SYNTACTIC AND SEMANTIC PARSABILITY

Geoffrey K. Pullum

Syntax Research Center, Cowell College, UCSC, Santa Cruz, CA 95064
and
Center for the Study of Language and Information, Stanford, CA 94305

ABSTRACT

This paper surveys some issues that arise in the study of the syntax and semantics of natural languages (NL´s) and have potential relevance to the automatic recognition, parsing, and translation of NL´s. An attempt is made to take into account the fact that parsing is scarcely ever thought about with reference to syntax alone; semantic ulterior motives always underly the assignment of a syntactic structure to a sentence. First I consider the state of the art with respect to arguments about the language-theoretic complexity of NL´s: whether NL´s are regular sets, deterministic CFL´s, CFL´s, or whatever. While English still appears to be a CFL as far as I can tell, new arguments (some not yet published) appear to show for the first time that some languages are not CFL´s. Next I consider the question of how semantic filtering affects the power of grammars. Then I turn to a brief consideration of some syntactic proposals that employ more or less modest extensions of the power of context-free grammars.

1. INTRODUCTION

Parsing as standardly defined is a purely syntactic matter. Dictionaries describe parsing as analysing a sentence into its elements, or exhibiting the parts of speech composing the sentence and their relation to each other in terms of government and agreement. But in practice, as soon as parsing a natural language (NL) is under discussion, people ask for much more than that. Let us distinguish three kinds of algorithm operating on strings of words:

recognition
: output: a decision concerning whether the string is a member of the language or not

parsing
: output: a syntactic analysis of the string (or an error message if the string is not in the language)

translation
: output: a translation (or set of translations) of the string into some language of semantic representation (or an error message if the string is not in the language)

Much potential confusion will be avoided if we are careful to use these terms as defined. However, further refinement is needed. What constitutes a "syntactic analysis of the string" in the definition of parsing? In applications development work and when modeling the whole of the native speaker´s knowledge of the relevant part of the language, we want ambiguous sentences to be repesented as such, and we want *Time flies like an arrow* to be mapped onto a whole list of different structures. For rapid access to a database or other back-end system in an actual application, or for modeling a speaker´s performance in a conversational context, we will prefer a program that yields one syntactic description in response to a given string presentation. Thus we need to refer to two kinds of algorithm:

all-paths parser
: output: a list of all structural descriptions of the string that the grammar defines (or an error message if the string is not in the language)

one-path parser
: output: one structural description that the grammar defines for the string (or an error message if the string is not in the language)

By analogy, we will occasionally want to talk of all-paths or one-path recognizers and translators as well.

There is a crucial connection between the theory of parsing and the theory of languages. There is no parsing without a definition of the language to be parsed. This should be clear enough from the literature on the definition and parsing of programming languages, but for some reason it is occasionally denied in the context of the much larger and richer multi-purpose languages spoken by humans. I frankly cannot discern a sensible interpretation of the claims made by some artificial intelligence researchers about parsing a NL without having a defined syntax for it. Assume that some program *P* produces finite, meaningful responses to sentences from some NL *L* over some terminal vocabulary *T*, producing error messages of some sort in response to nonsentences. It seems to me that automatically we have a generative grammar for *L*. Moreover, since *L* is clearly recursive, we can even enumerate the sentences of *L* in canonical order. One algorithm to do this simply enumerates the strings over the terminal vocabulary in order of increasing length and in alphabetical order within a given string-length, and for each one, tests it for grammaticality using *P*, and adds it to the output if no error message is returned.

Given that parsability is thus connected to definability, it has become standard not only for parser-designers to pay attention to the grammar for the language they are trying to parse, but also for linguists to give some thought to the parsability claims entailed by their linguistic theory. This is all to the good, since it would hardly be sensible for the study of NL´s to proceed for ever in isolation from the study of ways in which they can be used by finite organisms.

Since 1978, following suggestions by Stanley Peters, Aravind Joshi, and others, developed most notably in the work of Gerald Gazdar, there has been a strong resurgence of the idea that context-free phrase structure grammars could be used for the description of NL´s. A significant motivation for the original suggestions 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 context-free languages (CFL´s).

