New Generation Computing, 3 (1985) 273-306
OHMSHA, LTD. and Springer-Verlag

NEW GENERATION COMPUTING

Computationally Relevant Properties of Natural Languages and Their Grammars

Gerald GAZDAR
Cognitive Studies Programme, University of Sussex
Brighton BN1 9QN, England
Geoffrey K. PULLUM
University of California, Santa Cruz
Santa Cruz, Ca. 95064, USA

Received 25 February 1985
Revised manuscript received 3 June 1985

Abstract This paper surveys what is currently known about natural language morphology and syntax from the perspective of formal language theory. Firstly, the position of natural language word-sets and sentence-sets on the formal language hierarchy is discussed. Secondly, the contemporary use by linguists of a range of formal grammars (from finite state transducers to indexed grammars) in both word-syntax (i. e. morphology) and sentence-syntax is sketched. Finally, recent developments such as feature-theory, the use of extension and unification, default mechanisms, and metagrammatical techniques, are outlined.

Keywords: Syntax, Parsing, Grammar, Natural Language, Morphology, Formal Language Theory, Features.

§1 Introduction*

Our starting assumption is that as computers continue to increase in complexity and functionality by orders of magnitude, it will in due course become not just desirable but actually necessary for them to have command of

* Readers interested in pursuing the matters discussed in this paper in greater detail than space has allowed us are directed to Gunji[46] and Perrault[113] for excellent surveys of the relevant linguistics and mathematics, respectively. We are grateful to Takao Gunji, Lauri Karttunen, Jerrold Sadock, and Stuart Shieber for conversations and comments on topics covered by the present paper. The paper was prepared while the authors were visiting the Center for the Study of Language and Information (CSLI) at Stanford University, and whilst Gazdar was a fellow at the Center for Advanced Study in the Behavioral Sciences (CASBS), Stanford. Support was provided by a gift to CSLI from the System Development Foundation (Pullum), by grants from the Sloan Foundation and the System Development Foundation to CASBS (Gazdar), and by an ESRC personal research grant to Gazdar.

natural languages (henceforth NLs). They will need NL ability if they are to be used to their full capacity by human beings, whether expert or not. In this paper we review some crucial things that must be kept in mind as the necessary research and development is done to make it possible for computers to attain competence in NLs.

An article such as this cannot cover the entire extent of the field of natural language processing (NLP). We focus here on grammar: syntax, morphology, and lexicon. Omitted from our discussion for reasons of space are considerations having to do with the two endpoints of a linguistic system: meaning and sound. We have too little to say about computational implementation in semantics to merit a section here, but clearly this is a topic of fundamental importance that needs to be addressed at length. The same can be said for the crucially interrelated areas of pragmatics and multi-sentence discourse. Also omitted is any consideration of the recognition or synthesis of speech. What we do offer in this paper is a survey of recent results in the theory of NLs and their grammars, with an emphasis on issues and properties that appear to be of computational relevance.

There is a crucial connection between the theory of parsing and the formal theory of languages: *There can be no parsing without a grammar.* There are two senses in which this is true, we believe. To begin with, it is true trivially, in that a working parser for any language automatically instantiates a definition of its membership, and hence necessarily embodies a grammar. But there is a less trivial sense in which we must recognize that parsing implies the existence of a grammar. It is clear enough in the literature on the definition and parsing of programming languages, but it has often been denied in the context of the much larger and richer muti-purpose languages spoken by humans. As we shall hope to show, for serious, theoretically-based reasons, engineering in a domain as complex as NL will have to be based on what linguists can determine about the structure of languages.

The definition of the language that a parser instantiates need not by any means be a perspicuous one. Moreover, it will be an implementation-specific definition; and implementations — even implementations of a programming language that is thought to be well understood — can differ significantly and unexpectedly. It is for this reason that computer scientists have turned in recent years away from procedural definitions of the semantics for programming languages and toward denotational semantics (see e. g. Stoy.[152]). Rather similar considerations hold when we consider writing programs that process NL input: both syntactically and semantically, we need to have a secure defintion of the NL (or approximation to a NL) that we are processing if we are to have any idea how the system should behave under a wide range of conditions. Parsability is thus connected to definability, and it is therefore essential for parser-designers to pay attention to the grammar for the language they are trying to parse.

In assessing whether some formal theory of grammar is an adequate

theory for NLs, at least the following two criteria are relevant:

(1) Does it permit NLs, considered as sets of strings, to be defined?
(2) Does it permit significant generalizations about the defined NLs to be expressed?

In the second section of this paper, we look at the current state of knowledge in respect of the first of these questions. In the third section, we look at the use linguists have made of the grammatical tools made available by formal language theory, and make reference to the parsers that have been constructed with the help of these tools. In the fourth section, we look selectively at some recent syntactic developments that are addressed to the second question.

§2 Language Types

NLs can be regarded under a useful idealization as sets of strings of symbols, and can thus be made amenable to mathematical analysis of a straightforward kind. In this section, we look first at NL lexicons (i. e. word sets) construed as sets of strings defined over a vocabulary of atomic morphological elements, and then, in the second subsection, at NLs themselves (i. e. sentence sets), which are standardly construed as sets of strings defined over the set of well-formed words. (Notice that in formal language theory, the term **word** denotes a string that is a member of a language, and linguists use the term **sentence** for this; when we speak of words in this paper, we always mean words in a dictionary sense, and in the syntax of a NL these correspond to the atomic symbols of the terminal vocabulary.) Occasionally, when it is relevant to do so, we will make a comment about the properties of the sets of structural representations or parses that are associated with the strings. However, our primary concern in this section is with sets of strings.

2.1 Words

If we regard a sentence of a NL as a string of words, then there is a fundamental difference between the formal nature of NLs and the usual formalizations of computer programming languages. Although there may be structure to the individual symbols such as names of constants and variables that are the terminal symbols in the grammar of a programming language, that structure is usually trivial. For example, a variable name may be simply any string of alphanumeric symbols that does not appear on the list of reserved words. Moreover, such structure as a terminal symbol has is simply a matter of concatenation of symbols, and has no contribution to make to the syntax.

Things are different with the set of words in a NL. Words are not trivial in their structural properties. In many cases they have a complex internal constituent structure and a set of idiosyncratic properties that are crucial for determining aspects of the syntactic structure of sentences. Consider one simple example from English. The verb *donate* has a stem *don* (also appearing in

donor) and a verb-forming derivational suffix-*ate* (also appearing in *translate*); we can regard these as atomic symbols. The usual linguistic terminology for such word components is **morpheme**. *Donate* also has the property (shared with the synonym *give*) of syntactically demanding cooccurrence with a noun phrase and a prepositional phrase with *to* (as in *donate* this money to the church); it has the property (not shared by *give*) of not co-occurring with two following NP's (we do not find **donate the church some money* [the prefixed asterisk being used here to indicate a string disallowed by the grammar]); it has the property of not allowing intransitive use (contrasting with *translate*: compare *How does this translate into Japanese?* with the ungrammatical **When did this money donate to the church?*; it forms a related noun ending in the morpheme *-ion* denoting the act or result of donating (unlike, say, *berate*; donating something constitutes a donation, but berating someone does not constitute a **beration*); and so on.

Many items in many languages have much more complicated lexical properties than this; for example, all of the inflectional morphology of *donate* is predictable (*donate, donates, donated, donating*), whereas this is not at all true of the verb *be* (cf. *am, are, is, was, were, been, being*). Defining the complete sets of lexical items in a NL with all their internal structure and associated properties is a nontrivial language-definition task. The following subsections deal with various logical possibilities concerning the character of such languages.

[1] Finite languages

Do all languages have a finite lexicon? The common sense answer is "yes"; after all, dictionaries contain all the words in a language, and, while dictionaries may be very long (the *Oxford English Dictionary* runs to 12 very large volumes), they are not infinitely long. But the common sense answer is incorrect: there are few if any languages whose dictionaries contain all the words of the language. No Finnish dictionary contains all the possible forms of Finnish verbs — each one has around 10,000 inflected forms. In languages (such as certain American Indian languages, e. g. Tuscarora) that allow a noun stem to be incorporated into a verb stem, the number of distinct inflected forms for each verb goes into the millions.

However, this example only shows the premise on which the common sense answer is based to be false. It does not cast doubt on the answer itself. But we do not have to look far to find the evidence we need. Most languages employ word-formation processes that can apply iteratively to each other's output, and, in so doing, trivially induce an infinite language. Some lexical items are made up by compounding stems from a technical vocabulary (e. g. *deoxyribonucleic*) or by compounding out of whole words (*CPU-cycle-consumptive*) or by reduplicating affixes or stems (*anti-anti-missile-missile-missile*). Noting this, Langendoen[94] posed the question of what the power of the word-formation component of the grammar had to be. Since, as we have just seen, NL lexicons are typically

not finite, that grammar cannot simply be a list. In the following subsections we explore the alternatives to a list.

〔2〕 Finite state languages

Langendoen[94] raises in a short note the issue of whether infinite word-sets in NLs are always regular sets like, e. g., the set *(great-)*-grandparent* in English (a great-great-grandparent is the parent of a great-grandparent). He notes an incorrect claim to the contrary by Bar-Hillel and Shamir,[6] and characterizes some unattested but imaginable situations which, if found in the morphology of a NL would render the word-sets non-finite-state. He notes that actual word-sets encountered in NLs up to that time had apparently always been finite-state, though there was no reason in principle why they should be.

〔3〕 Context-free languages

Langendoen[94] also points out that certain patterns of prefixation and suffixation could in principle lead to a non-context-free (non-CF) word-set, yet (a fortiori, given the claim of the previous subsection) no language yet known appeared to have a non-CF word-set. He observes that if certain prefixes demanded the presence of certain suffixes, non-finite-state word-sets of (e. g.) the type $\{a^m cb^n \mid m = n + 1\}$ could result. He also notes that if substrings of arbitrary length could be reduplicated (doubled), word-sets that were not even CF could be derived.

Of both the finite-stateness property and the CF-ness property, Langendoen asks whether the absence of NL word-sets lacking them is accidental, or whether it is "a consequence of some yet-to-be formulated principles of word-formation" (p. 321).

〔4〕 Beyond the context-free languages

Facts recently reported by Culy[24] suggest that Langendoen's question can now be answered. Bambara, an African language of the Mande family, seems to have a set of words that is not a CFL. Culy notes that Bambara forms from noun stems compound words of the form "Noun-o-Noun" with the meaning "whatever N". Thus, given that *wulu* means "dog", *wulu-o-wulu* means "whatever dog." But Bambara also forms compound noun stems of arbitrary length; *wulu-filela* means "dog-watcher," *wulu-nyinila* means "dog-hunter," *wulu-filela-nyinila* means "dog-watcher-hunter," and so on. From this it is clear that arbitrarily long words like *wulu-filela-nyinila-o-wulu-filela-nyinila* "whatever dog-watcher-hunter" will be in the language. This is a realization of one of the hypothetical situations imagined by Langendoen,[94] in which reduplication applies to a class of stems whose members have no upper length bound. Culy provides a formal demonstration that this phenomenon renders the entire word-set of Bambara non-CF.

