \frac{\frac{\text{am}}{s(fin) \setminus np(1, sing, subj) / (s(pred) \setminus np(1, sing, subj))} \quad \frac{\text{eating}}{s(pred) \setminus np(1, sing, subj)}}{s(fin) \setminus np(1, sing, subj)}$$

$$(48) \quad \frac{\frac{I}{np(1, sing, subj)} \quad \frac{\text{am eating}}{s(fin) \setminus np(1, sing, subj)}}{s(fin)}$$

From now on we will suppress the semantic constants as we did in this example, since they are fully determined by the phrase-structure schemes and lexical constants.

We will see more of the predicative auxiliary when we consider lexical rules and give an account of the range of possible predicative complements. One benefit of our analysis is that we will not have to provide further lexical entries for the copula. GPSG and LFG treat the copula as a degenerate auxiliary which is not marked for any feature in its complement other than $PRED : +$ to explicitly mark the predicative aspect and $BAR : 2$ to restrict attention to saturated phrases (corresponding to maximal projections). These are features that show up on adjectives, progressive verb phrases, noun phrases and prepositional phrases, among others. This option of only restricting the complement by a few features is not open to us in a strict categorial grammar framework; the adjectives, verb phrases and adverbial phrases simply do not share a single category and could thus not be provided with a single syntactic lexical entry (although a single semantic constant might be provided for all of the different lexical entries in a mono-typed semantics such as that provided by Chierchia and Turner (forthcoming)).

Perfective

The various forms of *have* take perfective verb phrase arguments and produce a range of results, much like the predicative auxiliary forms. The base form perfective auxiliary entry is:

$$(49) \quad s(bse) \setminus np(P, N, subj) / (s(perf) \setminus np(P, N, subj)) \rightarrow have$$

Notice that like the other auxiliaries, the only function of the base form *have* is to carry along the nominal features and shift the verb form of its argument. The inflected forms will all be restrictions of this category and include the following:¹⁰

$$(50) \quad \begin{array}{ll} s(fin) \setminus np(P, N, subj) / (s(perf) \setminus np(P, N, subj)) & \rightarrow had \\ s(fin) \setminus np(3, sing, subj) / (s(perf) \setminus np(3, sing, subj)) & \rightarrow has \\ s(fin) \setminus np(P, N, subj) / (s(perf) \setminus np(P, N, subj)) & \rightarrow have \\ \text{where not } (P = 3 \text{ and } N = sing) & \end{array}$$

These lexical entries will allow the following phrase-structure tree:

$$(51) \quad \frac{\frac{\frac{have}{s(fin) \setminus np(P, N, subj) / (s(perf) \setminus np(P, N, subj))} \quad \frac{eaten}{s(bse) \setminus np(P, N, subj)}}{s(bse) \setminus np(P, N, subj)}}$$

Note that we can analyze even longer sequences of auxiliaries such as *will have been eating* without overgenerating parse trees for syntactically ill-formed sequences.

Infinitive

Following Gazdar *et al.* (1982), we will categorize *to* as a special kind of auxiliary without finite forms. The category we assign to *to* will produce infinitive form verb phrases from base form verb phrases, thus requiring the lexical entry:

$$(52) \quad s(inf) \setminus np(P, N, subj) / (s(bse) \setminus np(P, N, subj)) \rightarrow to$$

This entry will result in expressions such as *to eat* being categorized as infinitive verb phrases without person or number restrictions. Note that the categorization that we provide will allow us to wantonly generate split infinitives.

¹⁰The auxiliary forms of *have* should not be confused with the homonymous transitive verb *have* as in *having fun*.

We will only consider the role of infinitives as providing clausal complements to control verbs such as *promise* and *believe*. We do not discuss the use of infinitives in sentential subject constructions such as:

- (53) a. [to run] is fun.
 b. [(for us) to vote] would be useless.

We will also not discuss the role of *for* in optionally providing subjects to infinitive clauses. *To* also occurs with base form verbs in purpose clauses and infinitival relatives as described in Gazdar *et al.* (1985):

- (54) a. The man [(for us) to meet with] is here.
 b. They are too crazy [(for us) to meet with].

In all of these cases, there is an unbounded dependency construction where a noun phrase is missing from the infinitival verb phrase.

Negative

For lack of a better section in which to include the negative modifier, we include it under our treatment of auxiliaries. We take lexical entries for *not* according to the following scheme:

- (55) $s(V) \setminus np(P, N, C) / (s(V) \setminus np(P, N, C)) \rightarrow not$
 where $V \neq fin$

This accounts for the fact that negation can be applied to any form of verb phrase other than finite. Consider the following analysis:

- (56)
$$\frac{\frac{not}{s(bse) \setminus np(P, N, subj) / (s(bse) \setminus np(P, N, subj))} \quad \frac{sing}{s(bse) \setminus np(P, N, subj)}}{s(bse) \setminus np(P, N, subj)}$$

- (57)
$$\frac{\frac{does}{s(fin) \setminus np(3, sing, subj) / (s(bse) \setminus np(3, sing, subj))} \quad \frac{not \ sing}{s(bse) \setminus np(3, sing, subj)}}{s(fin) \setminus np(3, sing, subj)}$$

It should be noted that the category for the negative is actually a modifier and takes the verb form of its argument as the resulting verb form and carries along person and number features. This will allow us to correctly categorize expressions involving nested auxiliaries and negations such as:

- (58) Opus would not have been not eating herring.

3.5 Complemented Categories

In this section we will consider a number of additional lexical entries which produce categories that we have already discussed as applicative results.

Simple Polytransitive Verbs

A verb may take noun phrase arguments in addition to the subject, as is evidenced by transitive and ditransitive verbs such as *hit* and *give*. These verbs have the following lexical entries:

$$(59) \quad \begin{array}{l} s(bse) \setminus np(P, N, subj) / np(P2, N2, obj) \quad \rightarrow \quad hit \\ s(bse) \setminus np(P, N, subj) / np(P2, N2, obj) / np(P3, N3, obj) \quad \rightarrow \quad give \end{array}$$

From now on, we will only present verbs in their base form; the inflected categories of these verbs is wholly determined by their base forms, as only the verb form of the final sentential result changes. Notice that the complements other than the subject have to be noun phrases with object case marking. We thus capture the following simple contrast:

- (60) a. Opus gave him the herring.
 b. * Opus gave he herring.
 c. He hit Opus.
 d. * Him hit Opus.

In keeping with our methodology of distinguishing category assignments by means of surface distribution, we will distinguish between different verbs in terms of their lexical entries. In government-binding theory, on the other hand, all verbs are assigned the same basic lexical category V according to \bar{X} -theory and analyzed with identical phrase-structure rule instances. A verb phrase such as *sneezed* might be analyzed as

$$(61) \quad [[[run]_V]_{V'}]_{V''}$$

and a verb-phrase such as *hit Opus* might be analyzed as

$$(62) \quad [[[hit]_V [opus]_{N''}]_{V'}]_{V''}$$

The reason that the proponents of government-binding theory are led to this kind of analysis is because they adhere to a convention of assigning verbs such as *sneeze* and *hit* that require different numbers of arguments, to the same lexical category assigned to all verbs.¹¹ The number of arguments that a particular verb requires is also marked in government-binding theory, but this information is not dealt with in terms of differing phrase structure rules, but by an independent module called the θ -criterion, whose job it is to filter out analyses in which a lexical head is assigned to an inappropriate number of complements or complements go unassigned.¹² The fact that we allow different verbs to be assigned to different categories in which complements are determined directly greatly reduces the depth and complexity of the resulting phrase-structure trees. Of course, we must assume that there is some principled method of assigning lexical categories to basic expressions in the same way that government-binding theory must assume that there is some method of determining the θ -roles appropriate for each each basic expression. One step in the direction of determining θ -roles in a principled way has been taken within the LFG framework (see Bresnan (1982c) and Levin (1987)).

¹¹The problem is compounded in most recent versions of government-binding theory, which require maximal projections for agreement, inflection and sometimes even for tense and aspect.

¹²A similar factoring of labor between thematic role and phrase structure information is found in LFG (Kaplan and Bresnan 1982).

Sentential Complement Verbs

Besides taking noun phrase objects, a verb may take sentential complements. Consider the following lexical entries:

$$(63) \quad s(bse) \setminus np(P, N, subj) / s(fin) \rightarrow \textit{know, believe}$$

With this lexical scheme we produce analyses such as

$$(64) \quad \frac{\frac{\textit{knew}}{s(bse) \setminus np(P, N, subj) / s(fin)} \quad \frac{\textit{opus ran}}{s(fin)}}{s(fin) \setminus np(P, N, subj)}$$

Notice that the only verbs of this sort take finite sentences as complements. The subject case marking on the subject of the complement is enforced by the verb phrase within the complement sentence. For instance, we capture the following distinction:

- (65) a. Opus believed he ate.
 b. * Opus believed him ate.

This is because the sentence must be analyzed with *he ate* forming a subtree, and *ate* requires its subject argument to be marked with subject case.

There are sound lexical motivations for analyzing sentential complement verbs with the lexical entry in (63) rather than with the alternative given in (66):

$$(66) \quad s(bse) \setminus np(P, N, subj) / (s(fin) \setminus np(P2, N2, subj)) / np(P2, N2, subj) \rightarrow \textit{know}$$

The primary lexical evidence comes from rules which apply to verb complements, such as passivization and detransitivization; these rules do not apply to the subject within a sentential complement verb phrase such as *believe Opus ate*. For instance, we have:

- (67) a. Opus [[knew him] [to eat herring]].
 b. He was known to eat herring by Opus.
 c. Opus [saw [he [ate herring]]].
 d. * He was seen ate herring by Opus.

These examples also provide evidence for our analysis of control verbs below, which do allow lexical rules to act on their objects.

Complementized Sentential Complement Verbs

In this section we will consider complementized sentences and their role as complements, such as those bracketed in:

- (68) a. I believe [that Opus ate].
 b. I wonder [whether Opus ate].
 c. I persuaded Binkley [that Opus ate].
 d. I bet Binkley five dollars [that Opus ate].
 e. I prefer [that Opus eat].

Our account will follow that presented for GPSG in Gazdar *et al.* (1985). We will simply include some additional verb forms, which in this case are limited to *whether*, *that* and *thatb*. We assume the following lexical entries for the complementizers themselves:

- (69) $s(\textit{that}) / s(\textit{fin}) \rightarrow \textit{that}$
 $s(\textit{thatb}) / s(\textit{bse}) \rightarrow \textit{that}$
 $s(\textit{whether}) / s(\textit{fin}) \rightarrow \textit{whether}$

Note that the second entry for *that* takes sentential complements which are in base form. (69) allows us to produce analyses such as:

- (70)
$$\frac{\frac{\textit{whether}}{s(\textit{whether}) / s(\textit{fin})} \quad \frac{\textit{Opus}}{np(3, \textit{sing}, \textit{subj})} \quad \frac{\textit{ate}}{s(\textit{fin}) \setminus np(3, \textit{sing}, \textit{subj})}}{s(\textit{fin})}}{s(\textit{whether})}$$

We could then provide the main verbs in (68) with the following lexical entries:

- (71) $s(\textit{bse}) \setminus np(P, N, \textit{subj}) / s(\textit{that}) \rightarrow \textit{believe, know}$
 $s(\textit{bse}) \setminus np(P, N, \textit{subj}) / s(\textit{that}) / np(P2, N2, \textit{obj}) \rightarrow \textit{persuade}$
 $s(\textit{bse}) \setminus np(P, N, \textit{subj}) / s(\textit{that}) / np(P2, N2, \textit{obj}) / np(P3, N3, \textit{obj}) \rightarrow \textit{bet}$
 $s(\textit{bse}) \setminus np(P, N, \textit{subj}) / s(\textit{whether}) \rightarrow \textit{wonder}$
 $s(\textit{bse}) \setminus np(P, N, \textit{subj}) / s(\textit{thatb}) \rightarrow \textit{prefer}$

These entries provide the means to analyze all of the sentences in (68) with complementized sentential complements with different features. Complementized sentences display different behavior from sentences found without complementizers, as can be seen in the case of unbounded dependency constructions in relative clauses, such as:

- (72) a. * who I believe that ran
 b. who I believe ran

Transformational theories have gone to great length to differentiate the two cases, beginning with Chomsky and Lasnik's (1977) *that*-trace filter, which still surfaces in a more generalized form in current analyses employing Chomsky's government-binding framework.

Prepositional Complement Verbs

In this section we study the role of prepositions as case-markers in sentences such as:

- (73) a. Opus approved [of the herring].
 b. Bill gave the herring [to Opus].
 c. Bill bought the herring [for Opus].
 d. Opus talked [to Bill] [about the herring].
 e. Opus conceded [to Binkley] that Bill stunk.
 f. Binkley required [of Opus] that he eat.

Following the GPSG analysis (Gazdar *et al.* 1985), we will assume that the bracketed prepositions in (73) are simply marking the thematic participant roles of the verbs' arguments. Note that the traditional analysis of (73)b takes the *to* to be a dative case marker.

Under this analysis, we will simply have prepositions take noun phrase arguments and return specially case marked noun phrases such as $np(3, sing, to)$ or $np(2, plu, for)$. We use the following lexical entries:

$$(74) \quad \begin{array}{l} np(P, N, for) / np(P, N, obj) \quad \rightarrow \quad for \\ np(P, N, to) / np(P, N, obj) \quad \rightarrow \quad to \end{array}$$

We must include parallel entries for the prepositional complementizers such as *about*, *of* and *with*. This will allow us to produce parse trees for prepositional complements such as:

$$(75) \quad \frac{\frac{for}{np(2, plu, for) / np(2, plu, obj)} \quad \frac{you}{np(2, plu, obj)}}{np(2, plu, for)}$$

We can now simply subcategorize our prepositional complement verbs for the types of prepositional arguments that they take, as in the following:

$$(76) \quad \begin{array}{ll} s(bse) \setminus np(P, N, subj) / np(P2, N2, of) & \rightarrow \quad approve \\ s(bse) \setminus np(P, N, subj) / np(P2, N2, to) / np(P3, N3, obj) & \rightarrow \quad give \\ s(bse) \setminus np(P, N, subj) / np(P2, N2, for) / np(P3, N3, obj) & \rightarrow \quad buy \\ s(bse) \setminus np(P, N, subj) / np(P2, N2, about) / np(P3, N3, to) & \rightarrow \quad talk \\ s(bse) \setminus np(P, N, subj) / s(that) / np(P2, N2, to) & \rightarrow \quad conceded \\ s(bse) \setminus np(P, N, subj) / s(thatb) / np(P2, N2, of) & \rightarrow \quad required \end{array}$$

It is important to keep in mind the order in which objects are sequenced. In pure categorial grammars, a functor consumes arguments which are closest to it first. Taking the lexical entries in (76) would then allow the following analysis of a ditransitive dative verb like *give*:

$$(77) \quad \frac{\frac{give \ the \ herring}{s(fin) \setminus np(P, N, subj) / np(P, N, to)} \quad \frac{to \ opus}{np(3, sing, to)}}{s(fin) \setminus np(P, N, subj)}$$

Note that we will need two distinct lexical entries to get the different orderings displayed in:

- (78) a. Opus talked to Binkley about herring.
b. Opus talked about herring to Binkley.