This was not, however, the motivation for the interest that signficant numbers of linguists began to show in context-free phrase structure grammars (CF-PSG´s) from early 1979. Their motivation was in nearly all cases an interest sparked by the elegant solutions to purely linguistic problems that Gazdar and others began to put forward in various articles, initially unpublished working papers. We have now seen nearly half a decade of work using CF-PSG to successfully tackle problems in linguistic description (the Coordinate Structure Constraint (Gazdar 1981a), the English auxiliary system (Gazdar et al. 1982), etc.) that had proved somewhat recalcitrant even for the grossly more powerful transformational theories of grammar that had formerly dominated linguistics. The influence of the parsing argument on linguists has probably been overestimated. It seems to me that when Gazdar (1981b, 267) says

> our grammars can be shown to be formally equivalent to what are known as the context-free phrase structure grammars [which] has the effect of making potentially relevant to natural language grammars a whole literature of mathematical results on the parsability and learnability of context-free phrase structure grammars

he is making a point exactly analogous to the one made by Robert Nozick in his book *Anarchy, State and Utopia*, when he says of a proposed social organization (1974, 302):

> We seem to have a realization of the economists´ model of a competitive market. This is most welcome, for it gives us immediate access to a powerful, elaborate, and sophisticated body of theory and analysis.

We are surely not to conclude from this remark of Nozick´s that his libertarian utopia of interest groups competing for members is motivated solely by a desire to have a society that functions like a competitive market. The point is one of serendipity: if a useful theory turns out to be equivalent to one that enjoys a rich technical literature, that is very fortunate, because we may be able to make use of some of the results therein.

The idea of returning to CF-PSG as a theory of NL´s looks retrogressive until one realizes that the arguments that had led linguists to consign CF-PSG´s to the scrap-heap of history can be shown to be fallacious (cf. especially Pullum and Gazdar (1982)). In view of that development, I think it would be reasonable for someone to ask whether we could not return all the way to finite-state grammars, which would give us even more efficient parsing (guaranteed deterministic linear time). It may therefore be useful if I briefly reconsider this question, first dealt with by Chomsky nearly thirty years ago.

2. COULD NL´S BE REGULAR SETS?

Chomsky´s negative answer to this question was the correct one. Although his original argument in *Syntactic Structures* (1957) for the non-regular character of English was not given in anything like a valid form (cf. Daly 1974 for a critique), others can be given. Consider the following, patterned after a suggestion by Brandt Corstius (see Levelt 1974, 25-26). The set (1):

(1) {a white male (whom a white male)^n (hired)^n hired another white male | n $\geq$ 0}

is the intersection of English with the regular set *a white male* (*whom a white male*)* *hired** *another white male*. But (1) is not regular, yet the regular sets are closed under intersection; hence English is not regular. Q.E.D.

It is perfectly possible that some NL´s happen not to present the inherently self-embedding configurations that 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 not at all uncommon. However, it should not be thought that non-regular configurations will be found to be rare in languages of the world. There are likely to be many languages that furnish better arguments for non-regular character than English does; for example, according to Hagège (1976), center-embedding seems to be commoner and more acceptable in several Central Sudanic languages than it is in English. In Moru, we find examples such as this (slightly simplified from Hağege (1976, 200); *ri* is the possession marker for nonhuman nouns, and *ro* is the equivalent for human nouns):

```
(2) kokyE [toko [odrupi  [ma ro] ro] ri] drate
           1     2        3     3   2  1
    dog   wife  brother  me of  of  of  is-dead
    ´My brother´s chief wife´s black dog is dead.´
```

The center-embedding word order here is the only one allowed; the alternative right-branching order ("dog chief-wife-of brother-of me-of"), which a regular grammar could handle, is ungrammatical. Presumably, the intersection of *odrupi** *ma ro** *drate* with Moru is

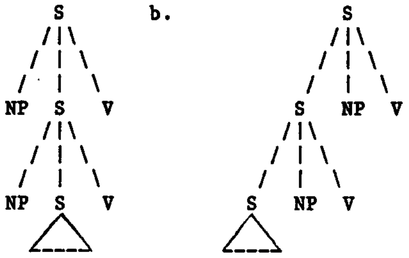
{odrupi^n ma ro^n drate | n > 0}

(an infinite set of sentences with meanings like "My brother´s brother´s brother is dead" where n = 3). This clearly non-regular, hence so is Moru.