Alexis Manaster-Ramer has observed in unpublished lectures that other

languages offer similar phenomena; he finds reduplication constructions that appear to have no length bound in Polish, Turkish, and a number of other languages.

This discovery raises a very interesting question: how hard can the recognition problem be for words in the (typically infinite) vocabulary of a NL? In fact, we believe that no significant problem arises for known non-CF cases, all of which involve simple string reduplication. This is fairly easy to show, as pointed out to us by Carl Pollard (personal communication). Determining whether the first half of a substring is identical to its second half takes time proportional to the length of the string. A standard algorithm for parsing CFLs, such as the CKY algorithm, could therefore be modified to included an operation of this sort as well as the usual operation of comparison against right hand sides of rules. For example, if a string x is analyzable as a noun, i. e. if $N \Rightarrow^{+} x$, then a string x-o-y could be allowed also to be analyzed as a noun provided $x = y$. The CKY algorithm runs in cubic time, so the modified algorithm will too, the string-comparison adding only a linear element to the total time taken. Hence recognition of strings in a language that fails to be CF solely in virtue of the occurrence of reduplication has a time complexity no worse than the general problem of CFL recognition.

We do not know whether there exists an independent characterization of the class of languages that includes the regular sets and languages derivable from them through reduplication, or what the time complexity of that class might be, but it currently looks as if this class might be relevant to the characterization of NL word-sets.

2.2 Sentences

In this section we take NLs to be sets of sentences, and sentences to be strings of words in the linguist's sense. As in the case of NL lexicons discussed above, the question we are addressing is: what is the smallest known natural class of formal languages that can reasonably be taken to include all the NLs?

(1) Finite languages

This section will be brief since it is so obvious that NLs are not finite languages. Indeed, as far as is known, no NL is a finite language. The range of constructions that make a language infinite is typically rather large. Coordination, for example, always permits an unbounded number of conjuncts (whether this happens by iteration or by nesting is irrelevant). And, in English, for example, adjectives can be iterated indefinitely (*a nice, large, cheerful,..., well-lit room*), as can relative clauses, which can contain verb phrases which can contain noun phrases which can contain relative clauses which...

(2) Finite state languages

Chomsky's[19] claim that NLs are not in general finite-state was correct,

although his own argument for the non-regular character of English was not given in anything like a valid form, as has often been remarked (cf. Daly[25] for a thorough critique). However, the following argument, patterned after a suggestion by Brandt Corstius (see Levelt,[98] pp. 25-26), is valid. The set (1):

$$\{\text{A white male (whom a white male)}^n \text{ (hired)}^n \text{ hired another white male.} \mid n > 0\} \quad (1)$$

is the intersection of English with the regular set (2):

$$\text{A white male (whom a white male)}^* \text{ hired}^* \text{ another white male.} \quad (2)$$

(In ordinary grammatical terms, this is because each occurrence of the phrase *a white male* is a noun phrase which needs a verb such as *hired* to complete the clause of which it is the subject.) But (1) is not regular; and the regular sets are closed under intersection; hence English is not regular. *Q. E. D.*

It is perfectly possible that some NLs happen not to present the inherently self-embedding configurations that are likely to make a language non-regular. Languages in which parataxis is used much more than hypotaxis (i. e. languages in which separate clauses are strung out linearly rather than embedded) are common. However, we would expect non-regular configurations to be at least as common in the languages of the world. There are a number of languages that furnish better arguments for a non-regular character than English does; for example, according to Hagège,[48] center-embedding phenomena in grammar seem to be commoner and more acceptable in several Central Sudanic languages than they are in English.

The fact that NLs are not regular sets is both surprising and disappointing from the standpoint of parsability. It is surprising because there is no simpler way to obtain infinite languages than to admit the operations of concatenation, union, and Kleene closure on finite vocabularies, and there is no obvious a priori reason why humans could not have been well served by regular languages. And it is disappointing because if NLs were regular sets, we know we could recognize them in deterministic linear time using the fastest and simplest abstract computing device of all, the finite state machine. Of course, given any limitation to finite memory in a given machine, we are in fact doing just that, but it is not theoretically revealing to use this as the basis for an understanding of the task.

[3] Deterministic context-free languages

The finite state languages, luckily, are not the only languages that can be efficiently recognized: there are much larger classes of languages that have linear-time recognition. One such class is the deterministic CFLs (DCFLs), i. e. those CFLs that are accepted by some deterministic pushdown stack automaton. It would be reasonable, therefore, to raise the question of whether at least some NLs were DCFLs. To the best of our knowledge, this question has never previously been considered, much less answered, in the literature of linguistics

or computer science. Rich[123] is not atypical in dismissing the entire literature on DCFLs, LR parsing, and related topics without a glance on the basis of an invalid argument (from subject-verb agreement) which is supposed to show that English is not even a CFL, hence a fortiori not a DCFL.

English cannot be shown to be a non-DCFL on the grounds that it is ambiguous. Ambiguity must be carefully distinguished from inherent ambiguity. An inherently ambiguous language is one such that all of the grammars that weakly generate it are ambiguous. LR grammars are never ambiguous; but the LR grammars characterize exactly the set of DCFLs, hence no inherently ambiguous language is a DCFL. But it has never been argued, as far as we know, that English or any other NL is inherently ambiguous. Rather, it has been argued that a descriptively adequate grammar for it should, to account for semantic intuitions, be ambiguous. But obviously, a DCFL can have an ambiguous grammar; **all** languages have ambiguous grammars.

The relevance of this becomes clear when we observe that in NLP applications it is often taken to be desirable that a parser or translator should yield just a single analysis of an input sentence. One can imagine an implemented NL system in which the language accepted is properly described by an ambiguous CF-PSG but is nonetheless (weakly) a DCFL.

The idea of a deterministic parser with an ambiguous grammar, which arises directly out of what has been done for programming languages in, for example, the **yacc** system (Johnson[67]), is explored for natural languages in work by Fernando Pereira and Stuart Shieber. Shieber[144] describes an implementation of a parser which uses an ambiguous grammar but parses deterministically. The parser uses shift-reduce scheduling in the manner proposed by Pereira,[111] and uses rules for resolving conflicts between parsing actions that are virtually the same as the ones given for **yacc** by Johnson.[67]

We believe that techniques such as LR parsing which come straight out of programming language and compiler design (and which have much greater formal interest and variety than has often been recognized; see Bermudez[10] for some theoretical explorations) may be of considerable interest in the context of NLP applications. For example, Tomita[160] uses pseudo-parallelism to extend the LR technique to encompass multiple parses in NLP, and Shieber goes so far as to suggest psycholinguistic implications. Interestingly, human beings are prone to fail almost as badly as Shieber's parser on certain types of sentence that linguists would regard as grammatical (basically, sentences that lack the prefix property — that is, they have an initial proper substring which is a sentence).

[4] Context-free languages

The belief that CF-PSGs cannot cope with the structure of NLs, and hence that NLs are not CFLs, is well entrenched. Introductory linguistics textbooks and other pedagogically oriented works have falsely stated that such phenomena as subject-verb agreement show English to be non-CF (see Pullum

and Gazdar[120] for references). This is not so. Even finite state languages can exhibit dependencies between symbols arbitrarily far apart. To take an artificial example, suppose the last word in every sentence had to bear some special marking that was determined by what the first morpheme in the sentence was; a finite automaton to accept the language could simply encode in its state the information about what the sentence-initial morpheme was, and check the last word's marking against the state before accepting.

Expository works in the field of generative grammar have generally offered nothing that could be taken seriously as an argument that NLs are not CFLs. Worse, even the technical literature exhibits a quarter-century of mistaken efforts to show that not all NLs are CFLs. This history is carefully reviewed by Pullum and Gazdar.[120] In addition to the fallacies concerning agreement just mentioned, they deal with arguments based on

(1) *respectively* constructions (Bar-Hillel and Shamir[6] ; Langendoen[93])
(2) English comparative clauses (Chomsky[20])
(3) Mohawk noun-stem incorporation (Postal[117])
(4) Dutch infinitival verb phrases (Huybregts[58])
(5) assertions involving numerical expressions (Elster[28]).

Such mistaken efforts have continued:

(6) English *such* that clauses (Higginbotham[50])
(7) English "sluicing" clauses (Langendoen & Postal[97])

Both these arguments are based on false claims about what is grammatical in English (see Pullum[119]).

However, recently at least one apparently valid instance of a natural language with a weakly non-CF syntax has been found. Shieber[146] argues that the dialects of German spoken around Zurich, Switzerland, show evidence of a pattern of word order in certain subordinate infinitival clauses that is very similar to that observed in Dutch: an arbitrary number of noun phrases (NP's) may be followed by a finite verb and a specific number of nonfinite verbs, the number of NP's being a function of the lexical properties of the verbs, and the semantic relations between verbs and NP's exhibiting a crossed serial pattern: verbs further to the right in the string of verbs take as their objects NP's further to the right in the string of NP's. The crucial substrings have the form NP^mV^n. In a simple case, where $m = n = 5$, such a substring might have a meaning like

Alf watched Bob let Cal help Don make Ed work (3)

but with a word order corresponding to

Alf	Bob	Cal	Don	Ed	watched	let	help	make	work
NP_1	NP_2	NP_3	NP_4	NP_5	V_1	V_2	V_3	V_4	V_5

(4)

This construction does not render Dutch non-CF, as was shown in Pullum and Gazdar.[120] But in Swiss German, unlike Dutch, there is an additional property that makes this phenomenon relevant to stringset argumentation: certain verbs demand dative rather than accusative case on their objects, as a matter of pure syntax. This pattern will in general not be one that a CF-PSG can describe. For example, if we restrict the situation (by intersecting with an appropriate regular set) to clauses in which all accusative NP's (NP_a) precede all dative NP's (NP_d), then the grammatical clauses will be just those where the accusative-demanding verbs (V_a) precede the dative-demanding verbs (V_d) and the numbers match up; schematically:

$$NP_a{}^m NP_d{}^n V_a{}^m V_d{}^n \tag{5}$$

But this schema has the form of a language like $\{a^m b^n c^m d^n \mid n > 0\}$, which is non-CF. Shieber presents a rigorously formulated argument along similar lines to show that the language does indeed fail to be a CFL because of this construction.

It is possible that other languages will also turn out to be non-CF, though the necessary configurations of properties seem at present to be very rare. Certain properties of Swedish have given rise to suggestions in this direction, though no careful argument has been published; Carlson[17] notes a possibly non-CF reduplication construction in the syntax of Engenni, an African language, though he does not regard the case as clear; Alexis Manaster-Ramer (personal communication) suggests that the English idiomatic construction exemplified by *RS-232 or no RS-232, this terminal isn't working* (where the pattern *X or no X* is essential to acceptability, and *X* can take infinitely many values) also illustrates this possibility; and there may well prove to be properties of other languages that are worth investigating further.

[5] Indexed languages

The indexed languages (ILs, Aho[2]) are a natural class of formal languages which form a proper superset of the CFLs and a proper subset of the context-sensitive languages. The class includes some NP-complete languages (Rounds[132]). They are of interest in the present context because no phenomena are known which would lead one to believe that the NLs fell outside their purview. In particular, it is clear that indexed grammars are available for the Swiss German facts and for most other sets of facts that have been even conjectured to hold problems for CF description (but cf. Marsh & Partee[101] for a possibly harder problem, relating more to semantics than syntax).