Another alternative would be to treat one of the orderings as basic and assume that the other orderings are derived with some sort of heavy constituent shifting rule.

It is important to keep in mind the distinction between prepositional complements, preposition-like expressions occurring in idiomatic verbs such as *hang up* and freely occurring modifiers such as *walk under*. Consider the following examples:

- (79) a. I hung up the phone.
b. I hung the phone up.
c. The phone was hung up.

- (80) a. I gave the herring to Opus.
 b. * Opus was given the herring to.
 c. * Opus was given to the herring.
- (81) a. I walked under the bridge.
 b. The bridge was walked under.

Bresnan (1972, 1982c) argued that particles such as *up* and even free modifiers such as *under* could, in some situations, form lexical compounds with closely related verbs like *hang* and *walk*, thus allowing lexical rules like passive to operate over the results. One possibility in this case would be to follow GPSG (Gazdar *et al.* 1985) and use categories such as the following:

$$(82) \quad \begin{array}{l} s(bse) \setminus np(P, N, subj) / np(P2, N2, obj) / part(up) \quad \rightarrow \quad hang \\ part(up) \quad \rightarrow \quad up \end{array}$$

So-called “particle movement” as displayed in (79)b could then be treated in exactly the same manner as other instances of heavy noun phrase shift (see Morrill 1988). The proper treatment of phrasal verbs and verbs with particles, as found in (79) and (81) is far from settled, so we have simply demonstrated how prepositional phrases can be analyzed as case-marked noun phrases.

Expletive Subject Verbs

Some verbs do not take normal subject noun phrases, and instead, require that their subjects be the expletive *it*. For instance, consider the case of:

- (83) a. It is raining.
 b. * Opus is raining.

We will introduce a special case marking *it* to indicate the expletive subject in the lexical entry:

$$(84) \quad np(3, sing, it) \quad \rightarrow \quad it$$

We then assume that an expletive subject verb like *raining* will be subcategorized for just this kind of subject, which leads to the lexical entry:

$$(85) \quad s(bse) \setminus np(3, sing, it) \quad \rightarrow \quad rain$$

We give these verbs third singular marking simply because that is their agreement pattern and will allow us to make other lexical entries more uniform, but not because we believe that they are in some sense assigned a person or number. With these lexical entries we will get the following analysis tree:

$$(86) \quad \frac{\frac{it}{np(3, sing, it)} \quad \frac{rained}{s(fin) \setminus np(3, sing, it)}}{s(fin)}$$

Of course, this will lead to a slight problem in that the meaning assigned will be **rained(it)**, whereas the *it* subject is really serving only as a dummy. This could be captured by assigning a constant function (with, for instance, a vacuous abstraction in the λ -term) as the semantics of *rain* and some arbitrary interpretation to the constant **it**.

To allow analyses of sentences involving both auxiliaries and expletive subjects, it is actually necessary to slightly generalize the lexical entries for the auxiliaries. Thus, where we previously allowed third person singular auxiliaries to have only subject case marking, we will also allow them to have expletive *it* case marking. Thus, for instance, we will need the additional entry:

$$(87) \quad s(bse) \setminus np(P, N, it) / (s(pred) \setminus np(P, N, it)) \rightarrow be$$

Of course, this could be marked on the original entry using a straightforward feature logic.

Control Verbs

We will take the class of *control* verbs to loosely include all verbs which take a verb phrase complement and determine the semantics of the complement verb phrase's subject, possibly in conjunction with other complements.¹³ We will not be concerned with the semantics of control verbs or how they supply subjects for their complements. It is quite easy to do so in unification grammars (see Bresnan 1982b) or with the kind of higher-order typed semantics employed here (see Klein and Sag 1985). Control verbs are traditionally classified along two dimensions. The first of these is the subject/object dimension, which determines whether the subject or direct object of the main verb controls the complement subject. The second distinction is between the so-called *raising* and *equi* verbs, the raising verbs being those which do not provide the controlling complement a thematic role, while the equi verbs do provide such a role. Examples of the various possibilities are as follows:

- | | | |
|------|-------------------|---|
| (88) | (Subject-Equi) | Opus wants to eat.
* It wants to rain. |
| | (Subject-Equi) | Opus promised Binkley to eat.
* It promised Opus to rain. |
| | (Object-Equi) | Binkley persuaded Opus to eat.
* Binkley persuaded it to rain. |
| | (Object-Equi) | Binkley appealed to Opus to eat.
* Binkley appealed to it to rain. |
| | (Subject-Raising) | Opus tends to eat.
It tends to rain. |
| | (Subject-Raising) | Opus seems to Binkley to eat.
It seems to Opus to rain. |
| | (Object-Raising) | Binkley believed Opus to eat.
Binkley believed it to rain. |

¹³These verbs have typically been referred to as *obligatory control* verbs, to distinguish them from verbs which can display control-like behavior that is not determined purely syntactically.

This distribution can be accounted for with the following lexical entries:¹⁴

- (89) $s(bse) \setminus np(P, N, subj) / (s(inf) \setminus np(P, N, subj)) / np(P2, N2, obj) \rightarrow promise, want$
 $s(bse) \setminus np(P, N, subj) / (s(inf) \setminus np(P2, N2, subj)) / np(P2, N2, obj) \rightarrow persuade$
 $s(bse) \setminus np(P, N, subj) / (s(inf) \setminus np(P2, N2, subj)) / np(P2, N2, to) \rightarrow appeal$
 $s(bse) \setminus np(P, N, C) / (s(inf) \setminus np(P, N, C)) \rightarrow tend$
 where $C = subj$ or $C = it$
 $s(bse) \setminus np(P, N, C) / (s(inf) \setminus np(P, N, C)) / np(P2, N2, to) \rightarrow seem$
 where $C = subj$ or $C = it$
 $s(bse) \setminus np(P, N, subj) / (s(inf) \setminus np(P2, N2, C1)) / np(P2, N2, C2) \rightarrow believe, want$
 where ($C1 = it$ and $C2 = it$) or ($C1 = subj$ and $C2 = obj$)

The contrast between subject and object control is expressed in terms of the agreement features of the noun phrase complement within the infinitive complement. The contrast between raising and equi control is captured by the distribution of the case marking. Presumably, these syntactic distinctions will be reflected in the meanings assigned to the control verbs. Using our lexical entries, we produce the following parse trees:

$$(90) \frac{\frac{\text{persuading}}{s(pred) \setminus np(P, N, subj)} \quad \frac{\text{opus}}{np(3, sing, obj)}}{\frac{\quad}{s(inf) \setminus np(3, sing, subj)}} \quad \frac{\quad}{np(3, sing, obj)}}{\frac{\quad}{s(pred) \setminus np(P, N, subj) / (s(inf) \setminus np(3, sing, subj))}}$$

$$(91) \frac{\frac{\text{persuading opus}}{s(pred) \setminus np(P, N, subj)} \quad \frac{\text{to run}}{s(inf) \setminus np(3, sing, subj)}}{\frac{\quad}{s(inf) \setminus np(3, sing, subj)}}}{s(pred) \setminus np(P, N, subj)}$$

Notice the contrast between the agreement between features found in previous analysis of the object control verb *persuade* and the subject control verb *promise* in the following parse trees:

$$(92) \frac{\frac{\text{promised}}{s(pred) \setminus np(P, N, subj) / (s(inf) \setminus np(P, N, subj)) / np(2, plu, obj)} \quad \frac{\text{them}}{np(2, plu, obj)}}{\frac{\quad}{s(pred) \setminus np(P, N, subj) / (s(inf) \setminus np(P, N, subj))}}$$

$$(93) \frac{\frac{\text{promised them}}{s(pred) \setminus np(P, N, subj) / (s(inf) \setminus np(P, N, subj))} \quad \frac{\text{to run}}{s(inf) \setminus np(P, N, subj)}}{\frac{\quad}{s(pred) \setminus np(P, N, subj)}}$$

We can analyze the raising verb *seem* with an expletive complement as follows:

$$(94) \frac{\frac{\text{seemed}}{s(fin) \setminus np(3, sing, it) / (s(inf) \setminus np(3, sing, it))} \quad \frac{\text{to rain}}{s(inf) \setminus np(3, sing, it)}}{\frac{\quad}{s(fin) \setminus np(3, sing, it)}}$$

¹⁴We will simply assume a basic entry for the version of *want* that does not take a direct object and for the version of *seem* that does not take a prepositionally marked noun phrase. The reason we do not try to account for detransitivization is that the rule is very selective and does not apply productively. We assume its operation is closely linked with an analysis of lexical semantics. See Bresnan (1982d) for an analysis of detransitivization and the interaction between the grammatical and thematic aspects of control.

These lexical categorizations will in fact respect the grammaticality judgements expressed in (88).

We will analyze perception verbs such as *see* and *watch* in the same manner as other object raising verbs. We thus assume the following lexical entries:

- (95) $s(bse) \setminus np(P, N, subj) / (s(pred) \setminus np(P2, N2, C1)) / np(P2, N2, C2) \rightarrow see, notice$
 where $(C1 = it \text{ and } C2 = it)$ or $(C1 = subj \text{ and } C2 = obj)$
 $s(bse) \setminus np(P, N, subj) / (s(bse) \setminus np(P2, N2, C1)) / np(P2, N2, C2) \rightarrow hear, watch$
 where $(C1 = it \text{ and } C2 = it)$ or $(C1 = subj \text{ and } C2 = obj)$

The only difference between these and the other raising verbs is the verb form of the complement clause. This will allow us to provide the correct analysis of the following sentences:

- (96) a. I saw Opus eating.
 b. I saw it raining.
 c. I probably did watch Opus eat.
 d. I will see it rain.

An interesting class of verbs that seem to share many properties of the control verbs are those that take adjectival complements such as *appear* and *look*. These can be captured by the lexical entries:

- (97) $s(bse) \setminus np(P, N, subj) / (n(N) / n(N)) \rightarrow appear, look$

These lexical entries are necessary to produce readings for sentences such as:

- (98) The herring will look red.

The more complicated entry

- (99) $s(bse) \setminus np(P, N, subj) / np(P2, N2, to) / (n(N) / n(N)) \rightarrow appear, look$

could be employed to handle sentences such as:

- (100) The herring appeared red to Opus.

Lexical Predicatives

There is an entire class of verb-like basic expressions which show up primarily as complements to the copula and as adverbial and adnominal expressions. Consider the following:¹⁵

- (101) a. angry [about the weather]
 b. afraid [that Opus sneezed]
 c. insistent [that Opus run]
 d. eager [to run]

These expressions will all be lexically categorized as having applicative results which are predicative verb phrases. They will not be inflected for other verb forms. Our lexical rules will then map the

¹⁵Note that the analysis of *easy*, which is usually contrasted with that of *eager* and other *tough*-class adnominals involves an analysis of unbounded dependency constructions and will thus fall outside the scope of this paper. See Morrill (1988) for a compatible analysis of *tough*-class adnominals.

predicative entry into the other functions that these phrases can perform. We need the following lexical entries:

- (102) $s(pred) \setminus np(P, N, subj) / np(P2, N2, about) \rightarrow angry$
 $s(pred) \setminus np(P, N, subj) / s(that) \rightarrow afraid$
 $s(pred) \setminus np(P, N, subj) / s(thatb) \rightarrow insistent$
 $s(pred) \setminus np(P, N, subj) / (s(inf) \setminus np(P, N, subj)) \rightarrow eager$

These lexical categorizations will insure that all of the examples in (101) will be able to serve as predicative verb phrases and thus act as complements to the various forms of *be*.

Sentential Adverbials

There are adverbs such as *because* and *while* which take finite sentential arguments and produce post-verbal modifiers as a result. This leads to the following lexical entries:

- (103) $s(V) \setminus np(P, N, subj) \setminus (s(V) \setminus np(P, N, subj)) / s(fin) \rightarrow because, while, after, if$

These entries allow us to provide expressions such as *because Opus ate the herring* with the same category as a simple post-verbal adverb such as *yesterday*.

Control Adverbials

Adverbs also parallel complex verbs in taking either verb phrases or a combination of a verb phrase and an object noun phrase complement. For instance, we have the following acceptable sentences:

- (104) a. Opus swam after eating.
 b. Opus probably swam to be moving.
 c. Opus swam with Binkley cheering.

For these adverbs, we assume the following lexical entries:

- (105) $s(V) \setminus np(P, N, subj) \setminus (s(V) \setminus np(P, N, subj)) \rightarrow while, before$
 $/ (s(pred) \setminus np(P, N, subj))$
 $s(V) \setminus np(P, N, subj) \setminus (s(V) \setminus np(P, N, subj)) \rightarrow to$
 $/ (s(bse) \setminus np(P, N, subj))$
 $s(V) \setminus np(P, N, subj) \setminus (s(V) \setminus np(P, N, subj)) \rightarrow with$
 $/ (s(pred) \setminus np(P2, N2, subj))$
 $/ np(P2, N2, obj)$

Notice that the first two entries are subject control adverbs in that the subject of the main clause is forced to agree with the subject of the clause embedded in the adverb, while in the last entry, the embedded verb phrase agrees with the embedded object. Of course, these facts have more effect on the semantics than the syntax, since non-finite clauses in English are not marked for agreement.

Complementized Nouns

Since we are concentrating primarily on details of the verb phrase, we will not present an extended analysis of nouns which are subcategorized for complements. Consider the following examples of complementized nouns drawn from the GPSG grammar in Gazdar *et al.* (1985):

- (106) a. love [of herring]
b. argument [with Opus] [about the herring]
c. gift [of herring] [to Opus]
d. belief [that it will rain]
e. request [that Opus eat]
f. plan [to eat]

It is possible to account for the complement behavior of all of these nouns, by simply adding the lexical entries:

- (107) $n(\textit{sing}) / np(P, N, \textit{of}) \rightarrow \textit{love}$
 $n(\textit{sing}) / np(P, N, \textit{about}) / np(P2, N2, \textit{with}) \rightarrow \textit{argument}$
 $n(\textit{sing}) / np(P, N, \textit{to}) / np(P2, N2, \textit{of}) \rightarrow \textit{gift}$
 $n(\textit{sing}) / s(\textit{that}) \rightarrow \textit{belief}$
 $n(\textit{sing}) / s(\textit{thatb}) \rightarrow \textit{request}$
 $n(\textit{sing}) / (s(\textit{inf}) \setminus np(P, \textit{sing}, \textit{subj})) \rightarrow \textit{plan}$

We will not consider the interesting phenomena of nominalization, as seen in such cases as:

- (108) a. Binkley's running
b. The running of Binkley
c. The giving of the herring to Opus by Binkley

It is not clear whether the genitive in (108)a or the prepositions in (108)b or (108)c are to be analyzed as modifiers or as complements. Either analysis would be possible within the framework we are developing here.