The fact that NL´s are not regular does not necessarily mean that techniques for parsing regular languages are irrelevant to NL parsing. Langendoen (1975) and Church (1980) have both, 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 the human parsing ability, 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.

The fact that NL´s 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 union, concatenation, and Kleene closure on finite vocabularies, and there is no apparent *a priori* reason why humans could not have been well served by regular languages. Expressibility considerations, for example, do not appear to be relevant: there is no reason why a regular language could not express any proposition expressible by a sentence of any finite-string-length language. Indeed, many languages provide ways of expressing sentences with self-embedding structure in non-self-embedding ways as well. In an SOV language like Korean, for example, sentences with the tree-structure (3a) are also expressible with left-branching tree-structure as shown in (3b).

(3)a.



Clearly such structural rearrangement will not alter the capacity of a language to express propositions, any more than an optimizing compiler makes certain programs inexpressible when it irons out true recursion into tail recursion wherever possible.

If NL´s were regular sets, we know we could recognize them in deterministic linear time using the fastest and simplest abstract computing devices of all, finite state machines. However, there are much larger classes of languages that have linear time recognition. One such class is the deterministic context-free languages (DCFL´s). It might be reasonable, therefore, to raise the question dealt with in the following section.

3. COULD NL´S BE DCFL´S?

To the best of my knowledge, this question has never previously been raised, much less answered, in the literature of linguistics or computer science. Rich (1983) is not atypical in dismissing the entire literature on DCFL´s without a glance on the basis of an invalid argument which is supposed to show that English is not even a CFL, hence *a fortiori* not a DCFL.

I should make it clear that the DCFL´s are not just those CFL´s for which someone has written a parser that is in some way deterministic. They are the CFL´s that are accepted by some deterministic pushdown stack automaton. The term "deterministic parsing" is used in many different ways (cf. Marcus (1980) for an attempt to motivate a definition of determinism specifically for the parsing of NL´s). For example, a translator system with a post-processor to rank quantifier-scope ambiguities for plausibility and output only the highest-ranked translation might be described as deterministic, but there is no reason why the language it recognizes should be a DCFL; it might be any recursive language. The parser currently being implemented by the natural language team at HP Labs (in particular, by Derek Proudian and Dan Flickinger) introduces an interesting compromise between determinism and nondeterminism in that it ranks paths through the rule system so as to make some structural possibilities highly unlikely ones, and there is a toggle that can be set to force the output to contain only likely parses. When this option is selected, the parser runs faster, but can still show ambiguities when both readings are defined as likely. This is an intriguing development, but again is irrelevant to the language-theoretic question about DCFL status that I am raising.

It would be an easy slip to assume that NL´s cannot be DCFL´s on the grounds that English is well known to be ambiguous. We need to distinguish carefully between ambiguity and *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 DCFL´s, hence no inherently ambiguous language is a DCFL. But it has never been argued, as far as I know, that English as a stringset 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. In fact, *all* languages have ambiguous grammars. (The proof is trivial. Let w be a string in a language *L* generated by a grammar *G* with initial symbol *S* and production set *P*. Let *B* be a nonterminal not used by *G*. Construct a new grammar *G´* with production set *P´* = *P* U {*S* --> *B*, *B* --> *w*}. *G´* is an ambiguous grammar that assigns two structural descriptions to w.)

The relevance of this becomes clear when we observe that in natural language processing 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 impemented natural language processing system in which the language accepted is described by an ambiguous CF-PSG but is nonetheless (weakly) a DCFL. When access to all possible analyses of an input is desired (say, in development work, or when one wants to take no risks in using a database front end), an all-paths parser/translator is used, but when quick-and-dirty responses are required, at the risk of missing certain potential parses of ambiguous strings, this is replaced by a deterministic one-path parser. Despite the difference in results, the language analyzed and the grammar used could be the same.