The indexed languages thus provide us, at least for the moment, with a kind of upper bound for syntactic phenomena. We can no longer be surprised by non-CFL patterns (though their rarity is a matter of some interest), but we should be very surprised at, and duly suspicious of, putatively non-IL phenomena.

〔6〕 Beyond the indexed languages

As we have just indicated, we do not believe that any currently known facts give one reason to believe that the NLs fall outside the ILs, and in the absence of such facts, the conservative conclusion to draw is that the NLs fall within the ILs. Chomsky[21] has speculated in rather vague terms that NLs may not even be recursively enumerable sets, but this speculation amounts to a rejection of the idealisation that makes generative grammar a possible enterprise, and, as such, it is not a speculation that we can see any grounds for embracing or any point in considering.

Unlike Chomsky, Hintikka[51] has actually argued that English is not recursively enumerable, since a decision as to grammaticality for some of its sentences depends, in his view, on an undecidable question of logical equivalence. His argument (which, incidentally, Chomsky[21] rejects) is based on a controversial claim concerning the grammaticality of sentences containing the word *any*, and is closely tied to a controversial proposal for game-theoretic treatments of the semantics for natural languages. As such, the claim is highly theory-dependent and we will not consider it further here.

Langendoen and Postal[96] also argue that English is not recursively enumerable (and nor is any other natural language), on the grounds that the simplest and most general idealization of natural languages is one that allows them to have sentences of infinite length. Of this, we note simply that if it is accepted, the questions discussed above can be rephrased as questions about the finite-length-string subsets of the natural languages. It is only these subsets that are of computational interest anyway.

§3 Grammar Types and Their Parsers

The theoretical linguist's primary criterion in evaluating a type of grammar has always been its ability to capture significant generalizations within the grammar of a language and across the grammars of different languages. However, capturing significant generalizations is largely a matter of notation, and classes of grammars, taken as sets of mathematical objects, have properties which are theirs independently of the notations that might be used to define them. Thus they determine a certain set of string sets, they determine a certain set of tree sets, they stand in particular equivalence relations, and so on. Unfortunately, theoretical linguists have consistently confused grammar formalisms with grammars. This tendency reaches its apogee in the "Government Binding" framework associated with N. Chomsky and his students where the formalism employed entirely lacks a mathematical underpinning in terms of a class of admissible grammars.

Thanks to the confusion just noted, argumentation purporting to show that some class of grammars will necessarily miss significant generalizations about some NL phenomenon has been woefully inadequate. Typically it has

consisted simply of providing or alluding to some member of the class which obviously misses the generalization in question. But, clearly, nothing whatever follows from such an exhibition. Any framework capable of handling some phenomenon at all will typically make available indefinitely many ugly analyses of the phenomenon. But this fact is neither surprising nor interesting. What is surprising, and rather disturbing, is that arguments of this kind (beginning, classically, in chapter 5 of Chomsky[19]) were taken so seriously for so long.

In this section we are concerned, not with the formalisms that have been employed in recent grammatical and morphological work, but rather with the underlying formal grammars that have been assumed, and with the parsers that have been used with these grammars.

3.1 Words

Linguists use the term "morphology" to refer to that branch of their subject that deals with the internal syntax of words. The subject was much studied in the 1940's and 1950's but was then largely neglected for two decades. The following subsections briefly examine some recent developments.

[1] Finite state transducers

The idea of using finite state transducers (FSTs) to determine the mapping between syntactic and morphological/phonological structures originates in Johnson,[66] but current interest in the topic was provoked by unpublished work of Kaplan and Kay.[73] Their proposal involved the use of a cascade of two-tape FSTs to mediate between a phonemic representation of a word and a more abstract lexical representation. It is in principle possible to convert any such cascade of FSTs into a single (large) FST. Subsequent work by Koskenniemi[89] showed that a serial arrangement of FSTs could be replaced by a parallel arrangement. It is quite feasible to reduce the latter to a single FST, although implementation is also possible without any reduction. A lot of further work has been done, both of a computational character (Karttunen,[74] Gajek et al.,[39] Khan et al.[84]), and on various languages including English (Karttunen and Wittenberg[76]), Japanese (Alam[4]), Rumanian (Khan[83]), French (Lun[99]), and Finnish (Koskenniemi[88]). The basic Koskenniemi two-tape model can handle infixation and (finite) reduplication but not, it seems, in an elegant or perspicuous manner. The most recent work by Kay[80] has explored the use of n-tape FSTs (for n greater than 2) in order to handle such phenomena as the vowel harmony and discontinuous roots found in Semitic languages.

[2] Context-free phrase structure grammars

The classical structuralist model of morphology, dubbed "Item and Arrangement" by Hockett,[54] which was prevalent in the 1940's and 1950's, was essentially a CF-PSG model although, of course, it predated the mathematical theory of CF-PSGs. This model of lexical structure was radically inconsistent

with the transformationalist view of sentence syntax that became dominant in the 1960's. The latter claimed, in effect, that there was no distinction to be made between the syntax of words and the syntax of sentences: they were to be handled with the same machinery and that machinery was not CF-PSG.

But recent influential work has seen a return to an Item and Arrangement position, though not *eo nomine*, most notably in that of Selkirk[143] who argues that "English word structure can be properly characterized solely in terms of a context-free grammar". In fact, Selkirk then goes on to employ context-sensitive rules to handle the subcategorization requirements of affixes although there is no need for her to do so. Interestingly from our perspective, Selkirk appears to regard the CF-PSG hypothesis that she espouses as the most conservative hypothesis that could be espoused. She never considers the possibility of using finite state machinery, and yet none of the phenomena she deals with show any trace of strict context-freeness when viewed language-theoretically. It may be true, however, that the structure of words cannot be adequately handled in terms of finite-state grammars; Carden[16] briefly argues that this is so. We discuss Selkirk's work further in Section 3.3〔1〕, below.

〔3〕 Context-sensitive phrase structure grammars

In an influential 1979 thesis on Semitic word structure, McCarthy[104] claimed that "morphological rules must be context-sensitive rewrite rules, and no richer rule type is permitted in the morphology" (p.201). Like Selkirk, McCarthy is really reacting here to the totally unconstrained views of morphology that linguists had previously found acceptable. He points out that Chomsky's[18] morphological transformations could "perform their arbitrary operations on only the prime or factor-of-twelve numbered segments in the word with no further enrichment of the formalism" (p.201). Seen in that context, his proposal is a restrictive one, but seen in the language-theoretic context assumed here, his proposal is, of course, radically unconstrained. None of the phenomena that he deals with involve even strict context-freeness, much less strict context-sensitivity. Indeed, as noted above, Kay[80] has been able to develop finite state analyses of McCarthy's data using multi-tape transducers.

3.2 Sentences

In the following subsections we look at the application of formal grammars in recent work on the syntax of sentences by linguists and computational linguists.

〔1〕 Finite state grammars

The fact that natural languages are not regular does not necessarily mean that techniques for parsing regular languages are irrelevant to natural language parsing. Such writers as Langendoen,[92] Church,[22] Ejerhed and Church,[27] and Langendoen and Langsam[95] have, in rather different ways, proposed that

hearers process sentences as if they were finite automata (or as if they were pushdown automata with a finite stack depth limit, which is weakly equivalent) rather than showing the behavior that would be characteristic of a more powerful device. To the extent that progress along these lines casts light on NL parsing, the theory of regular grammars and finite automata will continue to be important in the study of natural languages even though they are not regular sets.

[2] Categorial grammars

Categorial grammars, which were developed by Bar-Hillel and others in the 1950's, have always had a somewhat marginal status in linguistics. There has always been someone ready to champion them, but never enough people actually using them to turn them into a paradigm. The currency they have today is due in large measure to Montague[105] who based his semantic work on a modified categorial grammar.

The elegance and unprecedented explicitness of Montague's grammars provoked a good deal of work in computational linguistics, for example, that of Bronnenberg et al.,[14] Friedman,[34,35] Friedman, Moran & Warren,[36] Friedman & Warren,[37] Fuchi,[38] Hobbs & Rosenschein,[53] Indurkhya,[60] Ishimoto,[61] Janssen,[62,63,65] Landsbergen,[90,91] Matsumoto,[102,103] Moran,[106] Nishida et al.,[110] Nishida & Doshita,[108,109] Root,[127] Saheki,[136] Sawamura,[139] Sondheimer & Gunji,[150] Warren,[164] and Warren and Friedman.[165]

Montague's own generalizations of categorial grammar were not exactly principled and most of the work just cited is more concerned with semantic issues than it is with the niceties of the underlying syntactic theory. Pure categorial grammar is really a variant of CF-PSG and has exactly the same weak generative capacity. Recently some fairly principled attempts have been made, notably by Ades and Steedman[1] and Bach,[5] to preserve the spirit of categorial grammar (which Montague, arguably, did not) whilst extending it to non-CF constructions such as that found in Swiss German (cf. Steedman[151] on the analogous Dutch construction).

[3] Context-free phrase structure grammars

Since 1978, following suggestions by Stanley Peters, Aravind Joshi, and others, there has been a strong resurgence in the linguistics literature of the idea that phrase structure grammars could be used for the description of natural languages. PSGs had been all but abandoned in linguistics during the period from 1957 to 1978 because arguments given by the proponents of transformational grammar had convinced essentially all linguists interested in writing formal grammars that no phrase structure account of the grammar of a natural language could be adequate.

One of the motivations suggested for continuing to take an interest in PSGs, in particular CF-PSGs, was the existence of already known high-

efficiency algorithms (recognition in deterministic time proportional to the cube of the string length) for recognizing and parsing CFLs. Indeed, as Perrault[113] reminds us, "it is useful to remember that no known CFL requires more than linear time, nor is there even a nonconstructive proof of the existence of such a language". Context-free parsing is such a basic tool of computer science, including NLP, that there have even been proposals for implementing CF-PSG parsers in special-purpose NLP hardware (Dubinsky and Sanamrad[26]; Schnelle[140]).

But parsability has not been the central motivation for the interest that significant numbers of linguists began to show in CF-PSGs from early 1979. Linguists were mainly interested in achieving elegant solutions to purely linguistic problems, and work by linguists such as Borsley,[12] Cann,[15] Flickinger,[32] Gunji,[42-45] Horrocks,[56,57] Ikeya,[59] Kameshima,[71] Maling and Zaenen,[100] Nerbonne,[107] Sag,[134,135] Saito,[137] Stucky,[153,155] Udo,[161] Uszkoreit,[162] and Zwicky[168] is directed toward this end.

The idea of returning to CF-PSG as a theory of NLs may have appeared highly retrogressive to some linguists in 1979; but, as we have seen, the published arguments that had led linguists to consign CF-PSGs to the scrap-heap of history were all quite unsatisfactory. In view of that, the development of theories of the structure of English and other languages in terms that guaranteed context-freeness of the analyzed language became eminently sensible.