Another matter that is not handled by our lexical entries is the fact that prepositionally marked complements can often occur in any order. For instance, in (106)b and (106)c, we could also have

- (109) a. argument about herring with the silly penguin
b. gift to Opus of the funny little fish

It is not clear from these examples whether there is a true free word order or something like heavy noun phrase shift being employed. If it is some kind of noun phrase shift, then these examples would naturally be handled with the same syntactic mechanisms as are involved in unbounded dependency constructions (see Morrill 1987b, 1988, Gazdar *et al.* 1985). If the alternation between the examples in (106) and (109) is not an example of heavy shift, then an alternative explanation is required. One possibility would be to simply list multiple lexical entries with the different orders indicated. A similar problem is encountered in an attempt to apply the simple directed categorial grammar we employ here to free word order languages such as German (see Reape (1989) and Steedman (1985) for treatments of free word order in a categorial framework).

Possessives

We can analyze the simple possessive construction with the following lexical entry:

$$(110) \quad np(3, N, C) / n(N) \setminus np(P2, N2, obj) \quad \rightarrow \quad 's$$

This will allow us to produce the following noun phrase analysis.

$$(111) \quad \frac{\frac{Opus}{np(3, sing, C2)} \quad \frac{'s}{np(3, sing, C) / n(sing) \setminus np(3, sing, C2)}}{np(3, sing, C) / n(sing)} \quad \frac{herring}{n(sing)}}{np(3, sing, C)}$$

We have made sure that the possessive marking combines with a noun phrase to produce a determiner category which then applies to a noun to produce a noun phrase. Stricly speaking, the possessor argument can not be a pronoun, as there are special lexical entries for possessive pronouns. We will also not consider the use of genitive noun phrases such as *Opus's* where they are obviously playing some sort of complement role, in cases such as:

- (112) a. The herring of Opus's is red.
 b. Opus's giving of the herring to Bill was unwise.

We could, on the other hand, simply add another entry for the possessive marker to produce genitive noun phrase results, as in:

$$(113) \quad np(3, N, gen) \setminus np(3, N, C) \quad \rightarrow \quad 's$$

which would allow *Opus's* to be categorized as $np(3, sing, gen)$. The possessive *of* could then be lexically marked to take a genitive noun phrase object. Evidence for the fact that more than one entry is necessary for 's is given by the following two noun phrases:

- (114) a. my herring
 b. Opus's herring
 c. the herring of mine
 d. the herring of Binkley's.

Then we could classify *my* as a determiner, while *mine* would be classified as $np(3, sing, gen)$.

4 Lexical Rules

What we have presented up to this point forms the core lexicon of our grammar and consists of assignments of categories to basic expressions. We have also seen examples of how the application rules recursively determine the compositional assignment of syntactic and semantic categorizations to complex expressions in terms of the assignments to their constituent expressions. But the lexical entries that we have provided are not sufficient to account for many patterns of distribution displayed by the expressions that we have introduced. For instance, predicative verb phrases can occur as post-nominal modifiers and plural nouns can serve as noun phrases without the aid of a determiner. To account for additional categorizations which are predictable on the basis of core lexical assignments, we introduce a system of lexical rules.

Lexical rules serve the purpose of expressing lexical regularities. Functionally, they create new lexical entries which are predictable from existing lexical entries. By continuing to apply lexical rules until no more entries can be found, we generate a closed lexicon. Nothing in the lexical rule affects the application schemes, so distributional regularities must be accounted for solely in terms of additional lexical category assignments. Viewed in this way, our lexical rules serve much the same purpose as the lexical metarules of GPSG (Gazdar *et al.* 1985) and the lexical rules of LFG (Bresnan 1982). In GPSG, metarules produced new phrase structure schemes from existing ones. Our lexical categorizations, when coupled with the universal application schemes, correspond quite closely to phrase structure rules. Producing new lexical entries in a categorial grammar serves the same purpose as producing new basic categories and lexical phrase structure rules in a context-free grammar.

In categorial grammar contexts, it is common to follow Dowty’s (1982) proposal to encode grammatical role information such as the object and subject distinction directly in the category by means of argument order. Under this positional encoding of the obliqueness hierarchy, the arguments consumed earliest by a category are considered most oblique.¹⁶ For instance, in the case of the ditransitive verb category $s \setminus np_1 / np_2 / np_3$, we would have np_1 as the subject argument, np_2 as the object and np_3 as the indirect object. A transitive verb category such as $s \setminus np_1 / np_2$ would not have an indirect object, and other categories, such as those assigned to control verbs, would have phrasal or sentential complements. These assumptions concerning grammatical functions are directly incorporated into the semantic portion of GPSG (Gazdar *et al.* 1985) and into the lexical hierarchy of HPSG (Pollard and Sag 1987). The way our lexical rules are defined, they will have access to arguments of verbs in certain positions determined by obliqueness, thus giving them the power to perform operations which apply to subjects, the most oblique argument, and so on. Passive is a prime example of where this sort of re-encoding of complement order is carried out, and functionally, our lexical rules share a striking resemblance to those defined over grammatical roles such as subject and object in Lexical Functional Grammar (Bresnan 1982c). In addition, LFG lexical rules can delete noun phrase complements from lexical entries for verbs (where verbs are the heads of sentences, and thus have sentences as their applicative results, in our terminology), change the case marking of complements, alter the final result category, and so on. In this section, we will see examples of lexical rules with all of these functions.

It is significant that we have assumed a set of lexical rules rather than a set of unary rules that might apply at any stage in a derivation. By forcing our rules to apply before syntactic applications and more specifically, before any other kind of unbounded dependency construction, their effects will be strictly localized or *bounded*. That is, our rules operate solely over a lexical category and its complements. This property is also shared by the LFG lexical rules and GPSG metarules.¹⁷

We should make clear from the start that we will not be concerned with any form of inflectional or derivational morphology. We have avoided inflectional morphology by choosing an inflectionally impoverished language, namely English, for which it is possible to simply list all of the available lexical forms (sometimes using schematic entries employing variables over features). In particular,

¹⁶Our notion of obliqueness is different from that of Pollard and Sag (1987), where a distinction is made between obliqueness and order of argument combination.

¹⁷Unbounded dependencies were treated in GPSG (Gazdar *et al.* 1985) by the passing of features, some of which are *introduced* by lexical metarules. The *slash* features bear a striking similarity to the slashes in categorial grammar, and the way in which they are propagated is reminiscent of the abstraction rules applied in extended categorial grammars (Morrill 1988).

both singular and plural forms of nouns and various verb forms must be explicitly listed for each word in our core lexicon. We will similarly ignore derivational morphology such as prefixation and suffixation. Of course, it would be nice to have a characterization of inflectional and derivational morphology compatible with the lexicon presented here, but none has been worked out in detail. Nothing in the present system places any restrictions on the way that morphological or inflectional morphology might be realized. Both the approaches of Moortgat (1987c, 1988b) and Dowty (1979) to morphology in categorial grammar are compatible with the lexicon presented here. In fact, it has been argued that a categorial system is actually useful in describing derivational morphological operations (Hoeksema and Janda 1988; Keenan and Timberlake 1988; Moortgat 1987c, 1988b). To carry out a thorough morphological analysis, operations will be needed of similar functionality to those developed below for handling lexical redundancy rules. Conceptually, inflectional and derivational operations would apply before the types of lexical rules that we consider here. We simply assume a fully generated base lexicon consisting of the results of the inflectional and derivational systems. Our lexical rules are meant to account for distributional regularities of a fixed morphological form rather than generating more lexical entries by applying derivational or inflectional operations. Before going into the technical details of the lexical rule system, we will present simple rules for bare plurals and for so-called subject-auxiliary inversion.

Bare Plurals

We will consider the case of bare plurals first, where the basic pattern to be accounted for is as follows:

- (115) a. (Some) Penguins ate herring.
 b. (Some) Tall penguins ran quickly.
 c. (Some) Penguins in the band played yesterday.

The parenthetical determiners are optional for plural nouns. The sentences in (115) are all perfectly grammatical without the determiners. We will account for this fact with the simplest of our lexical rules:

$$(116) \ n(plu)\$ \implies np(3, plu, C)\$$$

$$\lambda\phi.\lambda x_1.\dots\lambda x_n.\mathbf{indef}(\phi(x_1)\dots(x_n))$$

There are a number of components of this rule which require individual explanation. The syntactic operation performed by a lexical rule is expressed by a category rewriting rule with the possible occurrence of the string variable $\$$. We use the $\$$ in the same way as Ades and Steedman (1982) to range over strings of alternating slashes and categories beginning with a slash and ending with a category, such as $/np$, $/np/np$, and $\backslash np \backslash (s \backslash np) / np$. The intuition here is that any lexical entry whose syntactic category matches the pattern $n(plu)\$$ will be an acceptable input to the bare pluralization rule. A category that matches this pattern will have a final applicative result of $n(plu)$ with an arbitrary string of complements coming in either direction. The $\$$ will always be used in this way to allow rules to operate over applicative results and simply pass the complementation of the input category to the output category. The same information could be expressed recursively rather than via string pattern matching, but the recursive presentation is quite messy.¹⁸ It should be

¹⁸One possibility is to build bindings for the $\$$ variable recursively, generating an infinite set of lexical rule instances without string variables, with the ground case being where the $\$$ is instantiated to the empty string.

noted that the unification categorial grammar system of Zeevat *et al.* (1987) treats complementation with a data structure consisting of a string of categories marked for directionality; in a system such as theirs, it is quite natural to express lexical rules such as the ones presented here by employing an operation of string unification, although this is not something that is actually done in their system.¹⁹ The syntactic output of the bare pluralization rule will then consist of the output pattern with the same instantiation of the \$ variable. Examples of the syntactic effects of this rule are given in the following table, where the input category, output category and string of slashes and categories that match the \$ string variable are given:

(117) Word	INPUT	OUTPUT	\$
<i>kids</i>	$n(plu)$	$np(3, plu, C)$	ϵ
<i>tall</i>	$n(plu) / n(plu)$	$np(3, plu, C) / n(plu)$	$/ n(plu)$
<i>in</i>	$n(plu) \setminus n(plu)$ $/ np(P2, N2, obj)$	$np(3, plu, C) \setminus n(plu)$ $/ np(P2, N2, obj)$	$\setminus n(plu) / np(P2, N2, obj)$

We have used the symbol ϵ to stand for the null string. The overall syntactic effect is to take a lexical entry whose syntactic category matches the input to the rule and produce a lexical categorization whose syntax is given by the output of the rule. Ignoring the semantics for the time being, the bare pluralization rule will take the lexical entries:

(118)	$n(plu)$	→	<i>penguins</i>
	$n(plu) / n(plu)$	→	<i>tall</i>
	$n(plu) \setminus n(plu) / np(P2, N2, obj)$	→	<i>in</i>

and produce the following lexical entries:

(119)	$np(3, plu, C)$	→	<i>penguins</i>
	$np(3, plu, C) / n(plu)$	→	<i>tall</i>
	$np(3, plu, C) \setminus n(plu) / np(P2, N2, obj)$	→	<i>in</i>

The new noun phrase lexical entry for the plural *penguins* has the obvious utility of allowing us to parse sentences such as *penguins waddle*. For the functor categories, consider the following parse trees:

$$(120) \quad \frac{\frac{tall}{np(3, plu, C) / n(plu)} \quad \frac{kids}{n(plu)}}{np(3, plu, C)}$$

$$(121) \quad \frac{\frac{kids}{n(plu)} \quad \frac{\frac{in}{np(3, plu, C) \setminus n(plu) / np(3, sing, obj)} \quad \frac{Pittsburgh}{np(3, sing, obj)}}{np(3, plu, C) \setminus n(plu)}}{np(3, plu, C)}$$

The significant thing to note here is that the lexical rules do not operate on the traditional *head* of a phrase, but on the category that produces the final applicative result. Thus, since the root of

¹⁹See Siekmann (1989) for a discussion of universal unification, which concerns building equational theories into the unification algorithm. The unification of strings built up by concatenation is captured by an equational theory with an associative binary operator representing concatenation and an identity element representing the null string.

the tree in (120) is determined by the applicative result of the adjective, the lexical rule must be applied to the adjective. Similarly, in (121), the rule must be applied to the preposition, as it will provide the final applicative result. The lexical rules, by means of the \$ variable, can access the eventual applicative result of a category and modify it. Thus, by applying the lexical rule to every category which can eventually produce a $n(plu)$ result, we are guaranteed that any tree which is rooted at an $n(plu)$ has a corresponding tree rooted at $np(3, plu, C)$, since the lexical rule could have been applied to the functor that produced the $n(plu)$ result. Thus, we have really achieved the result of adding a unary syntactic rule of the form:

$$(122) \ n(plu) \rightarrow \ np(3, plu, C).$$

With this rule operating in the syntax, we would get parse trees such as:

$$(123) \ \frac{\frac{\textit{tall}}{n(plu) / n(plu)} \quad \frac{\textit{kids}}{n(plu)}}{n(plu)}}{np(3, plu, C)}$$

In the case of bare pluralization, exactly the same set of strings will be accepted with the lexicon after applying the bare pluralization rule as would be accepted by adding a syntactic unary rule such as (122) to account for bare plurals.²⁰

We turn now to the semantic effects of lexical rules. The semantic component of a lexical rule is expressed as a polyadic λ -term which when applied to the semantics of the input category produces the semantics of the output category. The first argument, ϕ , will be the semantics of the input. Complications arise from the fact that lexical rules operate over varying syntactic types, so a single semantic operation will not suffice. For the bare pluralization rule in (116), and in subsequent rules, the semantic function will contain a string of abstracted variables x_1, \dots, x_n whose types will not be known until the types of the category matching the \$ symbol are known. It is assumed that if we take the \$ variable to match a string of categories and slashes $|_n \alpha_n |_{n-1} \alpha_{n-1} \cdots |_1 \alpha_1$ (where $|_i$ is either a forward or backward leaning slash), then the type of the variable x_i is taken to be the type of α_i (the orders are reversed because of the conventions for arguments being reversed in λ -abstractions and categorial grammar complements). Thus, in the case of the bare pluralization rule, the semantic effect is given as follows:

$$(124) \ \begin{array}{ll} \text{CATEGORY} & \text{SEMANTIC RULE} \\ n & \lambda\phi^n.\mathbf{indef}(\phi) \\ n / n & \lambda\phi^{\langle n, n \rangle}.\lambda x_1^n.\mathbf{indef}(\phi(x_1)) \\ n \setminus n / np & \lambda\phi^{\langle np, \langle n, n \rangle \rangle}.\lambda x_1^{np}.\lambda x_2^n.\mathbf{indef}(\phi(x_1)(x_2)) \end{array}$$

It is also worth noticing that the semantic constant **indef** must be of the type $\langle n, np \rangle$ to insure that the resulting term is well defined. Any constants introduced by lexical rules will be of a single fixed

²⁰In general, lexical rules will not be equivalent to the corresponding unary rules. For instance, adnominal predication, which applies to lexical categories of the form n/n to account for verb phrases such as *is tall* will not apply to expressions such as *tall fat* which can be syntactically analyzed as category n/n by means of extended composition rules necessary for unbounded dependency and coordination constructions. Similarly, there is no way to express a unary rule for a lexical operation like passivization without resorting to some sort of wrapping operation or \$ string variable convention (see Bach 1980, 1984 and Dowty 1978), since the most oblique argument and subject must be simultaneously affected.

type. The polymorphic behavior of the semantic operation comes from not knowing in advance how many complements are going to be taken to match the \$. In all of our lexical rules, these unknown arguments are simply fed back into the semantics of the input so that the semantics of the lexical rule can have a uniform effect. In the case of our bare plural rule, we will get the following lexical entries as a result (again suppressing the features on types for readability):

$$\begin{aligned}
 (125) \quad & np(3, plu, C) : \mathbf{indef}(\mathbf{kids}) && \rightarrow \quad kids \\
 & np(3, plu, C) / n(plu) : \lambda x_1^n. \mathbf{indef}(\mathbf{tall}(x_1)) && \rightarrow \quad tall \\
 & np(3, plu, C) \setminus n(plu) / np(P2, N2, obj) : \lambda x_1^{np}. \lambda x_2^n. \mathbf{indef}(\mathbf{in}(x_1)(x_2)) && \rightarrow \quad in
 \end{aligned}$$

The bare pluralization rule will also apply to other categories with plural nominal results, such as intensifiers and complementized nouns.