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 1978), is explored for natural languages in work by Fernando Pereira and Stuart Shieber. Shieber (1983) 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 (1984). Shieber (1983, 116) gives two rules for resolving conflicts between parsing actions:

(I) Resolve shift-reduce conflicts by shifting.

(II) Resolve reduce-reduce conflicts by performing the longer reduction.

The first of these is exactly the same as the one given for Yacc by Johnson (1978, 13). The second is more principled than the corresponding Yacc rule, which simply says that a rule listed earlier in the grammar should take precedence over a rule listed later to resolve a reduce-reduce conflict. But it is particularly interesting that the two are in practice equivalent in all sensible cases, for reasons I will briefly explain.

A reduce-reduce conflict arises when a string of categories on the stack appears on the right hand side of two different rules in the grammar. If one of the reducible sequences is longer than the other, it must properly include the other. But in that case the prior application of the properly including rule is mandated by an extension into parsing theory of the familiar rule interaction principle of Proper Inclusion Precedence, due originally to the ancient Indian grammarian Panini (see Pullum 1979, 81-86 for discussion and references). Thus, if a rule NP --> NP PP were ordered before a rule VP --> V NP PP in the list accessed by the parser, it would be impossible for the sequence "NP PP" ever to appear in a VP, since it would always be reduced to NP by the earlier rule; the VP rule is useless, and could have been left out of the grammar. But if the rule with the properly including expansion "V NP PP" is ordered first, the NP rule is not useless. A string "V NP PP PP", for example, could in principle be reduced to "V NP PP" by the NP rule and then to "VP" by the VP rule. Under a principle of rule interaction made explicit in the practice of linguists, therefore, the proposal made by Pereira and Shieber can be seen to be largely equivalent to the cruder Yacc resolution procedure for deterministic parsing with ambiguous grammars.

Techniques straight out of programming language and compiler design may, therefore, be of considerable interest in the context of natural language processing applications. Indeed, Shieber goes so far as to suggest psycholinguistic implications. He considers the class of "garden-path sentences" such as those in (4).

(4) The diners hurried through their meal were annoyed.
That shaggy-looking sheep should be sheared is important.

On these, his parser fails. Strictly speaking, therefore, they indicate that the language parsed is not the same under the one-path and the all-paths parsers. But interestingly, human beings are prone to fail just as badly as Shieber's parser on sentences such as these. The trouble with these cases is that they lack the prefix property---that is, they have an initial proper substring which is a sentence. (From this we know that English does not have an LR(0) grammar, incidentally.) English speakers tend to mis-parse the prefix as a sentence, and baulk at the remaining portion of the string. We might think of characterizing the notion "garden-path sentence" in a rigorous and non-psychological way in terms of an all-paths parser and a deterministic one-path parser for the given language: the garden path sentences are just those that parse under the former but fail under the latter.

To say that there might be an appropriate deterministic parser for English that fails on certain sentences, thus defining them as garden-path sentences, is not to deny the existence of a deterministic pushdown automaton that accepts the whole of English, garden-path sentences included. It is an open question, as far as I can see, whether English as a whole is weakly a DCFL. The likelihood that the answer is positive is increased by the results of Bermudez (1984) concerning the remarkable power and richness of many classes of deterministic parsers for subsets of the CFL's.

If the answer were indeed positive, we would have some interesting corollaries. To take just one example, the intersection between two dialects of English that were both DCFL's would itself be a DCFL (since the DCFL's are closed under intersection). This seems right: if your dialect and mine share enough for us to communicate without hindrance, and both our dialects are DCFL's, it would be peculiar indeed if our shared set of mutually agreed-upon sentences was not a DCFL. Yet with the CFL's in general we do not have such a result. Claiming merely that English dialects are CFL's would not rule out the strange situation of having a pair of dialects, both CFL's, such that the intersection is not a CFL.