The CF-PSGs enjoy a wealth of literature providing them with numerous distinct but equivalent mathematical characterizations which illuminate their many computationally relevant properties. They are relatively simple to write and to modify. They are associated with a successful tradition of work in computation that has provided us with a thorough understanding of how to parse, translate, and compile them (Aho and Ullman[3]). Much work was done in the period 1979-1984 to establish a basis for handling the syntax and semantics of NLs as effectively and precisely as the structures of programming languages or the artificial languages of logicians.

What attitude should NLP research and development work take toward the pieces of evidence that indicate that NLs are not all CFLs? We believe the fundamental thing that should be kept in mind is this: *The overwhelming majority of the structure of any NL can be elegantly and efficiently parsed using context-free parsing techniques.* That is, we think it is essential to keep a sense of proportion. Too often the most sweeping conclusions about the uselessness of context-free parsing for NLs have been made even on the basis of transparently fallacious arguments. The truth is that nearly all constructions in nearly all languages can be parsed using techniques that limit the system to the analysis of CFLs. It has taken linguists nearly thirty years (since 1956, when Chomsky raised the question of whether NLs were CFLs and whether CF-PSGs could be used to describe them) to correctly identify even one construction in a NL that lends non-CFL status to the whole language.

Unsurprisingly, the new linguistic work on CF-PSG has been ac-

companied by a whole genre of parallel work in computational linguistics by such researchers as Bear & Karttunen,[8] Evans & Gazdar,[31] Evans,[30] Gunji, et al.,[47] Hirakawa,[52] Joshi,[68] Joshi & Levy,[70] Karttunen,[75] Kay,[78] Keller,[81,82] Kilbury,[85] Konolige,[87] Phillips & Thompson,[114] Pulman,[121,122] Robinson,[125,126] Rosenschein & Shieber,[129] Ross,[130,131] Sampson,[138] Schubert,[141] Schubert & Pelletier,[142] Shieber,[144,145] Shirai,[148] Thompson,[156-158] Thompson & Phillips,[159] and Uszkoreit.[163] CF-PSGs have been used as the syntactic basis for sophisticated NL front-ends to databases (see the work reported initially by Gawron et al.,[40] and subsequently developed as outlined by Pollard & Creary,[116] Flickinger, Pollard & Wasow,[33] and Proudian & Pollard[118]) and highly effective machine translation systems (Slocum et al.[149]).

[4] Head grammars

Pollard[115] presents several generalizations of context-free grammar, the most restrictive of which he refers to as head grammar (HG). The extension Pollard makes in CF-PSG to obtain the HGs is in essence fairly simple. First, he treats the notion "head" as a primitive. The strings of terminals his syntactic rules define are headed strings, which means they are associated with an indication of a designated element to be known as the head. Second, he adds "wrapping" operations to the standard concatenation operation on strings that a CF-PSG can define. This permits a limited amount of interleaving of sister constituents, as opposed to the straightforward concatenation of sisters to which CF-PSG is restricted; a string x can be combined with a string yhz, where h is the head, not only to yield $xyhz$ or $yhzx$ but also by an operation that yields $yhxz$ or $yxhz$. The intuitive element of context-freeness that a grammar of this sort retains lies in the fact that constituents are defined independently of other constituents: where two headed strings are to be combined to form a new string A, no context outside of A can be relevant to the operation.

Using just concatenation and head wrapping, Pollard[115] shows how an analysis of the special subordinate verb phrase constructions of Dutch or Swiss German can readily be obtained. The discontinuities between syntactically associated constituents that is made available by head wrapping is just enough to associate the right verbs with the right noun phrases in Dutch or Swiss German subordinate VP's, without introducing the whole power of arbitrary context-sensitive grammars.

The HG framework is not just another notation for highly powerful arbitrarily augmented phrase structure grammars, and does not introduce exponential levels of difficulty into the recognition or parsing problems. HGs have a greater expressive power, in terms of weak and strong generative capacity, than the CF-PSGs, but only to a very limited extent. Pollard shows that an arbitrary HG language can be recognized, by means of a modified version of the CKY algorithm, in deterministic time proportional to the seventh power of the length of the string. Though worse than the worst-case result for CFLs, this is

not a result that indicates intractability of the recognition problem for HG languages.

Roach[124] has proved that the languages generated by head grammars constitute a full abstract family of languages, showing all the significant closure properties that characterize the class of CFLs, and has observed a striking similarity to the properties found in the tree-adjunction grammars (TAGs) studied by Joshi and others (cf. Joshi[69]), which are defined in an intuitively very different way and were conceived quite independently. TAGs and HGs both offer the prospect of efficient recognition for what Joshi calls "mildly context-sensitive languages", and the convergence between these two lines of research is very encouraging.

(5) Indexed grammars

If nonterminal symbols are built up using sequences of indices affixed to a members of a finite set of basic nonterminals, and rules are able to add or remove sequence-initial indices, then the expressive power achieved is that of the indexed grammars of Aho[2] and Hopcroft & Ullman.[55] Indexed grammars are similar to CF-PSGs which employ complex symbols, except that there is no finite limit on the number of distinct complex symbols that can be used. The indexed grammars have an automata-theoretic characterization in terms of a stack automaton that can build stacks inside other stacks but can only empty a stack after all the stacks within it have been emptied. The time complexity of the recognition problem is exponential.

No theoretical linguists have yet embraced indexed grammars, although it is clear that generalization of category valued features to allow n-tuples of categories as feature values (as envisaged by Maling & Zaenen,[100] and subsequently Pollard[115]) leads one directly to the indexed grammars unless the system is otherwise constrained. It is also clear that indexed grammars of a rather straightforward kind are available for the Swiss German/Dutch construction discussed in preceding sections, for reduplications, and for multiple wh-type dependencies in Scandinavian languages and the variable-binding issue that this gives rise to (see Engdahl[29] and Maling & Zaenen[100]). They can also handle the nesting of equative and comparative clauses discussed by Klein.[86] Thus, as noted in Section 2.2(5), above, indexed grammars provide us with an upper bound. There are no grammatical phenomena that we know of that they cannot handle, but the same is true, of course, of Turing machines.

(6) Beyond the indexed grammars

If nonterminal symbols have internal hierarchical structure and parsing operations are permitted to match hierarchical representations one with another globally to determine whether they unify (see Section 4.3, below), and if the number of parses for a given sentence is kept to a finite number by requiring that we do not have $A \Rightarrow^{+} A$ for any A, then the expressive power seems to be

weakly equivalent to the grammars that Bresnan and Kaplan have developed under the name lexical-functional grammar (LFG; see Bresnan, ed.[13]). The LFG languages include some non-indexed languages (Roach, unpublished work), and apparently have an NP-complete parsing problem.[9]

The LFG languages may not even be included in the context-sensitive languages. Kaplan and Bresnan[72] (260ff) state without proof that any language with an LFG grammar is accepted by some nondeterministic linear bounded automaton, provided each grammar observes a fixed numerical upper bound on the number of crossing dependencies permitted (a "crossing limit"). If limits of this sort are embraced, comparisons of generative power across frameworks are of course undercut; the reduplication of noun stems in Bambara could be handled by even a finite state grammar if, for example, grammars observed a restriction to a finite upper bound on the number of times a given noun stem could occur in a sentence. Likewise, a CF-PSG could be given for the cross-serial verb-object dependencies in Dutch and Swiss German, assigning suitable constituent structure, provided a crossing limit was imposed. Thus it is unclear whether Kaplan and Bresnan see the imposition of crossing limits as a motivated part of their theory or simply as a sufficient condition to achieve context-sensitivity. Note that further numerical conditions would allow a proof of CF-ness or even finite-state-ness for LFGs.

3.3 The Word-Sentence Interface

In this section we consider the relation that obtains between the syntax of words and the syntax of sentences. At least three distinct positions can be distinguished and have been maintained at one time or another. One possible position says that there is no distinction to be made and that a single grammar unifies both. We will call this the holistic position. This was essentially the position maintained by transformational grammar in the 1960's and early 1970's. A second position, which we refer to as "the orthodox view" below, maintains that they are distinct autonomous systems, but that the syntax of words is properly embedded within the syntax of sentences. Where the latter ends, the former begins. And a third position, recently dubbed "autolexical syntax", claims that they are distinct but parallel systems that need not define compatible analyses of the entire morpheme string.

(1) The orthodox view

The orthodox view, as we have characterised it above, is essentially that of the American structuralist tradition of the 1940's and 1950's. It is classically embodied in the analytical technique known as immediate constituent analysis. The immediate constituents of sentences were phrases, the immediate constituents of phrases were words (or other phrases), and the immediate constituents of words were morphemes. Thus the syntax of words was properly nested within the syntax of phrases and sentences.

This position finds its modern expression in the work of Selkirk.[143] For her, sentential syntactic structures are phrase structure trees and the leaves of these trees are words. But these words each have their own self-contained structure and this too is represented as a phrase structure tree.

It is easy to see that Selkirk's position is only separated from the holistic position by a very thin line. If the sentence syntax is simply a PSG (though it is not, for Selkirk) and the word syntax is also, and if the latter is permitted to introduce nonterminal symbols that belong conceptually to the former (as various incorporation phenomena might lead one to want), then one simply ends up with one large PSG whose terminal symbols are morphemes.

〔2〕 Autolexical syntax

In a radical break with the traditions just discussed, Sadock[133] has proposed that the sentence syntax and the word syntax are both CF-PSGs, but that strings of morphemes are to be regarded as grammatical just in case they receive both a sentence-syntactic structure and a word-syntactic structure. Crucially, these structures need not be isomorphic: words as defined by the morphology are not required to coincide with some syntactic constituent as they are in the orthodox view. Thus, for example, in English, the morpheme sequence *I'll* in *I'll go to bed* is a constituent in Sadock's word-syntax, but not, of course, in his sentential syntax.

Oversimplifying somewhat, Sadock's work suggests a parsing model in which two CF-PSG parsers, one corresponding to the sentence syntax and the other to the word syntax, run in series or in parallel, a string being grammatical just in case both parsers succeed. Such a dual CF-PSG parsing system would identify languages falling in the intersection of two CFLs. As is well known, the CFLs are not closed under intersection and thus, for example, such a system could recognize $a^n b^n c^n$ (cf. the cascaded RTNs of Woods[167]). However, the recognition time complexity is no worse than that of the CFLs, i. e. $O(n^3)$. Borgida[11] provides an excellent introduction to a range of dual grammar systems.

§4 Recent Developments in Formal Linguistics

4.1 Grammars for Grammars

The idea of using one grammar to generate another originates in computer science with the work of van Wijngaarden[166] who used the technique to give a perspicuous syntax for ALGOL68. A good introduction to his work can be found in Cleaveland & Uzgalis.[23] Janssen[64] employs a van Wijngaarden-style two-level grammar to define a generalization of Montague's PTQ syntax.

The same idea emerges in recent linguistic work in the guise of the "lexical rule"[13] or metarule.[41] A metarule is a grammar characterization device (i. e. a clause in the definition of the grammar), one which enables one to define

one set of rules in terms of another set, antecedently given. Generalizations which would be lost if the two sets of rules were merely listed are captured by the metarule.