Yes/No Questions

Before going on to handle the general case of predicatives, we will consider another simple lexical rule which can be used to account for yes/no questions such as the following:

- (126) a. Did Opus eat?
 b. Is Opus eating?
 c. Has Opus eaten?

Assuming that we include an additional verb form *ynq* for yes/no questions, we can use the following lexical rule to capture the use of auxiliaries to form questions:

$$\begin{aligned}
 (127) \quad & s(fin) \setminus np(P, N, C) && \Longrightarrow && s(ynq) \\
 & / (s(V) \setminus np(P, N, C)) && && / (s(V) \setminus np(P, N, C)) \\
 & && && / np(P, N, C) \\
 & \lambda \phi. \lambda x^{np}. \lambda v^{(np,s)}. \mathbf{ques}(v(x)) \\
 & \text{where } V \neq fin \text{ and } V \neq inf
 \end{aligned}$$

This rule will apply to an auxiliary category and output a category which takes a noun phrase and verb phrase argument to produce a yes/no question as a result. This rule will result in the following basic and derived lexical entries:

$$(128) \quad s(fin) \setminus np(2, plu, subj) / (s(bse) \setminus np(2, plu, subj)) \rightarrow do$$

do

$$\begin{aligned}
 & s(ynq) / (s(bse) \setminus np(2, plu, subj)) / np(2, plu, subj) && \rightarrow \quad do \\
 & \lambda x^{np}. \lambda v^{(np,s)}. \mathbf{ques}(\mathbf{do}(v)(x))
 \end{aligned}$$

$$(129) \quad s(fin) \setminus np(1, sing, subj) / (s(pred) \setminus np(3, sing, subj)) \rightarrow am$$

am

$$\begin{aligned}
 & s(ynq) / (s(pred) \setminus np(1, sing, subj)) / np(1, sing, subj) && \rightarrow \quad am \\
 & \lambda x^{np}. \lambda v^{(np,s)}. \mathbf{ques}(\mathbf{am}(v)(x))
 \end{aligned}$$

(130) $s(fin) \setminus np(3, sing, subj) / (s(perf) \setminus np(3, sing, subj)) \rightarrow has$
has

$s(ynq) / (s(perf) \setminus np(3, sing, subj)) / np(3, sing, subj) \rightarrow has$
 $\lambda x^{np}. \lambda v^{(np,s)}. \mathbf{ques}(\mathbf{has}(v)(x))$

These new lexical entries can be used in analyses such as the following:

(131)

am	I	$running$
$s(ynq)$	$np(1, sing, subj)$	$s(pred)$
$/ (s(pred) \setminus np(1, sing, subj))$	i	$\setminus np(1, sing, subj)$
$/ np(1, sing, subj)$		running
$\lambda x^{np}. \lambda v^{(np,s)}. \mathbf{ques}(\mathbf{am}(v)(x))$		
$s(ynq)$		
$/ (s(pred) \setminus np(1, sing, subj))$		
$\lambda v^{(np,s)}. \mathbf{ques}(\mathbf{am}(v)(i))$		
$s(ynq) : \mathbf{ques}(\mathbf{am}(\mathbf{running})(i))$		

5 Predicatives

In this section, we will present a detailed analysis of the distribution of predicatives in English and in so doing, demonstrate the utility of schematic lexical rules with complement string variables.

5.1 Passivization

While passivization might be more fairly characterized as a morphological operation, we will discuss its effects on syntactic and semantic categorizations and ignore its phonological and orthographic effects. The passive rule provides an excellent example of the full power of schematic lexical rules. The passive operation applies to a verbal category that produces a sentential result and takes at least one object-marked nominal complement to its right. The syntactic result is a new category whose most oblique *obj* marked noun phrase complement is replaced with an optional *by* marked noun phrase. Semantically, the subject and most oblique complement switch thematic roles. For instance, consider who is doing what to whom in the following active and passive constructions:

- (132) a. Opus loved bill.
 b. Bill was loved by Opus.
 c. Bill was loved.
 d. Bill was loved by something.

In both (132)a and (132)b, Opus is the ‘lover’ and Bill the ‘lovee’. In the case of (132)c Bill still plays the role of ‘lovee’, but there is now an existential quantification of some variety filling in the missing role so that (132)c is roughly equivalent in meaning to (132)d.²¹

Our primary consideration in treating the passive in this section is that it produces a predicative verb phrase result (note the necessary copula in the passives in (132)). Our claim is that in any

²¹In the case of quantified subjects as in the sentence *Every cat was loved*, the existentially quantified position for the ‘lover’ would have to take narrow scope to account for the fact that a different individual may love each cat.

context where a predicative verb phrase such as *hitting Opus* is licensed, it is also possible to have a passive verb phrase such as *hit by Opus* (of course, the thematic roles are reversed in the passive version). Consider the following examples:

- (133) a. Binkley saw Bill hitting Opus.
 b. Binkley saw Opus hit by Bill.
 c. Opus danced when watching Bill.
 d. Opus danced when watched by Bill.
 e. Opus ate herring with Bill offending him.
 f. Opus ate herring with Bill offended by him.
 g. Was Opus hit by Bill?

It should be kept in mind while reading this section that anything that gets classified as a predicative verb phrase will be able to occur in any location in which a lexical predicative verb phrase may be found. This is due to the fact that there is nothing in a categorial analysis that is sensitive to derivational histories; every analysis involving a complex expression is determined solely on the basis of the category assigned to the root of its parse tree. In government-binding theory, on the other hand, the number of traces and indices within a phrase-marker is quite significant as it interacts with modules such as the binding, subadjacency and empty category principles.

In our official notation, the passive rule is as follows:

$$(134) \quad s(bse) \setminus np(P1, N1, subj) \$ \implies s(pred) \setminus np(P2, N2, subj) \$ \\
 / np(P2, N2, obj) \qquad \qquad \qquad / np(P1, N1, by) \\
 \lambda\phi.\lambda x_1.\dots\lambda x_n.\lambda y_2.\lambda y_1.\phi(y_1)(x_1)\dots(x_n)(y_2)$$

Note that the *by* marked noun phrase occurs inside of the \$ in the result, thus forcing it to occur after the other complements in the verb phrase.

As we have allowed ourselves no direct representation of optional complementation, we will need the following additional rule to deal with the situation where the *by*-phrase is omitted:

$$(135) \quad s(bse) \setminus np(P1, N1, subj) \$ / np(P2, N2, obj) \implies s(pred) \setminus np(P2, N2, subj) \$ \\
 \lambda\phi.\lambda x_1.\dots\lambda x_n.\lambda y_1.\phi(y_1)(x_1)\dots(x_n)(\mathbf{something})$$

We will consider how these rules apply to the different categorizations that we have given to transitive and poly-transitive verbs in the rest of this section.

Simple Transitive Passives

Consider the distribution of the passives in the following simple transitive and bitransitive verbs:

- (136) a. Opus hit Binkley
 b. Binkley was hit (by Opus).
- (137) a. Bill gave Opus herring.
 b. Opus was given the herring (by Bill).
 c. The herring was given (by Bill).

To derive this example by application of the passive rule, it is first necessary to be able to derive the detransitivized active version: *Bill gave the herring*.

- (138) a. Bill gave herring to Opus.
 b. Herring was given to Opus (by Bill).

In the case of (136), we have the following basic lexical entries and entries derived by applying the passivization rules:²²

- (139) $s(bse) \setminus np(subj) / np(obj) : hit \rightarrow hit$
 $s(pred) \setminus np(subj) / np(by) : \lambda x. \lambda y. \mathbf{hit}(y)(x) \rightarrow hit$
 $s(pred) \setminus np(subj) : \lambda y. \mathbf{hit}(y)(\mathbf{something}) \rightarrow hit$

This will give us the following analysis for the passive in (136)b:

$$\begin{array}{c}
 (140) \quad \frac{\frac{\frac{\frac{Binkley}{np} \quad \mathbf{binkley}}{s(fin) \setminus np} \quad \mathbf{was}}{s(pred) \setminus np} \quad \mathbf{was}}{\frac{\frac{\frac{hit}{s(pred) \setminus np / np(by)} \quad \lambda x. \lambda y. \mathbf{hit}(y)(x)}{s(pred) \setminus np} \quad \lambda y. (\mathbf{hit}(y)(\mathbf{opus}))}}{\frac{by \quad Opus}{np(by)} \quad \mathbf{opus}} \quad \mathbf{opus}}{s(fin) \setminus np} \quad \mathbf{was}(\lambda y. (\mathbf{hit}(y)(\mathbf{opus})))}}{s(fin)} \quad \mathbf{was}(\lambda y. (\mathbf{hit}(y)(\mathbf{opus})))(\mathbf{binkley})}
 \end{array}$$

Now suppose that we fix a simple control-based semantics for **was**, such as:

(141) $\mathbf{was} =_{\text{def}} \lambda P^{(np,s)}. \lambda x^{np}. \mathbf{past}(P(x))$

In the case of the passive analysis, we would then have:

(142) $\mathbf{was}(\lambda y. \mathbf{hit}(y)(\mathbf{opus}))(\mathbf{binkley}) = \mathbf{past}(\mathbf{hit}(\mathbf{binkley})(\mathbf{opus}))$

This semantic treatment of passivization is for illustrative purposes only and should not be taken as a serious proposal. But it does illustrate how the thematic roles are preserved from the usual derivation of the active version in (136)a:

$$\begin{array}{c}
 (143) \quad \frac{\frac{\frac{Opus}{np} \quad \mathbf{opus}}{s \setminus np / np} \quad \mathbf{hit}}{s \setminus np} \quad \mathbf{hit}(\mathbf{binkley})}{s(fin)} \quad \mathbf{hit}(\mathbf{binkley})(\mathbf{opus})}
 \end{array}$$

²²We will not consider how the phonological or orthographic representation is affected in the move from the base form to the passive form, such as in the derivation of the passive forms *eaten*, *hit* and *loved* from the stem forms *eat*, *hit* and *love*.

In the situation where the *by*-phrase is omitted, the second lexical rule for passivization in (135) will apply, effectively filling the semantic position that would have been contributed by the subject with **something**. For instance, this would give us the following semantics for (136)b without the *by Opus* complement:

- (144) a. Binkley was hit.
 b. **past((hit)(binkley)(something))**

Next consider what happens to the semantics of a bitransitive verb such as *give* when it undergoes passivization:

- (145) $s \setminus np(subj) / np(obj) / np(obj) : \mathbf{give} \rightarrow give$
 $s \setminus np(subj) / np(by) / np(obj) : \lambda x. \lambda y. \lambda z. \mathbf{give}(z)(x)(y) \rightarrow given$

This will result in the following semantic analyses:

- (146) a. Bill gave Opus the herring.
 b. **give(opus)(the(herring))(bill)**
- (147) a. Opus was given the herring by Bill.
 b. **past(give(opus)(the(herring))(bill))**

In the case of the *to* marked bitransitive we would have the basic and derived lexical entries:

- (148) $s \setminus np(subj) / np(to) / np(obj) : \mathbf{giveto} \rightarrow give$
 $s \setminus np(subj) / np(by) / np(to) : \lambda x. \lambda y. \lambda z. \mathbf{giveto}(z)(x)(y) \rightarrow given$

The result is the following semantic assignments:

- (149) a. Bill gave the herring to Opus.
 b. **giveto(the(herring))(opus)(bill)**
- (150) a. The herring was given to Opus by Bill.
 b. **past(giveto(the(herring))(opus)(bill))**

Ordinarily, a lexical rule of some sort would be employed to produce a bitransitive entry for (145) from the *to* marked version for (148) (see Bresnan 1982c, for example). This operation is usually referred to as *dative shift*, to indicate that the preposition *to* marks what is classically the dative case. This is the type of rule which we will not consider, because it is not fully productive (that is, not all dative or *to* marked arguments can be shifted to an indirect object position). But the effects of such a shift could be described as follows in the cases where *to* marked arguments become indirect objects:

- (151) $s \setminus np(subj) / np(to) / np(obj) \implies s \setminus np(subj) / np(obj) / np(obj)$
 $\lambda \phi. \lambda x. \lambda y. \lambda z. \phi(y)(x)(z)$

If we assumed that the *to* marked version resulted from this rule, we would have:

- (152) **giveto** = $\lambda x. \lambda y. \lambda z. \mathbf{give}(y)(x)(z)$

Consequently, (149)a and (146)a would be assigned the same semantics. There are a number of subtle thematic effects that must be accounted for at the level of the base lexicon which we will not consider (but see Dowty (1979)).

The following transitive sentential complement verbs will undergo passivization in exactly the same way as bitransitive verbs such as *give*:

- (153) a. I convinced Bill that Opus ate.
 b. Bill was convinced that Opus ate (by me).

The syntactic category $s \setminus np / s$ assigned to *know* is not of the correct category to undergo passivization, so that we get the correct prediction:

- (154) a. I know Opus ate.
 b. * Opus was known ate (by me).

Passive and Control

Consider the following cases of passivization with control verbs:

- (155) a. Opus persuaded Bill to eat the herring.
 b. Bill was persuaded to eat the herring.
 (156) a. Opus promised Bill to eat the herring.
 b. * Bill was promised to eat the herring.
 (157) a. Bill saw Opus eating.
 b. Opus was seen eating.

In all of the situations where the passive is acceptable, the following syntactic conversion is carried out:

$$(158) s \setminus np(subj) / (s \setminus np) / np(obj) \implies s \setminus np(subj) / np(by) / (s \setminus np)$$

Just as in the previous cases, the semantics will be unaffected. What remains to be explained is *Visser's generalization* that only object-control verbs may undergo passivization. The distinction is explained in terms of the completeness and coherence conditions which require specified thematic roles to be filled (Bresnan 1982c). We must rely on general semantic selectional restrictions, which will be necessary in any case to handle morphological operations, which are known to be particularly selective about the lexical semantics of their inputs (see Dowty 1979, chapter 6, for similar arguments concerning such lexical operations as the causative).