4. ARE ALL NL'S CFL'S?

More than a quarter-century of mistaken efforts have attempted to show that not all NL's are CFL's. This history is carefully reviewed by Pullum and Gazdar (1982). But there is no reason why future attempts should continue this record of failure. It is perfectly clear what sorts of data from a NL would show it to be outside the class of CFL's. For example, an infinite intersection with a regular set having the form of a triple-counting language or a string matching language (Pullum 1983) would suffice. However, the new arguments for non-context-freeness of English that have appeared between 1982 and the present all seem to be quite wide of the mark.

Manaster-Ramer (1983) points to the contemptuous reduplication pattern of Yiddish-influenced English, and suggests that it instantiates an infinite string matching language. But does our ability to construct phrases like *Manaster-Ramer Schmanaster-Ramer* (and analogously for any other word or phrase) really indicate that the syntax of English constrains the process? I do not think so. Manaster-Ramer is missing the distinction between the structure of a language and the culture of verbal play associated with it. I can speak in rhyming couplets, or with adjacent word-pairs deliberately Spoonerized, or solely in sentences having an even number of words, if I wish. The structure of my language allows for such games, but does not legislate regarding them.

Higginbotham (1984) presents a complex pumping-lemma argument on the basis of the alleged fact that sentences containing the construction *a N such that S* always contain an anaphoric pronoun within the clause S that is in syntactic agreement with the noun N. But his claim is false. Consider a phrase like *any society such that more people get divorced than get married in an average year*. This is perfectly grammatical, but has no overt anaphoric pronoun in the *such that* clause. (A similar example is concealed elsewhere in the text of this paper.)

Langendoen and Postal (1984) consider sentences like *Joe was talking about some bourbon-lover, but WHICH bourbon-lover is unknown*, and argue that a compound noun of any length can replace the first occurrence of *bourbon-lover* provided the same string is substituted for the second occurrence as well. They claim that this yields an infinite string matching language extractable from English through intersection with a regular set. But this argument presupposes that the ellipsis in *WHICH bourbon-lover* [*Joe was talking about*] must find its antecedent in the current sentence. This is not so. Linguistic accounts of anaphora have often been overly fixated on the intrasentential syntactic conditions on antecedent-anaphor pairings. Artificial intelligence researchers, on the other hand, have concentrated more on the resolution of anaphora within the larger context of the discourse. The latter emphasis is more likely to bring to our attention that ellipsis in one sentence can have its resolution through material in a preceding one. Consider the following exchange:

(5)A: It looks like they're going to appoint another bourbon-hater as Chair of the Liquor Purchasing Committee.

B: Yes--even though Joe nominated some bourbon-lovers; but WHICH bourbon-hater is still unknown.

It is possible for the expression *WHICH bourbon-hater* in B's utterance to be understood as *WHICH bourbon-hater* [*they're going to appoint*] despite the presence in the same sentence of a mention of bourbon-lovers. There is thus no reason to believe that Langendoen and Postal's crucial example type is syntactically constrained to take its antecedent from within its own sentence, even though that is the only interpretation that would occur to the reader when judging the sentence in isolation.

Nothing known to me so far, therefore, suggests that English is syntactically other than a CFL; indeed, I know of no reason to think it is not a deterministic CFL. As far as engineering is concerned, this means that workers in natural language processing and artificial intelligence should not overlook (as they generally do at the moment) the possibilities inherent in the technology that has been independently developed for the computer processing of CFL's, or the mathematical results concerning their structures and properties.

From the theoretical standpoint, however, a different issue arises: is the context-free-ness of English just an accident, much like the accident it would be if we found that Chinese was regular? Are there other languages that genuinely show non-context-free properties? I devote the next section to this question, because some very important results bearing on it have been reported recently. Since these results have not yet been published, I will have to summarize them rather abstractly, and cite forthcoming or in-preparation papers for further details.