For example, suppose that our grammar contains, inter alia, the following set of rules expanding VP:

$$\left.\begin{array}{l} \text{VP} \rightarrow \text{V}[0] \\ \text{VP} \rightarrow \text{V}[1]\ \text{NP} \\ \text{VP} \rightarrow \text{V}[2]\ \text{NP NP} \\ \text{VP} \rightarrow \text{V}[3]\ \text{NP PP} \\ \text{VP} \rightarrow \text{V}[4]\ \text{NP VP} \\ \text{VP} \rightarrow \text{V}[5]\ \text{NP S} \\ \text{VP} \rightarrow \text{V}[6]\ \text{NP NP S} \\ \text{VP} \rightarrow \text{V}[7]\ \text{S} \end{array}\right\} \quad (6)$$

Then we can augment the grammar by means of the following metarule:

$$\left.\begin{array}{l} \text{VP} \rightarrow \text{V NP } W \quad \Longrightarrow \\ \text{VP[PAS]} \rightarrow \text{V } W \text{ (PP[by])} \end{array}\right\} \quad (7)$$

This says that for every rule in the grammar which expands VP as a verb followed by an NP possibly followed by arbitrary other material, there is also a rule expanding a passive VP as the verb followed by the other stuff (if there was any) followed optionally by an agentive PP. This metarule will thus add the following rules to our grammar:

$$\left.\begin{array}{l} \text{VP[PAS]} \rightarrow \text{V[1] (PP[by])} \\ \text{VP[PAS]} \rightarrow \text{V[2] NP (PP[by])} \\ \text{VP[PAS]} \rightarrow \text{V[3] PP (PP[by])} \\ \text{VP[PAS]} \rightarrow \text{V[4] VP (PP[by])} \\ \text{VP[PAS]} \rightarrow \text{V[5] S (PP[by])} \\ \text{VP[PAS]} \rightarrow \text{V[6] NP S (PP[by])} \end{array}\right\} \quad (8)$$

These rules will now allow the grammar to generate passive sentences directly. Another example is provided by Gunji,[45] who shows how metarules can capture reflexive pronoun generalizations in the definition of a CF-PSG for Japanese.

Recent work in computational linguistics that employs or explores the notion of metarule includes Gawron et al.,[40] Kay,[78] Konolige,[87] Robinson,[125] Schubert and Pelletier,[142] Shieber, Stucky, Uszkoreit and Robinson,[147] Stucky,[154] and Thompson.[157]

4.2 Feature-Theoretic Syntax

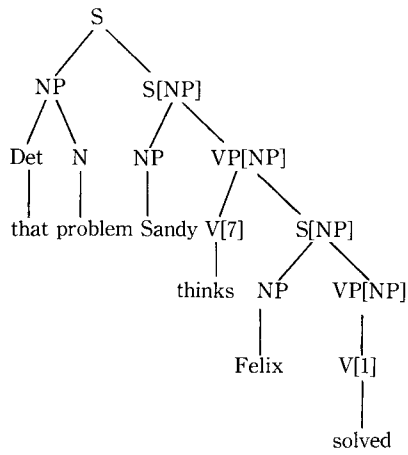
Harman[49] was the first person to see the linguistic potential of PSGs incorporating complex symbols. The use of a finite set of complex symbols, in place of the traditional finite set of monadic symbols, leaves the mathematical properties of grammars unchanged. For example, every CF-PSG employing

complex symbols generates a tree set that is isomorphic to the tree set generated by some CF-PSG not employing complex symbols.

Typically, syntactic categories are defined as sets of syntactic feature specifications. A feature specification is an ordered pair consisting of a feature (e. g. CASE) and a feature value. The latter may either be atomic (e. g. ACCUSATIVE) or it may be a syntactic category (i. e. features are allowed to take categories as their values). A syntactic category is then a partial function from features to their values. The internal make-up of categories is further constrained by feature cooccurrence restrictions which are simply Boolean conditions which restrict the permissible combinations of feature specifications.[41] Syntactic structures are thus phrase structure trees of the familiar kind whose nodes are labelled with syntactic categories as characterized above.

Principles of feature instantiation are then invoked to ensure the identity of certain features on adjacent or connected nodes. Most current work assumes a "Head Feature Convention" which is responsible for equating one class of feature specifications as they appear on the mother category and its head daughter(s). Thus, for example, a verb phrase inherits the tense of its verb. Other principles match agreement features between locally connected agreeing categories (e. g. between a subject noun phrase and its verb phrase sister), or deal with the copying of category valued features between mother and daughter categories.

Category-valued features allow many significant syntactic generalizations to be captured rather straightforwardly.[41] For example, they are able to capture those underlying the class of unbounded dependency constructions (e. g. relative clauses, wh-questions, topicalization, etc.). Here is a topicalization example, where the category-valued feature specification [NP] encodes the absence of the object in the final verb phrase.



(9)

4.3 Unification, Extension, and Generalization

The formal definitions of principles of feature instantiation, such as those mentioned above, crucially depend upon notions of extension and unification definable in a graph-theoretic or partial function theory of categories. These notions were introduced into linguistics by Kay[77] and have been profoundly influential, finding their way into essentially all current formal syntactic frameworks.

Assuming, for the sake of illustration, the partial function theory of categories sketched above, we can define extension as follows.

> A category C2 **is an extension of** a category C1 if and only if
> (1) every atom-valued feature specification in C1 is in C2, and
> (2) for every category-valued feature specification in C1, the value of the feature in C2 is an extension of the value in C1.

This recursive definition says first of all that any specification for an atom-valued (i. e. non-category valued) feature in a category is also in all extensions of that category. It also guarantees that if a category specifies a value v for some category-valued feature, then any extension of that category specifies a value for that same feature that is an extension of v. Note that an extension of a category C may contain a specification for a category-valued feature which is unspecified in C. The relation "is an extension of" is thus a generalization of the relation "is a superset of", one which takes proper account of category-valued features, and it defines a partial order on the set of categories.

An important operation on categories is that of unification. This notion is closely analogous to the operation of union on sets except that, as in the case of extension, the resulting set must be a function. Unification is undefined for categories containing features specifications that contradict each other.

> The **unification** of a set of categories K is the smallest category which is an extension of every member of K, if such a category exists; otherwise the unification of K is undefined.

As can be seen, this notion is equivalent to the standard notion of least upper bound in lattice theory. A second operation on categories is generalization, which provides the analogy to the operation of intersection on sets. It can be defined as follows.

> The **generalization** of a set of categories K is the smallest category which contains (1) the intersection of the categories in K, and (2) the set of category-valued feature specifications each of whose values is the generalization of the set of values assigned to the feature by the categories in K.

Karttunen[75] provides a good introduction to the linguistic uses of generaliza-

tion, and Pereira & Shieber[112] discuss the denotational semantics of unification-based linguistic formalisms.

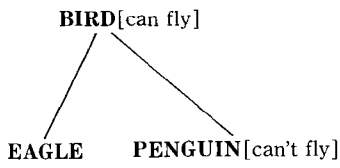
In discussing these notions, we have restricted ourselves to the theory of features for the purposes of illustration. However, some current linguistic frameworks, notably Kay's[79] "Functional Unification Grammar", allow one to perform unification on structural descriptions, and even on grammars.

4.4 Default Mechanisms

In any feature-theoretic linguistic framework, certain feature values are the expected case, the values that ordinarily get assigned, other things being equal. Linguists call these expected values the "unmarked" or default values, and they can be handled by feature specification defaults which are Boolean conditions analogous to feature cooccurrence restrictions, but employed differently. Feature cooccurrence restrictions are absolute conditions that have to be met, whereas feature specification defaults are conditions that a category must meet if it can, but need not meet if it cannot.[41] Thus, for example, the default value for CASE might be ACCUSATIVE, but a given noun phrase could appear in some other case if it was required to do so by a feature instantiation principle, say.

Feature instantiation principles themselves have typically imposed an absolute condition (identity or extension in one direction) on the relation between the feature sets found on adjacent or connected nodes in a tree. But this clearly does not need to be the case. An absolute condition could be replaced by a default inheritance mechanism, a technical device of considerable generality, and potentially wide application.

One, nonlinguistic, example of a default inheritance mechanism comes from the work on semantic networks in AI (see Barr and Feigenbaum[7]: pp. 180-189 for a survey). On the one hand one wishes to be able to say that all birds can fly (thus avoiding the need to stipulate that eagles can fly), but on the other hand one wishes to accord penguins the status of being a bird (even though they cannot fly). To take an example that is familiar from the AI literature, consider the network shown below.

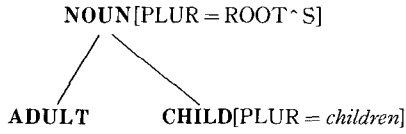


(10)

The arcs here represent the **ISA** relation, and the material in brackets stands for properties, so this network just records that eagles and penguins are birds, that birds can fly, and that penguins cannot. If properties are simply inherited from a dominating node, then we will derive a contradiction to the effect that

penguins are both able to fly and unable to fly. But if properties are inherited by a default inheritance mechanism, then we will be able to derive the flight of eagles without contradicting ourselves over penguins.

One potential linguistic application for a default inheritance mechanism concerns irregularity in NL lexicons. On the one hand we wish to give general morphological rules to predict the form of, say, plurals. And, on the other hand, we want these rules to be over-ridden by the mere existence of irregular forms.



(11)

This says that **adult** and **child** are both nouns and that the plural form (PLUR) of a noun is formed by concatenating its stem (ROOT) with **s**. This property of nouns will be inherited by **adult** which thus has **adults** as its plural. In the case of **child** however, this property of nouns is not inherited since it is inconsistent with an existing property of **child**, namely the property of having **children** as its plural. See Flickinger, Pollard & Wasow[33] for details of a lexicon that uses default inheritance to handle examples of just this kind.

Another application can be found in Gazdar et al.[41] who use a default inheritance mechanism in stating the "Head Feature Convention" mentioned in passing in Section 4.2, above. Identity for a given feature is only imposed when it is possible given the other constraints that apply to the features and categories involved. Essentially their definition works by examining the space of possible instantiations of a rule that are permitted by feature cooccurrence restrictions, other feature principles, and so on. If this space contains an instantiation exhibiting the relevant identity, then the principle requires identity; if it does not, then identity is not required.

§5 Conclusion

The arguments originally given at the start of the era of modern linguistics were correct in their conclusion that NLs cannot be treated as simply regular sets of strings, as some early information-theoretic models of language users would have had it. However, questions of whether NLs were CFLs were dismissed rather too hastily; English has never been shown to be outside the class of CFLs or even the DCFLs, and even for other languages the first apparently valid arguments for non-CFL status are only now being framed.

The two non-CF construction types that have been shown to exist in NLs are not indicative of profound difficulties standing in the way of the efficient processing of NLs by computers. They are well understood in linguistic terms, and efficient techniques for recognizing and parsing languages with these constructions are already known to exist, as we have noted. Thus, although

future work on NLP may face major difficulties in areas like speech, semantics, and pragmatics, there is reason to think that difficulties in the area of morphology and syntax have often been exaggerated. The vast majority of the regularities of structure in the words and sentences of NLs can be captured in terms of the tractable and mathematically developed framework of CF-PSG. And in the cases where supra-CF devices are called for, there are numerous promising extensions or generalizations of CF-PSG that are clearly capable of doing the job and are already being explored by linguists and computational linguists.