An interesting case arises for raising verbs such as *believe*, since passivization will have to account for the possibility of expletive objects moving into subject position. Consider the pair:

- (159) a. Milo believed it to be raining.
 b. It was believed to be raining (by Milo).

Our lexical rule will need to be generalized to apply to oblique complements marked with *it* and carry this marking into their new surface role as subjects. For instance, we would warrant the following syntactic operation on the expletive version of *believe*:

$$(160) s \setminus np(subj) / (s \setminus np(it)) / np(it) \implies s \setminus np(it) / np(by) / (s \setminus np(it))$$

Sentential Subjects and Passivization

Not only can expletives be made into subjects by passivization, but complementized sentences can also be promoted to subjecthood, as seen in:

- (161) a. Bill knew that Opus ate.
b. That opus ate was known (by Bill).
c. Bill wondered whether Opus ate.
d. * Whether Opus ate was wondered (by Bill).

These examples will not be handled by the current passivization rule. One solution to this would be to abandon our treatment of complementizers as marking verb forms, and instead consider them to be a special kind of case marked determiner of the category $np(3, sing, that) / s(fin)$. This is very similar to the proposal of Sag *et al.* (1985), based on Weisler (1982), which admits $NP[NFORM : S]$ as a categorization of complementized sentences. We would then need to assume that complementized sentential complement verbs like *knew* were syntactically categorized as $s \setminus np(subj) / np(that)$. In this case, we could treat *that* in the same way as the expletive marked *it* subjects in (160) using the generalized passive rule to deal with normal, expletive and sentential subjects. The details of this proposal remain to be worked out, and consideration should also be given to infinitival sentential subjects such as in:

- (162) ((For Opus) to eat herring) is annoying.

In this case, there is an additional complication stemming from the optionality of the *for* noun phrase, which plays the role of the subject of the infinitival subject.

Overgeneration, Selection and Phrasal Passives

The way in which our lexical rules are defined, passivization will also apply to verbal adjuncts that take object nominal arguments. For instance, consider the category assigned to prepositions and the resulting output of applying the passivization rule to it:

- (163) $s(bse) \setminus np(P, N, C) \setminus (s(bse) \setminus np(P, N, C)) / np(P2, N2, obj)$
 $\implies s(pred) \setminus np(P2, N2, subj) \setminus (s(pred) \setminus np(P2, N2, subj)) / np(P, N, by)$

This would allow us to generate both the acceptable and unacceptable examples in the following and will assign the same thematic roles in both the active and passive examples:

- (164) a. Opus walked under the bridge.
b. The bridge was walked under by Opus.
c. Opus hit Bill under the bridge.
d. * The bridge was hit opus under by Bill.
e. Opus ate with Bill watching.
f. * Bill ate with watching by Opus.

Obviously, some restrictions must apply to the application of passivization. The simplest thing to do would be to mark the sentential applicative result of actual verbs with some binary feature that marked whether or not it was a verb. This is the usual strategy employed in unification grammar theories such as GPSG or HPSG to account for this kind of selectional restriction.

To account for the grammatical instance of passivization in (164)b, it is necessary to treat *walk under* as a unit which is categorized as a transitive verb (Bresnan 1982c).

5.2 Adnominal Predicatives

One of the major types of predicative is the adnominal, in either its adjectival and post-nominal forms. Consider the following examples of the predicative functions of adnominals:

- (165) a. The herring was [red] yesterday.
 b. The herring was probably [extremely red].
 c. The herring was believed to be [with Opus].
 d. A herring is usually eaten when [red].
 e. Was the herring [with Opus] yesterday?
 f. Is the herring not [very tasty] in Pittsburgh?

In all of these examples, a nominal modifier is used in the same manner as a predicative verb phrase. For instance, the nominal modifiers are modified by pre and post-verbal adverbials, and may show up in yes/no questions. Our account of this distribution is different from other accounts such as those found in GPSG (Sag *et al.* 1985) in that we assume a lexical rule that provides an entry with the applicative result $s(pred) \setminus np$ when applied to adnominals. To achieve this, we assume the following two lexical rules for the pre-nominal and post-nominal modifiers:

$$(166) \quad n(N) / n(N) \ \$ \implies s(pred) \setminus np(P, N, subj) \ \$ \\ \lambda\phi.\lambda x_1 \cdots x_n.\lambda y.\mathbf{pred}(\phi(x_1) \cdots (x_n))(y)$$

$$(167) \quad n(N) \setminus n(N) \ \$ \implies s(pred) \setminus np(P, N, subj) \ \$ \\ \lambda\phi.\lambda x_1 \cdots x_n.\lambda y.\mathbf{pred}(\phi(x_1) \cdots (x_n))(y)$$

In these lexical rules, **pred** must be of the type $\langle\langle n(N), n(N) \rangle, \langle np(P, N, subj), s(pred) \rangle\rangle$, thus taking an adnominal and noun phrase argument to produce a predicative sentential category. We will not be concerned about the denotation of **pred**, but we have supplied it with all of the arguments that it could possibly require.

Consider the first of these rules, which applies to pre-nominal adnominals, with the following results:

$$(168) \quad n(N) / n(N) : \mathbf{red} \quad \rightarrow \quad red \\ s(pred) \setminus np(P, N, subj) : \mathbf{pred}(\mathbf{red}) \quad \rightarrow \quad red$$

$$(169) \quad n(N) / n(N) / (n(N) / n(N)) : \mathbf{very} \quad \rightarrow \quad very \\ s(pred) \setminus np(P, N, subj) / (n(N) / n(N)) : \lambda x^{(n,n)}.\mathbf{pred}(\mathbf{very}(x)) \quad \rightarrow \quad very$$

The second rule will apply to prepositions as follows:

$$(170) \quad n(N) \setminus n(N) / np(P2, N2, obj) : \mathbf{with} \quad \rightarrow \quad with \\ s(pred) \setminus np(P, N, subj) / np(P2, N2, obj) : \lambda x^{np}.\lambda y^{np}.\mathbf{pred}(\mathbf{with}(x))(y) \quad \rightarrow \quad with$$

A similar category will result when the lexical rule is applied to prepositions which do not take complements, such as *inside*. These new categories will allow the following syntactic derivations:

$$(171) \quad \frac{\frac{with}{s(pred) \setminus np(P, N, subj) / np(3, sing, obj)} \quad \frac{Opus}{np(3, sing, obj)}}{s(pred) \setminus np(P, N, subj)}$$

$$(172) \frac{\frac{\textit{probably}}{s(\textit{pred}) \setminus np(P, N, \textit{subj})} / (s(\textit{pred}) \setminus np(P, N, \textit{subj}))}{\frac{\frac{\textit{very}}{s(\textit{pred}) \setminus np(P, N, \textit{subj})} / (n(N) / n(N))}{\frac{\textit{tall}}{n(N) / n(N)}}} / (n(N) / n(N))}{s(\textit{pred}) \setminus np(P, N, \textit{subj})}$$

Again, it is important to note that this lexical rule will allow adnominals (those with result category $n \setminus n$ or n / n) to occur in any location that a predicative verb phrase might occur.

While we have not dealt with relative clauses, since their proper treatment depends on an analysis of unbounded dependency constructions, we present a simplified account of subject relative clauses for the sake of illustrating the way in which agreement is handled by lexical rules. The following pair consists of the basic entry for the subject relative *who* along with the result of applying the adnominal predication lexical rule:²³

$$(173) \begin{array}{ll} n(N) \setminus n(N) / (s(\textit{fin}) \setminus np(P, N, \textit{subj})) & \rightarrow \textit{who} \\ s(\textit{pred}) \setminus np(P2, N, \textit{subj}) / (s(\textit{fin}) \setminus np(P, N, \textit{subj})) & \rightarrow \textit{who} \end{array}$$

These entries will account for the following contrast:

- (174) a. The penguin who sings
 b. * The penguin who sing
 c. Opus is who sings.
 d. * Opus is who sing.

The reason that the second example can not be analyzed is that the number of the verb phrase argument to the relative clause will be the same as the number of the resulting adnominal, which in turn, will be the number of the predicative result. This can be seen in:

$$(175) \frac{\frac{\textit{who}}{n(\textit{sing}) \setminus n(\textit{sing})} / (s(\textit{fin}) \setminus np(P2, \textit{sing}, \textit{subj}))}{\frac{\textit{sings}}{s(\textit{fin}) \setminus np(P2, \textit{sing}, \textit{subj})}}}{n(\textit{sing}) \setminus n(\textit{sing})}$$

$$(176) \frac{\frac{\textit{who}}{s(\textit{pred}) \setminus np(P, \textit{sing}, \textit{subj})} / (s(\textit{fin}) \setminus np(P2, \textit{sing}, \textit{subj}))}{\frac{\textit{sings}}{s(\textit{fin}) \setminus np(P2, \textit{sing}, \textit{subj})}}}{s(\textit{pred}) \setminus np(P, \textit{sing}, \textit{subj})}$$

The auxiliary *be* will then pass along the number agreement information from the predicative *who sings*, requiring the subject of the main clause to be singular.

²³Object relative pronouns could be assigned to the category $n \setminus n / (s / np(\textit{obj}))$, with sentences lacking object noun phrases such as *Bill hit* and *Opus believed he hit* analyzed as being of the category s / np using extended categorial operations (see Morrill (1987b, 1988) and Steedman (1987, 1988)). Of course, it would also be possible to treat slashes denoting gaps or traces as occurring in a different feature than those for normal complements (see Pollard and Sag (forthcoming) or Zeevat, Klein and Calder (1987)).

5.3 Nominal Predicatives

Full noun phrases can also be used as predicatives, as long as they do not have quantificational force. This can be seen in the examples:

- (177) a. Opus is [a hero].
 b. * The penguins is a hero.
 c. The penguins were known to be [the real heros].
 d. Opus celebrated while still [a hero] in Bloom County.
 e. Was Opus really [a penguin]?

Note that there has to be agreement between the number of the predicative noun phrase and the number of the subject, as evidenced by (177)a and (177)b.²⁴ We can account for the distribution of nominal predicatives with the following lexical rule:

$$(178) \quad np(P, N, obj) \$ \implies s(pred) \setminus np(P2, N, subj) \$ \\ \lambda\phi.\lambda x_1 \cdots x_n.\lambda y^{np}.\mathbf{npred}(\phi(x_1) \cdots (x_n))(y)$$

Again, we will not be concerned with the actual content of the **npred** constant, other than the fact that it takes two noun phrase arguments and produces a predicative sentential result. The requirement that the noun phrase undergoing the lexical rule be in object case is so that we capture the contrast in:

- (179) a. Opus is him.
 b. * Opus is he.

Some examples of the application of this rule are as follows:

$$(180) \quad np(3, sing, obj) : \mathbf{opus} \quad \rightarrow \quad Opus \\ s(pred) \setminus np(P, sing, subj) : \mathbf{npred(opus)} \quad \rightarrow \quad Opus$$

$$(181) \quad np(3, N, obj) / n(N) : \mathbf{the} \quad \rightarrow \quad the \\ s(pred) \setminus np(P, N, subj) / n(N) : \lambda x^n.\mathbf{npred(the)}(x) \quad \rightarrow \quad the$$

This rule will provide our first example of nested lexical rule application. By applying the bare pluralization rule, it was possible to convert any category that produced a result of category $n(plu)$ into one with the same complements that produced an applicative result of the category $np(3, plu, C)$. All of these plural nominals, such as *penguins*, *tall* and *with* will also serve as input to the predication rule. For instance, we have:

$$(182) \quad np(plu) \setminus n(plu) / np(P, N, obj) \quad \rightarrow \quad with \\ s(pred) \setminus np(P2, plu, subj) \setminus n(plu) / np(P, N, obj) \quad \rightarrow \quad with$$

²⁴Complications arise in the case of conjoined noun phrases, as in *Opus and Bill are animals*, where the two proper names are singular, but their coordination is plural. To see that the coordination is plural, note that *Opus and Bill run* is grammatical, while ** Opus and Bill runs* is not. Also note that disjunctions remain singular, so that *Opus or Bill runs* is grammatical, while ** Opus or Bill run* is not.

This entry will allow the following analysis:

$$(183) \frac{\frac{\textit{penguins}}{n(\textit{plu})} \quad \frac{\textit{with}}{s(\textit{pred}) \setminus np(\textit{P}, \textit{plu}, \textit{subj}) \setminus n(\textit{plu})} \quad \frac{\textit{Opus}}{np(3, \textit{sing}, \textit{obj})}}{\frac{\textit{with}}{s(\textit{pred}) \setminus np(\textit{P}, \textit{plu}, \textit{subj}) \setminus n(\textit{plu})} / np(3, \textit{sing}, \textit{obj})}}{\frac{s(\textit{pred}) \setminus np(\textit{P}, \textit{plu}, \textit{subj}) \setminus n(\textit{plu})}{s(\textit{pred}) \setminus np(\textit{P}, \textit{plu}, \textit{subj})}}$$

5.4 Sentential Coordination and Predicatives

The standard coordination scheme used in phrase structure grammars to account for sentential coordination allows two identical verbal categories to be conjoined to form a result of the same category. This is usually captured by means of a phrase structure scheme of the form:

$$(184) \alpha : \phi(\mathbf{f}_1)(\mathbf{f}_2) \rightarrow \alpha : \mathbf{f}_1 \quad co : \phi \quad \alpha : \mathbf{f}_2$$

where α is taken to be an arbitrary category that produces a sentential applicative result of any verb form and where co is the syntactic category assigned to coordinators such as *and* and *or*.²⁵ As it stands, this coordination scheme is not sufficient to deal with noun phrase coordinations, which raise a number of syntactic and semantic difficulties (see Hoeksema 1987, Carpenter 1989).

For instance, we want to be able to produce analyses such as:

$$(185) \frac{\frac{\textit{Opus ate}}{s(\textit{fin}) : \mathbf{ate}(\mathbf{opus})} \quad \frac{\textit{and}}{co : \mathbf{and}} \quad \frac{\textit{Opus drank}}{s(\textit{fin}) : \mathbf{ate}(\mathbf{opus})}}{\frac{\textit{and}}{s(\textit{fin}) : \mathbf{and}(\mathbf{ate}(\mathbf{opus}))(\mathbf{drank}(\mathbf{opus}))}}$$

As usual, we will not be concerned with the value of **and**, but the situation is slightly different here in that **and** must be polymorphic and apply to an arbitrary pair of verbal categories (see Gazdar 1980). A verbal category is defined to be any category with an applicative result of s . For the sake of illustration, we make the following semantic assumption:

$$(186) \phi(\mathbf{f})(\mathbf{g}) = \lambda x_1 \cdots x_n. \phi(\mathbf{f}(x_1) \cdots (x_n))(\mathbf{g}(x_1) \cdots (x_n))$$

where ϕ is the semantics of the coordinator and where $\mathbf{f}(x_1) \cdots (x_n)$ is of type s .²⁶ Thus, we would have the following semantic assignment:

$$(187) \frac{\frac{\textit{loves}}{s \setminus np / np : \mathbf{love}} \quad \frac{\textit{and}}{co : \mathbf{and}} \quad \frac{\textit{hates}}{s \setminus np / np : \mathbf{hate}}}{s \setminus np / np : \lambda x_1. \lambda x_2. \mathbf{and}(\mathbf{love}(x_1)(x_2))(\mathbf{hate}(x_1)(x_2))}$$

The problem that is usually encountered with the coordination of predicatives is that they are not assigned to the same categories, so that they can not be coordinated according to this coordination scheme. This has led those working within unification grammar formalisms to extend the operations

²⁵This scheme is usually extended to deal with constructions consisting of multiple coordinators such as *either-or* constructions, as well as constructions consisting of more than two conjuncts (see Gazdar 1981b). Binary coordination will serve to illustrate our major points.

²⁶A non-distributive semantics will actually be necessary to avoid the usual problems surrounding the interaction between quantification and coordination in sentences such as *every penguin waddles or swims*, which does not mean that every penguin waddles or that every penguin swims (see Carpenter 1989).

and allow an operation of generalization, since predicatives are usually assumed to share a feature $PRED : +$ (see Karttunen 1984). Having insured by means of lexical rules that the predicatives are uniformly assigned to the category $s(pred) \setminus np$, there is no difficulty encountered with the coordination of “unlike” categories. For instance, we would have the analysis:

$$(188) \frac{\frac{\textit{short}}{s(pred) \setminus np : \mathbf{pred}(\mathbf{short})} \quad \frac{\textit{and}}{co : \mathbf{and}} \quad \frac{\textit{a penguin}}{s(pred) \setminus np : \mathbf{npred}(\mathbf{a}(\mathbf{penguin}))}}{s(pred) \setminus np : \lambda x. \mathbf{and}(\mathbf{pred}(\mathbf{short})(x))(\mathbf{npred}(\mathbf{a}(\mathbf{penguin}))(x))}$$

Using this verb phrase analysis and our simple control semantics for *was* would produce the following semantic analysis:

- (189) a. Opus was [[short] and [a penguin]].
 b. $\mathbf{past}(\mathbf{and}(\mathbf{pred}(\mathbf{short})(\mathbf{opus}))(\mathbf{npred}(\mathbf{a}(\mathbf{penguin}))(\mathbf{opus})))$

In a similar fashion, we could analyze all of the sentences in:

- (190) a. Opus is [short] and [a penguin].
 b. Opus is [in the kitchen] and [eating].
 c. Opus ate herring with Binkley [sick] and [watching the whole affair].
 d. Opus is [tired], but [eager to eat].
 e. Milo is [the editor of the paper] and [afraid that Opus will leave].

Of course, extending the binary coordination scheme to an n -ary version would allow multiple predicatives to be coordinated.

5.5 Predicatives as Adjuncts

Besides occurring as complements to the copula *be*, another major function of predicatives is to act as adjuncts. In this capacity, predicatives can modify either nouns or verb phrases. We will consider these uses in turn.

Predicatives as Adnominals

When used as adnominals, predicatives show up post-nominally. This distribution can be accounted for with the following lexical rule:

$$(191) \quad s(pred) \setminus np(P, N, C) \ \$ \implies n(N) \setminus n(N) \ \$ \\ \lambda\phi. \lambda x_1 \cdots x_n. \lambda y^n. \mathbf{adn}(\phi(x_1) \cdots (x_n))(y)$$

In this case, \mathbf{adn} is of the semantic type $\langle\langle np, s \rangle, \langle n, n \rangle\rangle$ so that it takes a predicative verb phrase as input and produces an adnominal. We will not be concerned with the actual content of \mathbf{adn} .

This rule of predicative adnominalization will simply allow predicatives to occur as post-nominal modifiers. When applied to derived verbal predicatives, this will result in the grammaticality of the following noun phrases:

- (192) a. the kid [**talking**]
 b. $s(pred) \setminus np \implies n \setminus n$

- (193) a. the cat [**hitting** Opus]
 b. $s(pred) \setminus np / np \implies n \setminus n / np$
- (194) a. the kid [[**persuading** Opus] [to eat the herring]]
 b. $s(pred) \setminus np / (s(inf) \setminus np) / np \implies n \setminus n / (s(inf) \setminus np) / np$
- (195) a. the herring [**being** eaten]
 b. the herring [**not** [seen by opus]]
 c. $s(pred) \setminus np / (s(pred) \setminus np) \implies n \setminus n / (s(pred) \setminus np)$

In these examples, the verb that undergoes the lexical shift has been put in bold face, with the resulting categorial transformation listed below. Notice that in the last example, the negative *not* and auxiliary *being* are assigned to identical syntactic categories, and thus undergo exactly the same lexical rules. Also note that with respect to coordination, an adnominalized predicative such as *admiring* is assigned to the same category as a nominal preposition, so that the following sentence would be allowed:

- (196) The kid [with and admiring] the short penguin eating herring.

Predicative adnominalization will also apply to verbal modifiers, with the following effects:

- (197) a. the herring [**probably** [being eaten]]
 b. $s(pred) \setminus np / (s(pred) \setminus np) \implies n \setminus n / (s(pred) \setminus np)$
- (198) a. the herring [eaten **yesterday**]
 b. $s(pred) \setminus np \setminus (s(pred) \setminus np) \implies n \setminus n \setminus (s(pred) \setminus np)$

It is important to note that the adverbials will still be required to modify a predicative verb phrase, since that is their only categorization that can serve as input to the rule. This will rule out potential noun phrases such as:

- (199) * the herring eat yesterday

Adjunct attachment ambiguities will be preserved by the adnominalization rules, and hence the following noun phrase has two analyses, depending on which adverb is being operated on by the lexical rule:

- (200) a. the herring [**probably** [being eaten today]]
 b. the herring [[probably being eaten] **today**]

Besides the simple adverbs, predicative adnominalization will also apply to prepositional phrases and other complementized adjuncts. For instance, we will have the following:

- (201) a. the herring [swimming [**beside** Opus]]
 b. $s(pred) \setminus np \setminus (s(pred) \setminus np) / np \implies n \setminus n \setminus (s(pred) \setminus np) / np$
- (202) a. the penguin [[eating herring] [**while** swimming]]
 b. $s(pred) \setminus np \setminus (s(pred) \setminus np) / (s(pred) \setminus np) \implies n \setminus n \setminus (s(pred) \setminus np) / (s(pred) \setminus np)$

- (203) a. the penguin [swimming [**with** [the water] [nearly freezing]]]
 b. $s(pred) \setminus np \setminus (s(pred) \setminus np) / (s(pred) \setminus np) / np$
 $\implies n \setminus n \setminus (s(pred) \setminus np) / (s(pred) \setminus np) / np$

The predicative adnominalization rule produces output with a final applicative result of n , with the possibility that it will be $n(plu)$. Thus the output of any application of the adnominalization rule will serve as valid input to the bare pluralization rule. Consequently, all of the examples given above with a final applicative result of $n(plu)$ will also have a categorization with a final applicative result of $np(3, plu, C)$. Thus, we could derive all of the previous examples without determiners if plural nouns were substituted for the singular ones. For instance, all of the following can be analyzed as plural noun phrases after the bold faced predicative adnominals undergo bare pluralization:

- (204) a. cats **hitting** opus
 b. penguins **being** hit by cats
 c. penguins **probably** hitting cats
 d. penguins swimming **beside** the herring
 e. penguins eating herring **while** swimming

In conjunction with the adnominal predication rule (167), the predicative adnominalization rule (191) will lead to circularity in the lexicon. This can easily be seen from the categorial effects of the rules:

- (205) $s(pred) \setminus np(P, N, C) \$$
 $\implies n(N) \setminus n(N) \$$
 $\implies s(pred) \setminus np(P, N, subj) \$$

This leads to circular derivations such as the following in the case of nominal prepositions:

- (206) $n(sing) \setminus n(sing) / np$
 $\implies s(pred) \setminus np / np$
 $\implies n(sing) \setminus n(sing) / np$

While this does not lead to any additional strings being accepted, it will cause problems for implementations if there is not some test for redundancy. For instance, every time a new lexical entry is created, a test could be performed to determine whether or not it has already been generated. This is a typical step in any kind of straightforward closure algorithm. The real problem, though, will be semantic, if there is no way to make sure that the semantic value of the circular derivations turn out to be the same.

Another interesting thing to recognize about the interaction of the predication and adnominalization rules is that they will generate post-nominal adnominal categories for every pre-nominal adjective (but not conversely), according to the following derivation chain:

- (207) $n(N) / n(N) \$ \implies s(pred) \setminus np(P, N, subj) \$ \implies n(N) \setminus n(N) \$$

The resulting categorizations will allow us to derive the following “poetic” uses of adjectives such as:

- (208) a. the ocean **blue**
 b. the trees [**tall** and **broad**]
 c. the herring [**red** and **delicious**]

The only explanation for why these adjectives do not show up in this location more often seems to be that they already have a perfectly good home before the nominals that they modify. Presumably, there is some pragmatic rule operating which requires more simple forms to be used in the case where there are two possible locations. Such rules can, and will, be overridden where other pragmatic considerations are more significant.

Predicatives as Adverbials

It is possible for predicatives to function as adverbials in the same manner that they function as adnominals. Their standard position is post-verbal in these cases, but they will often be found in a fronted position. We will only be concerned with their standard post-verbal position and assume that the fronted versions are analyzed by some kind of topicalization rule. The lexical rule we propose is as follows:

$$(209) \quad s(pred) \setminus np(P, N, C) \ \$ \implies s(V) \setminus np(P, N, C) \setminus (s(V) \setminus np(P, N, C)) \ \$ \\ \lambda\phi.\lambda x_1 \dots x_n.\lambda y^{(np,s)}. \mathbf{predadv}(\phi(x_1) \dots (x_n))(y)$$

In this case, the constant **predadv** must be of the type $\langle\langle np, s(pred) \rangle, \langle\langle np, s(V) \rangle, \langle np, s(V) \rangle \rangle\rangle$, so that it takes a predicative verb phrase and returns a modifier of an arbitrary verb.

The application of this rule can be seen in the following examples of well formed sentences, using the same notational conventions as previously, where we have included the topicalized versions in some cases for comparison:²⁷

- (210) a. Opus ate herring **swimming**.
 b. Swimming, Opus ate herring.
 c. $s(pred) \setminus np \implies s(V) \setminus np \setminus (s(V) \setminus np)$
- (211) a. Opus swam upstream [**singing** [a little song]].
 b. Singing a little song, Opus swam upstream.
 c. $s(pred) \setminus np / np \implies s(V) \setminus np \setminus (s(V) \setminus np) / np$
- (212) a. The band performed [**wanting** Opus [to sing]].
 b. $s(pred) \setminus np / np / s(adv) \setminus np \implies s(V) \setminus np \setminus (s(V) \setminus np) / np / s(adv) \setminus np$
- (213) a. Opus performed [**looking** awfully red]
 b. $s(pred) \setminus np / (n / n) \implies s(V) \setminus np \setminus (s(V) \setminus np) / (n / n)$

Just as in the adnominal case, the adverbialization rule will apply to adjuncts, thus allowing modified predicative verb phrases to act as adverbials. Consider the following examples:

- (214) a. Opus was probably bored [[singing in the shower] **yesterday**].
 b. $s(pred) \setminus np \setminus (s(pred) \setminus np) \implies s(V) \setminus np \setminus (s(V) \setminus np) \setminus (s(pred) \setminus np)$

²⁷If in fact, the fronted versions of these adjuncts are not topicalized, then they could be accounted for by another lexical rule of the syntactic form:

$$s(V) \setminus np \setminus (s(V) \setminus np) \ \$ \implies s(V) / s(V) \ \$.$$

This rule would allow for arbitrary post-verbal adverbs to be fronted, but not create unbounded dependencies where they could modify embedded clauses.

- (215) a. Opus showed his skill yesterday [swimming [**in** [the ocean]]].
b. $s(pred) \setminus np \setminus (s(pred) \setminus np) / np \implies s(V) \setminus np \setminus (s(V) \setminus np) \setminus (s(pred) \setminus np) / np$
- (216) a. Opus [set a new record] [swimming [**after** [Binkley danced]]].
b. $s(pred) \setminus np \setminus (s(pred) \setminus np) / s(fin)$
 $\implies s(V) \setminus np \setminus (s(V) \setminus np) \setminus (s(pred) \setminus np) / s(fin)$
- (217) a. Opus [was happy] [swimming [**while** [watched by Binkley]]].
b. $s(pred) \setminus np \setminus (s(pred) \setminus np) / (s(pred) \setminus np)$
 $\implies s(V) \setminus np \setminus (s(V) \setminus np) \setminus (s(pred) \setminus np) / (s(pred) \setminus np)$

Not counting the circular derivations derived from the interaction between predicative adnominalization and adnominal predication, the lexicon generated from a finite base lexicon would always be finite. But with the inclusion of the predicative adverbialization rule, a lexicon with an infinite number of distinct categories will be generated, since the rule will apply to its own output to form a larger category in terms of the number of complements that it takes. Simply consider the following initial segment of an infinite derivation chain:

- (218) $s(pred) \setminus np$
 $\implies s(pred) \setminus np \setminus (s(pred) \setminus np)$
 $\implies s(pred) \setminus np \setminus (s(pred) \setminus np) \setminus (s(pred) \setminus np)$
 $\implies s(pred) \setminus np \setminus (s(pred) \setminus np) \setminus (s(pred) \setminus np) \setminus (s(pred) \setminus np)$
 $\implies \dots$

Examples employing the first few elements of this sequence are as follows:

- (219) a. Opus performed the piece **shaking**.
b. Opus performed the piece [[singing the words] **shaking**].
c. Opus performed the piece \dots

As with most constructions that can be nested, the acceptability of the sentences drops off quickly above a few levels of embedding. The usual argument is that the constructions are grammatical, but simply difficult to process.

6 Conclusion

The grammar that finally results from the application of all of our rules will be equivalent to a fairly straightforward context free grammar, and is thus decidable. But the use of rules that produce infinite sequences of unique categorizations will not allow the entire lexicon to be pre-compiled in an implementation. Some sort of top-down information will be necessary to insure that useless categories are not generated. Unfortunately, as we prove in the appendix, arbitrary lexical rules operating over a finite lexicon can generate undecidable languages.

The benefit of the system presented here is that it is possible to retain a universal set of phrase-structure schemata, preserving the radical lexicalist hypothesis that all language-specific structure is encoded in the lexicon. The lexicon presented here should be of use to anyone working on topics such as unbounded dependency and coordination phenomena in extended categorial grammars, as it provides evidence that the basic phrase-structure of a language can be captured naturally

in terms of categorial lexical entries. Viewed from the most abstract level, a categorial lexicon is simply a method for encoding information about the complements an expression can take; the lexicon presented here shows how many constructions can be captured when this information is employed with only simple applicative categorial phrase-structure rules.