The traditional attitude toward natural languages in computer science has probably not been very different from the traditional attitude among logicians. Rosenbloom[128] (p.153), for example, asserts:

> As in all natural languages...the rules of word and sentence formation in English are so complicated and full of irregularities and exceptions that it is almost impossible to get a general view of the structure of the language, and to make generally valid statements about the language.

Modern work on morphology and syntax does not bear out this pessimistic view. On the contrary, our conclusion from this review of morphological and syntactic work on the computationally relevant properties of NLs and their grammars is, in short, that a cautious optimism is in order.

References

1) Ades, Anthony E. and Steedman, Mark J., "On the order of words," *Linguistics and Philosophy, 4*, pp. 517-558, 1982.
2) Aho, Alfred V., "Indexed grammars — an extension of context-free grammars," *Journal of the ACM, 15*, pp. 647-671, 1968.
3) Aho, Alfred V. and Ullman, Jeffrey D., *The Theory of Parsing, Translation, and Compiling*, Prentice-Hall, Englewood Cliffs, New Jersey, 1972.
4) Alam, Yukiko Sasaki, "A two-level morphological analysis of Japanese," in *Texas Linguistic Forum, 22* (Mary Dalrymple et al., eds.), University of Texas, Austin, Texas, pp. 229-252, 1983.
5) Bach, Emmon, "Discontinuous constituents in generalized categorial grammars," in *Proceedings of the 11th Annual Meeting of the North Eastern Linguistic Society*, (V. A. Burke and J. Pustejovsky, eds.), Department of Linguistics, Unversity of Massachusetts at Amherst, Amherst, Massachusetts, pp. 1-12, 1981.
6) Bar-Hillel, Yehoshua and Shamir, E., "Finite state languages: formal representations and adequacy problems," 1960. As reprinted in *Language and Information* (Yehoshua Bar-Hillel, ed.), Addison-Wesley, Reading, Massachusetts, pp. 87-98, 1964.
7) Barr, Avron and Feigenbaum, Edward (eds.), *The Handbook of Artificial Intelligence, Volume 1*, William Kaufman, Los Altos, California, 1981.
8) Bear, John and Karttunen, Lauri, "PSG: a simple phrase structure parser," *Texas Linguistic Forum, 15*, pp. 1-46, 1979.
9) Berwick, Robert C., "Computational complexity and lexical-functional grammar," *American Journal of Computational Linguistics, 8, 3-4*, pp. 97-109, 1982.

10) Bermudez, Manuel, "Regular Lookahead and Lookback in LR Parsers," *PhD thesis*, University of California, Santa Cruz, California, 1984.
11) Borgida, Alexander T., "Some formal results about stratificational grammars and their relevance to linguistics," *Mathematical Systems Theory, 16*, pp. 29-56, 1983.
12) Borsley, Robert, "On the nonexistence of VP's," in *Sentential Complementation*, (Willem de Geest and Yvan Putseys, eds.), Foris, Dordrecht, pp. 55-65, 1984.
13) Bresnan, Joan W. (ed.), *The Mental Representation of Grammatical Relations*, MIT Press, Cambridge, Massachusetts, 1982.
14) Bronnenberg, W. J. H. J., Bunt, H. C., Landsbergen, S. P. J., Scha, R. J. H., Schoenmakers, W. J. and van Utteren, E. P. C., "The question-answering system PHLIQA 1," in *Natural Language Question-Answering Systems* (L. Bolc, ed.), Carl Hanser Verlag, Munich, West Germany, pp. 217-305, 1980.
15) Cann, Ronald, "An approach to the Latin accusative and infinitive," in *Order, Concord and Constituency* (Gerald Gazdar, Ewan H. Klein and Geoffrey K. Pullum, eds.), Foris, Dordrecht, pp. 113-137, 1983.
16) Carden, Guy, "The non-finite-state-ness of the word formation component," *Linguistic Inquiry, 14*, pp. 537-541, 1983.
17) Carlson, Greg, "Marking constituents," in *Auxiliaries and Related Puzzles, Vol. 1.* (Frank Heny, ed.), D. Reidel, Dordrecht, Holland, pp. 69-98, 1983.
18) Chomsky, Noam, "Morphophonemics of Modern Hebrew," *MA thesis*, University of Pennsylvania, Philadelphia, Pennsylvania, 1951.
19) Chomsky, Noam, *Syntactic Structures*, Mouton, The Hague, Holland, 1957.
20) Chomsky, Noam, "Formal properties of grammars," in *Handbook of Mathematical Psychology, Volume II* (R. D. Luce, R. R. Bush and E. Galanter, eds.), Wiley, New York, pp. 323-418, 1963.
21) Chomsky, Noam, *Rules and Representations*, Blackwell, Oxford, England, 1980.
22) Church, Kenneth, "On Memory Limitations in Natural Language Processing," *M. Sc. thesis*, MIT, Cambridge, Massachusetts, 1980.
23) Cleaveland, J. and Uzgalis, R., *Grammars for programming languages: what every programmer should know about grammar*, Elsevier, New York, New York, 1975.
24) Culy, Christopher, "The complexity of the vocabulary of Bambara," to appear in *Linguistics and Philosophy*, *8*,1985.
25) Daly, R. T., *Applications of the Mathematical Theory of Linguistics*, Mouton, The Hague, Holland, 1974.
26) Dubinsky, Stanley and Sanamrad, Mohammad Ali, "A universal natural language processor suitable for the hardware realization of phrase structure grammars," unpublished paper, Kobe University, 1984.
27) Ejerhed, Eva and Church, Kenneth, "Recursion-free context-free grammar," paper presented at the *Workshop on Scandinavian Syntax and Theory of Grammar*, University of Trondheim, June 3-5, 1982.
28) Elster, J., *Logic and Society: Contradictions and Possible Worlds*, Wiley, New York, New York, 1978.
29) Engdahl, Elisabet, "The syntax and semantics of questions in Swedish," *PhD dissertation*, University of Massachusetts at Amherst, Amherst, Massachusetts, 1980.
30) Evans, Roger, "ProGram — A development tool for GPSG grammars," to appear in *Linguistics*, *23*,1985.
31) Evans, Roger and Gazdar, Gerald, "The ProGram Manual," *Cognitive Science Research Paper, 35 (CSRP 035)*, University of Sussex, Brighton, England, 1984.
32) Flickinger, Daniel, "Lexical heads and phrasal gaps," in *Proceedings of the Second West Coast Conference on Formal Linguistics* (Michael Barlow, Daniel Flickinger and Michael Wescoat, eds.), Stanford Linguistics Department, Stanford, pp. 89-101,

1983.

33) Flickinger, Daniel, Pollard, Carl and Wasow, Thomas, "Structure-sharing in lexical representation," *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Morristown, NJ, pp. 262-267, 1985.

34) Friedman, Joyce, "Computational and theoretical studies in Montague Grammar at the University of Michigan," *SISTM Quarterly, 1*, pp. 62-66, 1978.

35) Friedman, Joyce, "Expressing logical formulas in natural language," in *Formal Methods in the Study of Language* (Jeroen A. G. Groenendijk, Theo Janssen and Martin Stokhof, eds.), Mathematical Centre Tracts, Amsterdam, pp. 113-130, 1981.

36) Friedman, Joyce, Moran, D. and Warren, D., "An interpretation system for Montague Grammar," *American Journal of Computational Linguistics, microfiche, 74*, pp. 23-96, 1978.

37) Friedman, Joyce and Warren, David, "A parsing method for Montague Grammars," *Linguistics and Philosophy, 2*, pp. 347-372, 1978.

38) Fuchi, Kazuhiro, "Natural language and its formal representation: a case study of translation in Montague style from a programmer's point of view," paper presented to the *First Colloquium on Montague Grammar and Related Topics*, Kyoto, February, 1981.

39) Gajek, Oliver, Beck, Hanno T., Elder, Diane and Whittemore, Greg, "KIMMO Lisp implementation," in *Texas Linguistic Forum, 22* (Mary Dalrymple et al., eds.), University of Texas, Austin, Texas, pp. 187-202, 1983.

40) Gawron, Jean Mark, King, Jonathan, Lamping, John, Loebner, Egon, Paulson, Anne, Pullum, Geoffrey K., Sag, Ivan A. and Wasow, Thomas, "The GPSG linguistics system," in *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Menlo Park, California, pp. 74-81, 1982. Also distributed as *Hewlett Packard Computer Science Technical Note, CSL-82-5.*

41) Gazdar, Gerald, Klein, Ewan, Pullum, Geoffrey K. and Sag, Ivan A., *Generalized Phrase Structure Grammar*, Blackwell, Oxford, and Harvard University Press, Cambridge, Ma., 1985.

42) Gunji, Takao, "A phrase structural analysis of the Japanese language," *MA dissertation*, Ohio State University, 1981.

43) Gunji, Takao, "Apparent object control of reflexives in a restrictive theory of grammar," *Papers in Japanese Linguistics, 8*, pp. 63-78, 1982.

44) Gunji, Takao, "Control of gaps and reflexives in Japanese," in *Proceedings of the Second Japanese-Korean Joint Workshop on Formal Grammar*, Logico-Linguistic Society of Japan, pp. 151-186, 1983. [in Japanese]

45) Gunji, Takao, "Generalized phrase structure grammar and Japanese reflexivization," *Linguistics and Philosophy, 6*, pp. 115-156, 1983.

46) Gunji, Takao, *Introduction to Linguistics for Computer Scientists*, Information Technology Promotion Agency, Tokyo, 1983. [in Japanese]

47) Gunji, Takao, et al., "Some aspects of generalized phrase structure grammar," *ICOT Technical Memo, TM-0103*, Institute for New Generation Computer Technology, Tokyo, 1985.

48) Hagège, Claude, "Relative clause center-embedding and comprehensibility," *Linguistic Inquiry, 7*, pp. 198-201, 1976.

49) Harman, Gilbert, "Generative grammars without transformation rules: a defense of phrase structure," *Language, 39*, pp. 597-616, 1963.

50) Higginbotham, James, "English is not a context-free language," *Linguistic Inquiry, 15*, pp. 225-234, 1984.

51) Hintikka, Jaakko, "On the limitations of generative grammar," in *Proceedings of the Scandinavian Seminar on Philosophy of Language, Vol. 1* (*Filosofiska Studier, Vol. 26*), Philosophical Society and Department of Philosophy, Uppsala University, Uppsala, Sweden, pp. 1-92, 1975.

52) Hirakawa, Hideki, "Chart parsing in Concurrent Prolog," *ICOT Technical Report, TR-008*, Institute for New Generation Computer Technology, Tokyo, 1983.

53) Hobbs, Jerry and Rosenschein, Stanley, "Making computational sense of Montague's intensional logic," *Artificial Intelligence, 9*, pp. 287-306, 1978.

54) Hockett, Charles F., "Two models of grammatical description," *Word, 10*, pp. 210-233, 1954.

55) Hopcroft, John and Ullman, Jeffrey, *Introduction to automata theory, languages, and computation*, Addison-Wesley, Reading, Massachusetts, 1979.