A Generative Power of Categorial Grammars with Lexical Rules

In this section, our main result will be the fact that string recognition in our language is R.E.-complete. What this means is that an arbitrary recursively enumerable language can be generated by a finite lexicon closed under a finite set of lexical rules. Furthermore, every language generated by a finite lexicon closed under a finite set of lexical rules will in fact be recursively enumerable. Of course, this means that in general, string recognition with respect to a specified grammar and lexical rule system will be undecidable in the worst case. Besides this result, we will also present a characterization of the possible parse trees that limits the categories that can arise as a linear function of the number of basic expressions in a string. Before going on to our system, we will briefly review similar results that have been found to hold for formal grammars such as GPSG, which employ context-free rules and metarules.

A.1 Generative Power of Context-Free Grammars with Metarules

The main result in this direction is a theorem of Uszkoreit and Peters (1985) which shows that context-free grammars augmented with metarules of one essential variable are R.E.-complete in the sense that they generate all and only the set of recursively enumerable languages.

A *meta context-free grammar* (MCFG) is a quintuple $G = \langle C, s, E, R, M \rangle$ where

- (220)
- C is a finite set of *category symbols*
 - s is the *start symbol*
 - E is a finite set of *basic expressions*
 - R is a finite set of *context-free rules* of one of the two following forms:
 - (Lexical Entry)
 $c \rightarrow e$ where $c \in C$ and $e \in E$
 - (Phrase Structure Rule)
 $c_0 \rightarrow c_1 c_2 \cdots c_n$ where $c_i \in C$ and $n \geq 1$
 - M is a finite set of *metarules* of the form:

$$(c_0 \rightarrow c_1 c_2 \cdots c_n X c_{n+1} c_{n+2} \cdots c_k)$$

$$\implies (d_0 \rightarrow d_1 d_2 \cdots d_m X d_{m+1} d_{m+2} \cdots d_j)$$
 where X is a special symbol which will be interpreted to range over arbitrary strings of categories and $c_i, d_j \in C$.

We think of an MCFG $G = \langle C, s, E, R, M \rangle$ as generating a possibly infinite set of phrase structure

rules $M(R)$ defined to be the minimal set such that:

$$(221) \quad \begin{aligned} &\bullet R \subseteq M(R) \\ &\bullet \text{ if } (c_0 \rightarrow c_1 \cdots c_n b_1 \cdots b_i c_{n+1} \cdots c_k) \in M(R) \\ &\quad \text{and } ((c_0 \rightarrow c_1 c_2 \cdots c_n X c_{n+1} c_{n+2} \cdots c_k) \in M \\ &\quad \quad \quad \implies (d_0 \rightarrow d_1 d_2 \cdots d_m X d_{m+1} d_{m+2} \cdots d_j)) \\ &\quad \text{then } (d_0 \rightarrow d_1 d_2 \cdots d_m b_1 b_2 \cdots b_i d_{m+1} \cdots d_j) \in M(R) \end{aligned}$$

We are thus thinking of the X as a variable ranging over arbitrary strings on the right-hand sides of rules. Acceptability of a string with respect to an MCFG G is then determined by acceptability with respect to the possibly infinite phrase structure grammar $M(R)$ in the usual way, starting from the start symbol s . Various tricks were employed in GPSG, which used this sort of metarule system, to insure that the set of rules generated remained finite, and thus generated a purely context-free grammar. The primary restriction which insured the finiteness of the result was not to use the fully closed set $M(R)$, but rather generate a finite set of rules by applying the metarules to the basic set, making sure to never apply a rule to its own output, even indirectly (Thompson 1982). Uszkoreit and Peter's theorem tells us that things can be much worse in the general case.

Theorem 1 (Uszkoreit and Peters) *If L is a recursively enumerable language then there is a meta-context-free grammar $G = \langle C, s, E, R, M \rangle$ such that the language generated by the phrase-structure grammar $M(R)$ is exactly L .*

The proof of this theorem employs an effective reduction of an arbitrary generalized rewriting system to a context-free grammar and set of metarules that generates exactly the same language. That is, for every generalized rewriting system, an MCFG could be found that generates exactly the same set of strings and conversely.

In the proofs presented below, we use a direct reduction from generalized rewriting systems, so we pause to define them now. A *generalized rewriting grammar* $G = \langle V, s, T, R \rangle$ is a quadruple such that V is a finite set of non-terminal category symbols, $s \in V$ is the start symbol, T is a set of terminal expression symbols, and $R \subseteq (V^* \times V^*) \cup (V \times T)$ is a finite set of rewriting rules and lexical rules, which are usually written in the forms:

$$(222) \quad \begin{aligned} &\bullet v_1 \cdots v_n \rightarrow v'_1 \cdots v'_m \text{ where } v_i, v'_j \in V \\ &\bullet v \rightarrow t \text{ where } v \in V \text{ and } t \in T. \end{aligned}$$

String rewriting is defined so that:

$$(223) \quad x_1 \cdots x_n \sigma x_{n+1} \cdots x_{n+m} \rightarrow x_1 \cdots x_n \tau x_{n+1} \cdots x_{n+m}$$

if $\sigma \rightarrow \tau \in R$ is a rule, where σ and τ are strings in V^* . The language $\mathbf{L}(G)$ generated by a general rewriting system G is defined to be

$$(224) \quad \mathbf{L}(G) = \{\sigma \in T^* \mid s \xrightarrow{*} \sigma\}$$

where s is the start symbol and $\xrightarrow{*}$ is the transitive closure of the \rightarrow relation. It is well known that:

Theorem 2 *A language L is recursively enumerable if and only if there is a generalized rewriting grammar $G = \langle V, s, T, R \rangle$ such that $L = \mathbf{L}(G)$.*

Thus, the problem of generalized rewriting system recognition is R.E.-complete. MCFG recognition is just as hard as recognizing arbitrary recursively enumerable languages, since every recursively enumerable language can be expressed as an MCFG. Of course, MCFG recognition is no harder, since all possible derivations can be easily enumerated by considering derivations in order of complexity.

A.2 Categorical Grammars with Lexical Rules

To formally define our system, we will say that a *categorical grammar with lexical rules (CG+L)* is a tuple $G = \langle \text{Exp}, s, \text{BasCat}, \Lambda, L \rangle$ with a finite set Exp of basic expressions, finite set BasCat of basic categories, a start symbol $s \in \text{BasCat}$, a finite lexicon Λ where $\Lambda \subseteq \text{Exp} \times \text{Cat}(\text{BasCat})$ and set of lexical rules L of the form:

$$(225) \ a_0 \mid_1^a a_1 \cdots \mid_i^a a_i \ \$ \mid_1^b b_1 \cdots \mid_j^b b_j \implies c_0 \mid_1^c c_1 \cdots \mid_k^c c_k \ \$ \mid_1^d d_1 \cdots \mid_m^d d_m$$

where \mid_j^i is a forward or backward slash. In this section, we will only be concerned with the syntactic portion of our categorical grammars and lexical rule systems.²⁸ We will assume that a CG+L grammar $G = \langle \text{Exp}, \text{BasCat}, s, \Lambda, L \rangle$ generates a possibly empty set of lexical entries closed under the lexical rules by taking the least set $L(\Lambda)$ such that:

$$(226) \quad \begin{aligned} &\bullet \ \Lambda \subseteq L(\Lambda) \\ &\bullet \ \text{if } a_0 \mid_1^a a_1 \cdots \mid_i^a a_i \mid_1^e e_1 \cdots \mid_n^e e_n \mid_1^b b_1 \cdots \mid_j^b b_j \rightarrow w \in L(\Lambda) \\ &\quad \text{and } a_0 \mid_1^a a_1 \cdots \mid_i^a a_i \ \$ \mid_1^b b_1 \cdots \mid_j^b b_j \in L \\ &\quad \implies c_0 \mid_1^c c_1 \cdots \mid_k^c c_k \ \$ \mid_1^d d_1 \cdots \mid_m^d d_m \\ &\quad \text{then } c_0 \mid_1^c c_1 \cdots \mid_k^c c_k \mid_1^e e_1 \cdots \mid_n^e e_n \mid_1^d d_1 \cdots \mid_m^d d_m \rightarrow w \in L(\Lambda) \end{aligned}$$

We assume exactly the same application phrase structure schemata:

$$(227) \quad \begin{aligned} &\bullet \ \alpha \rightarrow \alpha / \beta \ \beta \quad (\text{forward application}) \\ &\bullet \ \alpha \rightarrow \beta \ \alpha \setminus \beta \quad (\text{backward application}) \end{aligned}$$

where $\alpha, \beta \in \text{Cat}(\text{BasCat})$, and generate analyses in the usual way according to our now possibly infinite set of rules and lexical entries. Note that we can now no longer infer that the set of rule instances necessary will be finite, because the lexical rules can generate an infinite number of unique categories, as could be seen with the predicative adverbialization rule.

We present two theorems, the first of which characterizes the complexity of the output of lexical rules and the second of which characterizes the weak generative power of the CG+L grammar formalism.

²⁸The semantic function attached to this rule would have to be of the form:

$$\lambda \phi^{(\tau(b_j), \dots, \tau(b_1)\tau(y_n), \dots, \tau(y_1), \tau(a_i), \dots, \tau(a_1))} . x_m^{\tau(d_m)} \dots x_1^{\tau(d_1)} . y_n \dots y_1 . z_k^{\tau(c_k)} \dots z_1^{\tau(c_1)} . \\ \mathbf{R}(\lambda w_j^{\tau(b_j)} \dots w_1^{\tau(b_1)} . \phi(w_j) \cdots (w_1)(y_n) \cdots (y_1)(x_m) \cdots (x_1)(z_k) \cdots (z_1))$$

where \mathbf{R} is a semantic constant of the appropriate type.

Argument Complexity Bounds

The complexity of a category is measured in terms of the number of complements it takes to result in a basic category. The complexity of a category is given as follows:

- (228) • $C(\alpha) = 0$ if $\alpha \in \text{BasCat}$
 • $C(\alpha / \beta) = C(\alpha \setminus \beta) = 1 + C(\alpha)$

Thus the complexity of $\alpha_0 \mid_1 \alpha_1 \cdots \mid_n \alpha_n$ is n if α_0 is a basic category. In the following theorem, we show that there is a finite bound to the complexity of arguments, but no upper bound to the complexity of the overall category resulting from closing a finite lexicon under a finite set of lexical rules.

Theorem 3 *Given a finite categorial grammar with lexical rules $G = \langle \text{Exp}, s, \text{BasCat}, \Lambda, L \rangle$ there is a bound k such that the result $L(\Lambda)$ of closing the categorial grammar under the lexical rules contains only lexical entries with arguments of complexity less than k .*

Proof: Since there are only a finite number of lexical entries in Λ , there will be a bound on the maximal complexity of arguments in the core lexicon. Since there are only a finite number of lexical rules in L , there will be a finite bound on the maximal complexity of arguments in the output to lexical rules. Since lexical rules can only produce outputs whose arguments were in the input or in the lexical rule, there will be a finite bound for the resulting grammar. \square

Note that it is possible to derive categories of unbounded complexity, as seen with (218), it is not possible to derive categories with *arguments* of unbounded complexity.

It should be noted that every derivation tree rooted at the start symbol s for a string $e_1 e_2 \cdots e_n \in \text{Exp}^*$ of length n cannot involve a main functor category of complexity greater than n , since the complexity of the mother is only going to be one less than the complexity of the functional daughter. Together with the previous theorem, this gives us an upper bound on the number of parse trees that need to be considered for any given input string. Alas, the problem is still undecidable, as the previous theorem shows. Of course, this situation will change when extended categorial grammar systems are considered, although many of these systems provide normal form derivation results that allow every derivation to be carried out within some complexity bound on the size of the categories based on the size of the input string.

Decidability

In this section, we show how to effectively reduce an arbitrary generalized rewriting grammar to a categorial grammar with lexical rules. Since it should be obvious that categorial grammar recognition with lexical rules is a recursively enumerable problem, we get the following:

Theorem 4 *A language S is recursively enumerable if and only if there is a CG+L grammar $G = \langle \text{Exp}, s, \text{BasCat}, \Lambda, L \rangle$ such that $\{\text{start} \cdot \sigma \mid \sigma \in S\}$ is the set of strings generated from s with the lexicon $L(\Lambda)$.*

Proof: We proceed by a reduction of generalized rewriting grammars. Suppose that we have a generalized rewriting system $G = \langle V, s, T, R \rangle$. We will show how to construct a weakly equivalent categorial grammar $G' = \langle \text{Exp}, s', \text{BasCat}, \Lambda, L \rangle$. We begin by assuming that

$$(229) \text{BasCat} = V \cup \{t' \mid t \in T\} \cup \{\#\}.$$

We have just added a basic category t' for every terminal symbol t in the rewriting system.

We will represent an arbitrary string $v_1 v_2 \cdots v_n \in V^*$ by means of the categorial grammar category $\#/v_n \cdots /v_1$, where the $\#$ symbol is just an arbitrary end marker.

We will then need a pair of special lexical rules of the form:

$$(230) \begin{aligned} &\bullet v_1 \$ /v_2 \implies v_2 /v_1 \$ \\ &\bullet v_2 /v_1 \$ \implies v_1 \$ /v_2 \end{aligned}$$

for each $v_1, v_2 \in V \cup \{\#\}$. These will mean that we will also get a lexical entry of the form:

$$(231) v_{m+1}/v_{m+2} \cdots /v_n/\#/v_1/v_2 \cdots /v_m \rightarrow e$$

for every lexical entry we have of the form:

$$(232) \#/v_1 \cdots /v_n \rightarrow e$$

where $0 \leq m \leq n$. Furthermore, for every rule in the rewriting system R of the form

$$(233) v_1 \cdots v_n \rightarrow v'_1 \cdots v'_m$$

we will take a lexical rule of the form

$$(234) \$ /v_n/v_{n-1} \cdots /v_1 \implies \$ /v'_m/v'_{m-1} \cdots /v'_1 \in L$$

Given our previous observation, this means that if we have

$$(235) x_1 \cdots x_i v_1 \cdots v_n y_1 \cdots y_j \rightarrow x_1 \cdots x_i v'_1 \cdots v'_m y_1 \cdots y_j$$

then we will have

$$(236) \begin{aligned} &\#/y_j \cdots /y_1/v_n \cdots /v_1/x_i \cdots /x_1 \\ &\implies x_i \cdots /x_1/\#/y_j \cdots /y_1/v_n \cdots /v_1 \\ &\implies x_i \cdots /x_1/\#/y_j \cdots /y_1/v'_m \cdots /v'_1 \\ &\implies \#/y_j \cdots /y_1/v'_m \cdots /v'_1/x_i \cdots /x_1 \end{aligned}$$

A simple induction then gives us the fact that

$$(237) x_1 \cdots x_i \xrightarrow{*} y_1 \cdots y_j \text{ if and only if } \#/x_i \cdots /x_1 \xrightarrow{*} \#/y_j \cdots /y_1$$

where $\xrightarrow{*}$ is simply the transitive closure of the lexical rule derivation relation.