56) Horrocks, Geoffrey, "The order of constituents in Modern Greek," in *Order, Concord and Constituency* (Gerald Gazdar, Ewan H. Klein and Geoffrey K. Pullum, eds.), Foris, Dordrecht, pp. 95-112, 1983.

57) Horrocks, Geoffrey, "The lexical head constraint, X'-theory and the 'pro-drop' parameter," in *Sentential Complementation* (Willem de Geest and Yvan Putseys, eds.), Foris, Dordrecht, pp. 117-125, 1984.

58) Huybregts, M. A. C., "Overlapping dependencies in Dutch," *Utrecht Working Papers in Linguistics, 1*, pp. 24-65, 1976.

59) Ikeya, Akira, "Japanese honorific systems in generalized phrase structure grammar," in *Proceedings of the ICOT Workshop on Non-Transformational Grammars*, Institute for New Generation Computer Technology, Tokyo, pp. 17-20, 1983.

60) Indurkhya, B., "Sentence analysis programs based on Montague grammar," *M. E. E. thesis*, Netherlands Universities Foundation for International Cooperation, 1981.

61) Ishimoto, I., "A Lesniewskian version of Montague grammar," in *COLING 82* (Jan Horecky, ed.), North-Holland, Amsterdam, pp. 139-144, 1982.

62) Janssen, Theo, "A computer program for Montague Grammar: theoretical aspects and proofs for the reduction rules," *Amsterdam Papers in Formal Grammar, 1*, pp. 154-176, 1976.

63) Janssen, Theo, "Simulation of a Montague Grammar," *Annals of System Research, 6*, pp. 127-140, 1977.

64) Janssen, Theo, "On problems concerning the quantification rules in Montague grammar," in *Time, tense, and quantifiers* (C. Rohrer, ed.), Max Niemeyer, Tubingen, pp. 113-134, 1980.

65) Janssen, Theo, *Foundations and Applications of Montague Grammar*, Mathematisch Centrum, Amsterdam, 1983.

66) Johnson, C. Douglas, "On the formal properties of phonological rules," *POLA Report, 11*, University of California, Berkeley, 1970.

67) Johnson, S. C., "YACC — yet another compiler compiler," *CSTR, 32*, Bell Laboratories, Murray Hill, N. J., 1975.

68) Joshi, Aravind, "Factoring recursion and dependencies: an aspect of tree-adjoining grammars (TAG) and a comparison of some formal properties of TAGs, GPSGs, PLGs, and LFGs," in *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pp. 7-15, 1983.

69) Joshi, Aravind, "Tree Adjoining Grammars: How much context-sensitivity is required to provide reasonable structural descriptions?" in *Natural Language Processing: Psycholinguistic, Computational, and Theoretic Perspectives* (David R. Dowty, Lauri Karttunen and Arnold M. Zwicky, eds.), Cambridge University Press, New York, New York, 1985.

70) Joshi, Aravind and Levy, Leon, "Phrase structure trees bear more fruit than you would

have thought," *American Journal of Computational Linguistics, 8*, pp. 1-11, 1982.

71) Kameshima, Nanako, "CNPC Violations in Japanese; A GPSG Account," unpublished paper, University of Wisconsin-Madison, 1984.

72) Kaplan, Ronald M. and Bresnan, Joan, "Lexical-functional grammar: a formal system for grammatical representation," in (Bresnan, ed.), pp. 173-281, 1982.

73) Kaplan, Ronald M. and Kay, Martin, "Phonological rules and finite state transducers," *ACL/LSA paper*, New York, 1981.

74) Karttunen, Lauri, "KIMMO: A general morphological processor," in *Texas Linguistic Forum, 22* (Mary Dalrymple et al., eds.), University of Texas, Austin, Texas, pp. 165-186, 1983.

75) Karttunen, Lauri, "Features and values," in *Proceedings of Coling 84*, Association for Computational Linguistics, Menlo Park, pp. 28-33, 1984.

76) Karttunen, Lauri and Wittenburg, Kent, "A two-level morphological analysis of English," in *Texas Linguistic Forum, 22* (Mary Dalrymple et al., eds.), University of Texas, Austin, Texas, pp. 217-228, 1983.

77) Kay, Martin, "Functional grammar," in *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistic Society* (Christina Chiarello et al., eds.), pp. 142-158, 1979.

78) Kay, Martin, "When meta-rules are not meta-rules," in *Automatic Natural Language Parsing* (Karen Sparck-Jones and Yorick Wilks, eds.), Ellis Horwood, Chichester, pp. 94-116, 1983. Also in *Developments in Generalized Phrase Structure Grammar: Stanford Working Papers in Grammatical Theory, Volume 2* (Michael Barlow, Daniel Flickinger, and Ivan A. Sag, eds.), Indiana University Linguistics Club, Bloomington, pp. 69-91, 1983.

79) Kay, Martin, "Functional unification grammar: a formalism for machine translation," in *Proceedings of Coling 84*, Association for Computational Linguistics, Menlo Park, pp. 75-78, 1984.

80) Kay, Martin, "Two-level morphology with tiers," presented to the *CSLI Workshop on Morphology*, July, 1985.

81) Keller, William R.,"Generating logic from ProGram parse trees," *Cognitive Science Research Paper, 39 (CSRP 039)*, University of Sussex, Brighton, England, 1984.

82) Keller, William R., "A lexicon handler for the ProGram grammar development system," *Cognitive Science Research Paper, 40 (CSRP 040)*, University of Sussex, Brighton, England, 1984.

83) Khan, Robert, "A two-level morphological analysis of Rumanian," in *Texas Linguistic Forum, 22* (Mary Dalrymple et al., eds.), University of Texas, Austin, Texas, pp. 253-270, 1983.

84) Khan, Robert, Liu, Jocelyn S., Ito, Tatsuo and Shuldberg, Kelly, "KIMMO user's manual," in *Texas Linguistic Forum, 22* (Mary Dalrymple et al., eds.), University of Texas, Austin, Texas, pp. 203-215, 1983.

85) Kilbury, James, "GPSG-based parsing and generation," to appear in *Probleme des (Text-) Verstehens — Ansatze der Kunstlichen Intelligenz* (Claus-Rainer Rollinger, ed.), Max Niemeyer, Tubingen, 1984.

86) Klein, Ewan H., "The syntax and semantics of nominal comparatives," in *Atti de Seminario su Tempo e Verbale Strutture Quantificate in Forma Logica* (M. Moneglia, ed.), Presso l'Accademia della Crusca, Florence, pp. 223-253, 1981.

87) Konolige, Kurt, "Capturing linguistic generalizations with metarules in an annotated phrase-structure grammar," in *Proceedings of the 18th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Menlo Park, California, pp. 43-48, 1980.

88) Koskenniemi, Kimmo, "Two-level Morphology: A General Computational Model for Word-Form Recognition and Production," *Publications, No. 11*, Department of

General Linguistics, University of Helsinki, Helsinki, 1983.

89) Koskenniemi, Kimmo, "Two-level model for morphological analysis," *Proceedings of IJCAI-83*, pp. 683-685, 1983.

90) Landsbergen, Jan, "Adaptation of Montague grammar to the requirements of parsing," in *Formal Methods in the Study of Language* (Jeroen A. G. Groenendijk, Theo Janssen and Martin Stokhof, eds.), Mathematical Centre Tracts, Amsterdam, pp. 399-419, 1981.

91) Landsbergen, Jan, "Machine translation based on logically isomorphic Montague grammars," in *COLING 82* (Jan Horecky, ed.), North-Holland, Amsterdam, pp. 175-182, 1982.

92) Langendoen, D. Terence, "Finite-state parsing of phrase-structure languages and the status of readjustment rules in grammar," *Linguistic Inquiry, 5*, pp. 533-554, 1975.

93) Langendoen, D. Terence, "On the inadequacy of type-2 and type-3 grammars for human languages," in *Studies in descriptive and historical linguistics* (P. J. Hopper, ed.), John Benjamin, Amsterdam, pp. 159-171, 1977.

94) Langendoen, D. Terence, "The generative capacity of word-formation components," *Linguistic Inquiry, 12*, pp. 320-322, 1981.

95) Langendoen, D. Terence and Langsam, Yedidyah, "The representation of constituent structures for finite-state parsing," in *Proceedings of Coling 84*, Association for Computational Linguistics, Menlo Park, pp. 24-27, 1984.

96) Langendoen, D. Terence and Postal, Paul M., *The Vastness of Natural Languages*, Blackwell, Oxford, 1984.

97) Langendoen, D. Terence and Postal, Paul M., "English and the class of context-free languages," *Computational Linguistics, 10*, pp. 177-181, 1985.

98) Levelt, W. J. M., *Formal Grammars in Linguistics and Psycholinguistics (Vol. II): Applications in Linguistic Theory*, Mouton, The Hague, Holland, 1974.

99) Lun, S., "A two-level morphological analysis of French," in *Texas Linguistic Forum, 22* (Mary Dalrymple et al., eds.), University of Texas, Austin, Texas, pp. 271-278, 1983.

100) Maling, Joan and Zaenen, Annie, "A phrase structure account of Scandinavian extraction phenomena," in *The Nature of Syntactic Representation* (Pauline Jacobson and Geoffrey K. Pullum, eds.), D. Reidel, Dordrecht, pp. 229-282, 1982.

101) Marsh, William E. and Partee, Barbara H., "How non-context-free is variable binding?," in *Proceedings of the Third West Coast Conference on Formal Linguistics* (Mark Cobler et al., eds.), Stanford Linguistics Association, Stanford, California, pp. 179-190, 1984.

102) Matsumoto, Yuji, "Software implementation of Montague grammar and related problems," in *Formal Approaches to Natural Language: Proceedings of the First Colloquium on Montague Grammar and Related Topics* (Shogo Iguchi, ed.), Kyoto Working Group of Montague Grammar, Kyoto, pp. 148-158, 1981.

103) Matsumoto, Yuji, "A Montague grammar of Japanese with special regard to meaning adjustment," paper presented to the *Second Colloquium on Montague Grammar and Related Topics*, Kyoto, March, 1982.

104) McCarthy, John J., "Formal Problems in Semitic Phonology and Morphology," *PhD thesis*, MIT, 1979. Reproduced by the Indiana University Linguistics Club, Bloomington, Indiana, 1982.

105) Montague, Richard, *Formal Philosophy*, Yale University Press, New Haven, Connecticut, 1974.

106) Moran, Douglas B., "Dynamic partial models," *PhD dissertation*, University of Michigan, Ann Arbor, Michigan, 1980.

107) Nerbonne, John, "German temporal semantics: three-dimensional tense logic and a GPSG fragment," *Working Papers in Linguistics, 30*, Ohio State University, Colum-

bus, Ohio, 1984.

108) Nishida, Toyo-aki and Doshita, Shuji, "An English-Japanese machine translation system based on formal semantics of natural language — a progress report," in *COLING 82* (Jan Horecky, ed.), North-Holland, Amsterdam, pp. 277-282, 1982.

109) Nishida, Toyo-aki and Doshita, Shuji, "An application of Montague Grammar to English-Japanese machine translation," in *Proceedings of the Conference on Applied Natural Language Processing* (Santa Monica, California), Association for Computational Linguistics, Menlo Park, California, February, 1983.