Finally, we assume that we have a lexical entry of the form:

$$(238) \#/s \rightarrow \text{start} \in \Lambda$$

to account for the initialization and lexical entries of the form

$$(239) t' \rightarrow t \in \Lambda$$

This means that we have

$$(240) s \xrightarrow{*} v_1 \cdots v_n$$

if and only if

$$(241) \#/v_1 \cdots /v_n \rightarrow start$$

With the lexical entries that we have, we only need the lexical rule:

$$(242) \# \$ \Longrightarrow s' \$$$

where s' is the start symbol of the categorial grammar G' , which will insure that the categorial grammar $L(\Lambda)$ generates exactly the set $\{start \cdot t_1 \cdots t_n \mid t_1 \cdots t_n \in \mathbf{L}(G)\}$ of strings. \square

NOTES

Acknowledgements

The primary motivation for this paper was the long hours spent working out categorial grammar fragments for English with Glyn Morrill. Roughly similar though less extensive lexicons can be found in both of our doctoral dissertations, both of which were concerned with other matters (Glyn's with unbounded dependencies and coordination, and mine with event based semantics and quantification). Our goal was to develop and implement categorial grammars with broader coverage than other formal grammars. Our primary methodology was a minimalist approach that only invoked those mechanisms which were absolutely necessary. The resulting grammar was obviously greatly influenced by LFG, GPSG and HPSG and other categorial and unification grammar analyses.

The fragment presented in this paper has evolved over a number of years and presentations. Comments were kindly received at the University of Edinburgh, University of Pennsylvania, University of Pittsburgh, University of California – Los Angeles, The Ohio State University, Xerox Palo Alto Research Center and the Center for the Study of Language and Information at Stanford University. Mark Steedman and Ewan Klein deserve special thanks in this regard for sitting through more than one presentation of this material. Mark also deserves extra credit for inventing the best notation in which to display categorial grammar categories and trees. I should also thank Stu Shieber for suggesting that the lexical rule system was undecidable, but the primary thanks along formal lines go to Kevin Kelly, who saw how the reduction in the proof should be carried out. Finally, I would like to thank Alex Franz and Mitzi Morris for reading the final draft and providing a number of useful suggestions.

References

- Ades, A.E. and M.J. Steedman, 1982. On the order of words. *Linguistics and Philosophy* **4**, pps. 517–558.
- Aho, A. V. and J. D. Ullman, 1972. *The Theory of Parsing, Translation and Compiling*. Prentice Hall, Englewood Cliffs, New Jersey.
- Ajdukiewicz, K., 1935. Die syntaktische Konnexität. *Studia Philosophica* **1**, 1–27.

- Bach, E., 1980. In defense of passive. *Linguistics and Philosophy* **3**, 297–341.
- Bach, E., 1983. On the relationship between word-grammar and phrase-grammar. *Natural Language and Linguistic Theory* **1**, 65–89.
- Bach, E., 1983b. Generalized categorial grammars and the English auxiliary. In Heny, F. and B. Richards, editors, *Linguistic Categories, Auxiliaries, and Related Puzzles – II*, D. Reidel, Dordrecht.
- Bach, E., 1984. Some generalizations of categorial grammars. In Landman, F. and F. Veltman, editors, *Varieties of Formal Semantics: Proceedings of the Fourth Amsterdam Colloquium, September 1982*, Foris, Dordrecht.
- Bach, E. and R. Cooper, 1978. The NP-S analysis of relative clauses and compositional semantics. *Linguistics and Philosophy* **2**, 265–299.
- Bar-Hillel, Y., 1953. A quasi-arithmetical notation for syntactic description. *Language* **25**.
- Bar-Hillel, Y., C. Gaifman, and E. Shamir, 1960. On categorial and phrase structure grammars. *Bulletin of the Research Council of Israel* **9F**, 1–16.
- van Benthem, J., 1986. Categorial grammar. In *Essays in Logical Semantics*, D. Reidel, Dordrecht.
- van Benthem, J., 1986b. Categorial grammar and lambda calculus. In Skordev, V., editor, *Druzhba Summer School in Applied Logic*, Plenum Press, New York.
- van Benthem, J., 1987. Semantic type change and syntactic recognition. In Chierchia, G., B. Partee, and R. Turner, editors, *Categories, Types and Semantics*, D. Reidel, Dordrecht.
- Bresnan, J.W., 1972. Theory of complementation in English syntax. MIT Ph.D. Thesis. Also published 1979, Garland Press, New York.
- Bresnan, J. W., editor, 1982. *The Mental Representation of Grammatical Relations*, MIT Press, Cambridge, MA.
- Bresnan, J. W., 1982b. Control and complementation. *Linguistic Inquiry* **13**, 343–434.
- Bresnan, J. W., 1982c. The passive in lexical theory. In Bresnan (1982), pages 3–86.
- Bresnan, J. W., 1982d. Polyadicity. In Bresnan (1982), pages 149–172.
- Carpenter, B., 1989. *Phrase Meaning and Categorial Grammar*. Ph.D. Dissertation, University of Edinburgh, Center for Cognitive Science.
- Chierchia, G. and R. Turner, forthcoming. Semantics and property theory. *Linguistics and Philosophy*.
- Chomsky, N. and H. Lasnik, 1977. Filters and control. *Linguistic Inquiry* **8**, 425–504.
- Church, A., 1940. A formulation of a simple theory of types. *Journal of Symbolic Logic* **5**, 56–68.

- Dowty, D. R., 1978. Lexically governed transformations as lexical rules in a Montague grammar. *Linguistic Inquiry* **9**, 393–426.
- Dowty, D. R., 1979. *Word Meaning and Montague Grammar*. Synthese Language Library, Vol. 7, D. Reidel, Dordrecht.
- Dowty, D. R., 1982. Grammatical relations and Montague Grammar. In P. Jacobson and G.K. Pullum, eds., *the Nature of Syntactic Representation*, pps. 79–130, Reidel, Dordrecht.
- Dowty, D. R., 1988. Type-raising, functional composition and non-constituent coordination. In Oehrle, R. T. et al., eds., 1988.
- Gazdar, G., 1980. A cross-categorial semantics for coordination. *Linguistics and Philosophy* **3**, pages 407–409.
- Gazdar, G., 1981. On syntactic categories. *Philosophical Transactions of the Royal Society (Series B)* **295**, 267–283.
- Gazdar, G., 1981b. Unbounded dependencies and coordinate structure. *Linguistic Inquiry* **12**, 155–184.
- Gazdar, G., 1982. Phrase structure grammar. In P. Jacobson and G.K. Pullum, eds., *The Nature of Syntactic Representation*. Dordrecht, Reidel.
- Gazdar, G., E. Klein, G. Pullum, and I. Sag, 1985. *Generalized Phrase Structure Grammar*. Basil Blackwell, Oxford.
- Gazdar, G. and G. Pullum, 1981. Subcategorization, constituent order and the notion of ‘Head’. In Moortgat, M., H. van der Hulst, and T. Hoekstra, editors, *The Scope of Lexical Rules*, pages 107–123, Foris, Dordrecht.
- Gazdar, G., G. K. Pullum, and I. A. Sag, 1982. Auxiliaries and related phenomena in a restrictive theory of grammar. *Language* **58**, 591–638.
- Geach, P. T., 1972. A program for syntax. *Synthese* **22**, 3–17.
- Haddock, N. J., 1987b. Incremental interpretation and combinatory categorial grammar. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 661–663.
- Hendriks, H., 1987. Type change in semantics: the scope of quantification and coordination. In Klein, E. and J. van Benthem, editors, *Categories, Polymorphism and Unification*, pages 95–120, Centre for Cognitive Science, University of Edinburgh, Edinburgh.
- Hoeksema, J., 1987. Semantics of non-boolean “and”. *Journal of Semantics* **6**, pages 19–40.
- Hoeksema, J. and R.D. Janda, 1988. Implications of process morphology. In Oehrle, R. T. et al., eds., 1988.
- Hopcroft, J. and J. Ullman, 1979. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, Massachusetts.

- Kaplan, R. and J. Bresnan, 1982. Lexical-functional grammar: a formal system for grammatical representation. In Bresnan (1982).
- Karttunen, L., 1984. Features and values. In *Proceedings of Coling 1984*.
- Karttunen, L., 1986. *Radical Lexicalism*. Report CSLI-86-68, Center for the Study of Language and Information, Stanford University.
- Keenan, E.L. and A. Timberlake, 1988. Natural language motivations for extending categorial grammar. In Oehrle, R. T. et al., eds., 1988.
- Klein, E. and I. A. Sag, 1985. Type-driven translation. *Linguistics and Philosophy* **8**, 163–201.
- Lambek, J., 1958. The mathematics of sentence structure. *American Mathematical Monthly* **65**, 154–169.
- Lambek, J., 1961. On the calculus of syntactic types. In Jakobson, R., editor, *Structure of Language and its Mathematical Aspects, Proceedings of Symposia in Applied Mathematics*, pages 166–178, American Mathematical Society, Providence, Rhode Island.
- Levin, L., 1987. *Toward a Linking Theory of Relation Changing Rules in LFG*. Report CSLI-87-115, Center for the Study of Language and Information, Stanford University.
- Montague, R., 1974c. Universal grammar. In Thomason, R., editor, *Formal Philosophy*, pages 222–246, Yale University Press, New Haven.
- Moortgat, M., 1984. A Fregean restriction on metarules. *Proceedings of NELS 14*.
- Moortgat, M., 1987. *Parsing Lambek Categorial Grammars*. Report INL WP 87-05, Instituut voor Nederlandse Lexicologie, Leiden.
- Moortgat, M., 1987b. Lambek theorem proving. In Klein, E. and J. van Benthem, editors, *Categories, Polymorphism and Unification*, pages 169–200, Centre for Cognitive Science, University of Edinburgh, Edinburgh.
- Moortgat, M., 1987c. Compositionality and the syntax of words. In Groenendijk, J., D. de Jongh, and M. Stokhof, editors, *Foundations of Pragmatics and Lexical Semantics*, Foris, Dordrecht.
- Moortgat, M., 1988. Mixed composition and discontinuous dependencies. In Oehrle, R. T. et al., eds., 1988.
- Moortgat, M., 1988b. *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus*. Academisch Proefschrift (Dissertation), University of Amsterdam.
- Morrill, G., 1987. Meta-categorial grammar. In Morrill, G., N. J. Haddock, and E. Klein, editors, *Categorial Grammar, Unification Grammar and Parsing*, Centre for Cognitive Science, University of Edinburgh, Edinburgh.
- Morrill, G., 1987b. Phrase structure grammar and categorial grammar. In Klein, E. and J. van Benthem, editors, *Categories, Polymorphism and Unification*, pages 201–277, Centre for Cognitive Science, University of Edinburgh, Edinburgh.

- Morrill, G., 1988. *Extraction and Coordination in Phrase Structure Grammar and Categorical Grammar*. PhD thesis, University of Edinburgh.
- Morrill, G. and B. Carpenter, forthcoming. Compositionality, implicational logics and theories of grammar. *Linguistics and Philosophy*.
- Oehrle, R., E. Bach, and D. Wheeler, editors, 1988. *Categorical Grammars and Natural Language Structures*. D. Reidel, Dordrecht.
- Partee, B., 1975. Montague grammar and transformational grammar. *Linguistic Inquiry* **6**, 203–300.
- Pereira, F. C. N. and D. H. D. Warren, 1980. Definite clause grammars for language analysis — a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence* **13**, 231–278.
- Pollard, C. J., 1988. Phrase-structure grammars and categorial grammars: an excursion on the syntax-semantics frontier. In Oehrle, R. T. et al., eds., 1988.
- Pollard, C. J. and I. A. Sag, 1987. *Information-Based Syntax and Semantics: Volume I – Fundamentals*. CSLI Lecture Notes, Vol. 13, Chicago University Press, Chicago.
- Pollard, C. J. and I. A. Sag, forthcoming. *Information-Based Syntax and Semantics: Volume II* CSLI Lecture Notes, Chicago University Press, Chicago.
- Reape, M., 1989. A logical treatment of semi-free word order and bounded discontinuous constituency. *Proceedings of European Association for Computational Linguistics*.
- Sag, I., G. Gazdar, T. Wasow and S. Wexler, 1985. Coordination and how to distinguish categories. *Natural Language and Linguistic Theory* **3**, 117–171.
- Schönfinkel, M., 1924. Über die Bausteine der mathematischen Logik. *Mathematische Annalen* **92**, 305–316.
- Siekman, J.H., 1989. Unification theory, *Journal of Symbolic Computation* **7**, pps. 207–274.
- Steedman, M., 1985. Dependency and coordination in the grammar of Dutch and English. *Language* **61**, 523–568.
- Steedman, M., 1987. Combinatory grammars and parasitic gaps. In Morrill, G., N. J. Haddock, and E. Klein, editors, *Categorial Grammar, Unification Grammar and Parsing*, Centre for Cognitive Science, University of Edinburgh, Edinburgh.
- Steedman, M., 1988. Combinators and grammars. In Oehrle, R. T. et al., eds., 1988.
- Steedman, M., 1989. Prosody and Combinatory Grammars, University of Pennsylvania, manuscript
- Thompson, H., 1984. Handling metarules in a parser for GPSG. Proceedings of the 21st Meeting of the ACL, pages 26–37.
- Uszkoreit, H., 1986. Categorial unification grammars. In *Proceedings of Coling 1986*, University of Bonn, Bonn.

- Uszkoreit, H. and P. S. Peters, 1986. *On Some Formal Properties of Metarules*. Report CSLI-85-43, Center for the Study of Language and Information, Stanford University, Stanford.
- Weisler, S., 1982. Coordination and the syntax of that-clauses. In A. Prince and S. Weisler (eds.), *U. Mass. Amherst Occasional Working Papers in Cognitive Science*, 112–134. Amherst, University of Massachusetts.
- Wittenberg, K. B., 1986. *Natural Language Parsing with Combinatory Categorical Grammar in a Graph-Unification Based Formalism*. PhD thesis, University of Texas, Austin.
- Zeevat, H., E. Klein, and J. Calder, 1987. Unification categorial grammar. In Morrill, G., N. J. Haddock, and E. Klein, editors, *Categorical Grammar, Unification Grammar and Parsing*, Centre for Cognitive Science, University of Edinburgh, Edinburgh.

Contents

1	Introduction	1
2	Applicative Categorical Grammar	2
2.1	Category System	2
2.2	Semantic Domains	4
2.3	Lexical Assignments	5
2.4	Application Schemes	5
3	Basic English Lexicon	6
3.1	Simple Noun Phrases	6
3.2	Simple Verb Phrases	8
3.3	Basic Modifiers	10
3.4	Auxiliaries	13
3.5	Complemented Categories	16
4	Lexical Rules	27
5	Predicatives	33
5.1	Passivization	33
5.2	Adnominal Predicatives	39
5.3	Nominal Predicatives	41
5.4	Sentential Coordination and Predicatives	42
5.5	Predicatives as Adjuncts	43
6	Conclusion	47
A	Generative Power of Categorical Grammars with Lexical Rules	48
A.1	Generative Power of Context-Free Grammars with Metarules	48
A.2	Categorical Grammars with Lexical Rules	50