110) Nishida, Toyo-aki, Kiyono, Masaki and Doshita, Shuji, "An English-Japanese machine translation system based on formal semantics of natural language," in *Formal Approaches to Natural Language: Proceedings of the First Colloquium on Montague Grammar and Related Topics*, Kyoto Working Group of Montague Grammar, Kyoto, pp. 104-147, 1981.

111) Pereira, Fernando C. N., "A new characterization of attachment preferences," in *Natural Language Processing: Psycholinguistic, Computational and Theoretical Perspectives* (David R. Dowty, Lauri Karttunen and Arnold M. Zwicky, eds.), Cambridge University Press, New York, New York, 1985.

112) Pereira, Fernando C. N. and Shieber, Stuart M., "The semantics of grammar formalisms seen as computer languages," in *Proceedings of Coling 84*, Association for Computational Linguistics, Menlo Park, pp. 123-129, 1984.

113) Perrault, C. Raymond, "On the mathematical properties of linguistic theories," *Computational Linguistics* (formerly *American Journal of Computational Linguistics*), *10*, pp. 165-176, 1985.

114) Phillips, John D. and Thompson, Henry S., "GPSGP — A parser for generalized phrase structure grammars," to appear in *Linguistics, 23*, 1985.

115) Pollard, Carl J., "Generalized Phrase Structure Grammars, Head Grammars, and Natural Languages," *PhD thesis*, Stanford University, Stanford, California, 1984.

116) Pollard, Carl and Creary, Lewis, "A computational semantics for natural language," *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Morristown, NJ, 1985.

117) Postal, Paul, "Limitations of phrase structure grammars," in *The structure of language: readings in the philosophy of language* (J. A. Fodor and J. J. Katz, eds.), Prentice-Hall, Englewood Cliffs, New Jersey, pp. 137-151, 1964.

118) Proudian, Derek and Pollard, Carl,"Parsing head-driven phrase structure grammar," *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Morristown, NJ, 1985.

119) Pullum, Geoffrey K., "On two recent attempts to show that English is not a CFL," *Computational Linguistics* (formerly *American Journal of Computational Linguistics*), *10*, pp. 182-186, 1985.

120) Pullum, Geoffrey K. and Gazdar, Gerald, "Natural languages and context free languages," *Linguistics and Philosophy, 4*, pp. 471-504, 1982.

121) Pulman, Stephen, "Generalised phrase structure grammar, Earley's algorithm, and the minimisation of recursion," in *Automatic Natural Language Parsing* (Karen Sparck-Jones and Yorick Wilks, eds.), Ellis Horwood, Chichester, pp. 117-131, 1983.

122) Pulman, Stephen, "Limited domain systems for language teaching," in *Proceedings of Coling 84*, Association for Computational Linguistics, Menlo Park, pp. 84-87, 1984.

123) Rich, Elaine, *Artificial Intelligence*, McGraw-Hill, New York, New York, 1983.

124) Roach, Kelly, "Formal properties of head grammars," unpublished paper, Xerox Palo Alto Research Center, Palo Alto, California, 1984.

125) Robinson, Jane, "Computational aspects of the use of metarules in formal grammars," *Research Proposal No. ECU 80-126*, S. R. I. International, Menlo Park, California,

1980.

126) Robinson, Jane, "DIAGRAM: a grammar for dialogs," *Communications of the ACM, 25*, pp. 27-47, 1982.

127) Root, Rebecca, "SMX: a program for translating English into Montague's intensional logic," unpublished manuscript, University of Texas at Austin, Austin, Texas, 1981.

128) Rosenbloom, Paul, *The Elements of Mathematical Logic*, Dover, New York, New York, 1950.

129) Rosenschein, S. J. and Shieber, Stuart M., "Translating English into logical form," in *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Menlo Park, California, pp. 1-8, 1982.

130) Ross, Kenneth, "Parsing English phrase structure," *PhD dissertation*, University of Massachusetts at Amherst, Amherst, Massachusetts, 1981.

131) Ross, Kenneth, "An improved left-corner parsing algorithm," in *COLING 82* (Jan Horecky, ed.), North-Holland, Amsterdam, pp. 333-338, 1982.

132) Rounds, William C., "Complexity of recognition in intermediate-level languages," *Proceedings of the IEEE Symposium on Switching and Automata Theory*, pp. 145-158, 1973.

133) Sadock, Jerrold M., "Autolexical syntax: A theory of noun incorporation and similar phenomena," in *Natural Language and Linguistic Theory*, in press, 1985.

134) Sag, Ivan A., "A semantic theory of 'NP-movement' dependencies," in *The Nature of Syntactic Representation* (Pauline Jacobson and Geoffrey K. Pullum, eds.), D. Reidel, Dordrecht, pp. 427-466, 1982.

135) Sag, Ivan A., "On parasitic gaps," *Linguistics and Philosophy, 6*, pp. 35-45, 1983. Also in *Proceedings of the First West Coast Conference on Formal Linguistics* (Daniel Flickinger, Marlys Macken and Nancy Wiegand, eds.), Stanford Linguistics Department, Stanford, pp. 35-46, 1982.

136) Saheki, Motoji, "A software program for a language like natural language," paper presented to the *Second Colloquium on Montague Grammar and Related Topics*, Kyoto, March, 1982.

137) Saito, Mamoru, "An analysis of the *tough* construction in Japanese," *MA dissertation*, Stanford University, Stanford, California, 1980.

138) Sampson, Geoffrey, "Context-free parsing and the adequacy of context-free grammars," in *Parsing Natural Language* (Margaret King, ed.), Academic Press, London, pp. 151-170, 1983.

139) Sawamura, Hajime, "Intensional logic as a basis of algorithmic logic," paper presented to the *First Colloquium on Montague Grammar and Related Topics*, Kyoto, February, 1981.

140) Schnelle, Helmut, "Concurrent parsing in programmable logic array (PLA-) nets: problems and proposals," in *Proceedings of Coling 84*, Association for Computational Linguistics, Menlo Park, pp. 150-153, 1984.

141) Schubert, Lenhart, "An approach to the syntax and semantics of affixes in 'conventionalized' phrase structure grammar," in *Proceedings of the 4th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, pp. 189-195, 1982.

142) Schubert, Lenhart and Pelletier, Jeffry, "From English to logic: Context-free computation of 'conventional' logical translation," *American Journal of Computational Linguistics, 8*, pp. 27-44, 1982.

143) Selkirk, Elisabeth O., *The Syntax of Words*, MIT Press, Cambridge, Massachusetts, 1982.

144) Shieber, Stuart M., "Sentence disambiguation by a shift-reduce parsing technique," in

Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics, pp. 113-118, 1983.

145) Shieber, Stuart M., "Direct parsing of ID/LP grammars," *Linguistics and Philosophy, 7*,pp. 135-154, 1984.

146) Shieber, Stuart M., "Evidence against the context-freeness of natural language," to appear in *Linguistics and Philosophy*,*8*, 1985.

147) Shieber, Stuart M., Stucky, Susan, Uszkoreit, Hans and Robinson, Jane, "Formal constraints on metarules," *Technical Note, 283*, SRI International, Menlo Park, California, 1983. Also in *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics,* pp. 22-27, 1983.

148) Shirai, Hidetoshi, "Deterministic parser," in *Proceedings of the ICOT Workshop on Non-Transformational Grammars*, Institute for New Generation Computer Technology, Tokyo, pp. 57-61, 1983.

149) Slocum, Jonathan, Bennett, Winfield S., Bear, John, Morgan, Martha and Root, Rebecca, "METAL: The LRC machine translation system," *Linguistics Research Center Working Paper LRC-84-2*, Austin, Texas, 1984.

150) Sondheimer, Norman and Gunji, Takao, "Applying model-theoretic semantics to natural language understanding: representation and question-answering," in *Proceedings of the Seventh International Conference on Computational Linguistics*, Bergen, Norway, 1978.

151) Steedman, Mark, "Dependency and coordination in the grammar of Dutch and English," to appear in *Language*, 1985.

152) Stoy, Joseph E., *Denotational Semantics*: *The Scott-Strachey Approach to the Semantics of Programming Languages*, MIT Press, Cambridge, Massachusetts, 1977.

153) Stucky, Susan, "Word order variation in Makua: a phrase structure grammar analysis," *PhD dissertation*, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1981.

154) Stucky, Susan, "Metarules as meta-node-admissibility conditions," *Technical Note, 304*, SRI International, Menlo Park, California, 1983.

155) Stucky, Susan, "Verb phrase constituency and linear order in Makua," in *Order, Concord and Constituency* (Gerald Gazdar, Ewan H. Klein and Geoffrey K. Pullum, eds.), Foris, Dordrecht, pp. 75-94, 1983.

156) Thompson, Henry, "Chart parsing and rule schemata in PSG," in *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Menlo Park, California, pp. 167-172, 1981.

157) Thompson, Henry, "Handling metarules in a parser for GPSG," *Edinburgh D. A. I. Research Paper, No. 175*, University of Edinburgh, U. K., 1982. Also in *Developments in Generalized Phrase Structure Grammar*: *Stanford Working Papers in Grammatical Theory, Volume 2* (Michael Barlow, Daniel Flickinger and Ivan A. Sag, eds.), Indiana University Linguistics Club, Bloomington, pp. 26-37. Also in *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pp. 26-37.

158) Thompson, Henry, "Crossed serial dependencies: a low-power parseable extension to GPSG," in *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pp. 16-21, 1983.

159) Thompson, Henry and Phillips, John, "An implementation of GPSG within the MCHART chart parsing framework," *Technical Report*, Department of Artificial Intelligence, University of Edinburgh, U. K., 1984.

160) Tomita, Masaru, "LR parsers for natural languages," in *Proceedings of Coling 84*, Association for Computational Linguistics, Menlo Park, pp. 354-357, 1984.

161) Udo, Mariko, "Syntax and morphology of the Japanese verb — a phrase structural approach," *MA thesis*, University College London, 1982.

162) Uszkoreit, Hans, "German word order in GPSG," in *Proceedings of the First West Coast Conference on Formal Linguistics* (Daniel Flickinger, Marlys Macken and Nancy Wiegand, eds.), Stanford Linguistics Department, Stanford, pp. 137-148, 1982.

163) Uszkoreit, Hans, "A framework for processing partially free word order," in *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pp. 106-112, 1983.

164) Warren, David S., "Syntax and semantics in parsing: an application to Montague Grammar," *PhD dissertation*, University of Michigan, Ann Arbor, Michigan, 1979.

165) Warren, David S. and Friedman, J., "Using semantics in non-context-free parsing of Montague grammar," *American Journal of Computational Linguistics, 8*, pp. 123-138, 1982.

166) Wijngaarden, A. van, "Report on the algorithmic language ALGOL68," *Numerische Mathematik, 14*, pp. 79-218, 1969.

167) Woods, William A., "Cascaded ATN grammars," *American Journal of Computational Linguistics, 6*, pp. 1-12, 1980.

168) Zwicky, Arnold M., "German adjective agreement in GPSG," to appear in *Linguistics*, 1985.