5. NON-CONTEXT-FREENESS IN NATURAL LANGUAGES

Some remarkable facts recently reported by Christopher Culy suggest that the African language Bambara (Mande family, spoken in Senegal, Mali, and Upper Volta by over a million speakers) may be a non-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." He then observes that 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 a hypothetical situation sketched by Langendoen (1981), in which reduplication applies to a class of stems that have no upper length bound. Culy (forthcoming) attempts to provide a formal demonstration that this phenomenon renders Bambara non-context-free.

If Bambara turns out to have a reduplication rule defined on strings of potentially unbounded length, then so might other languages. It would be reasonable, therefore, to investigate the case of Engenni (another African language, in the Kwa family, spoken in Rivers State, Nigeria by about 12,000 people). Carlson (1983), citing Thomas (1978), notes that Engenni is reported to have a phrasal reduplication construction: the final phrase of the clause is reduplicated to indicate "secondary aspect." Carlson is correct in noting that if there is no grammatical upper bound to the length of a phrase that may be reduplicated, there is a strong possibility that Engenni could be shown to be a non-CFL.

But it is not only African languages in which relevant evidence is being turned up. Swiss German may be another case. In Swiss German, there is evidence of a pattern of word order in subordinate infinitival clauses that is very similar to that observed in Dutch. Dutch shows a pattern in which an arbitrary number of noun phrases (NP′s) may be followed by a finite verb and an arbitrary number of nonfinite verbs, and the semantic relations between them exhibit a crossed serial pattern---i.e. 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. Bresnan et al. (1982) have shown that a CF-PSG could not assign such a set of dependencies syntactically, but as Pullum and Gazdar (1982, section 5) show, this does not make the stringset non-context-free. It is a semantic problem rather than a syntactic one. In Swiss German, however, there is a wrinkle that renders the phenomenon syntactic: 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 there are two verbs *v* and *v*′ and two nouns *n* and *n*′, the set

> {*xy* | *x* is in (n, n′)* and *y* is in (v, v′)* and for all *i*, if the *i*′th member of *x* is *n* the *i*′th member of y is *v*}

is not a CFL. Shieber (1984) has gathered data from Swiss German to support a rigorously formulated argument along these lines that the language is indeed not a CFL because of this construction.

It is possible that other languages will have properties that render them non-context-free. One case discussed in 1981 in unpublished work by Elisabet Engdahl and Annie Zaenen concerns Swedish. In Swedish, there are three grammatical genders, and adjectives agree in gender with the noun they describe. Consider the possibility of a "respectively"-sentence with a meaning like "The N1, N2, and N3 are respectively A1, A2, and A3," where N1, N2, and N3 have different genders and A1, A2, and A3 are required to agree with their corresponding nouns in gender. If the gender agreement were truly a syntactic matter (*contra* Pullum and Gazdar (1982, 500-501, note 12)), there could be an argument to be made that Swedish (or any language with these sort of facts) was not a CFL.

It is worth noting that arguments based on the above sets of facts have not yet been published for general scholarly scrutiny. Nonetheless, what I have seen convinces me that it is now very likely that we shall soon see a sound published demonstration that some natural language is non-context-free. It is time to consider carefully what the implications are if this is true.

6. CONTEXT-FREE GRAMMARS AND SEMANTIC FILTERING

What sort of expressive power do we obtain by allowing the definition of a language to be given jointly by the syntax and the semantics rather than just by the syntax, so that the syntactic rules can generate strings judged ill-formed by native speakers provided that the semantic rules are unable to assign interpretations to them?

This idea may seem to have a long history, in view of the fact that generative grammarians engaged in much feuding in the seventies over the rival merits of grammars that let "semantic" factors constrain syntactic rules and grammars that disallowed this but allowed "interpretive rules" to filter the output of the syntax. But in fact, the sterile disputes of those days were based on a use of the term "semantic" that bore little relation to its original or current senses. Rules that operated purely on representations of sentence structure were called "semantic" virtually at whim, despite matching perfectly the normal definition of "syntactic" in that they concerned relations holding among linguistic signs. The disputes were really about differently ornamented models of syntax.

What I mean by semantic filtering my be illustrated by reference to the analysis of expletive NP′s like *there* in Sag (1982). It is generally taken to be a matter of syntax that the dummy pronoun subject *there* can appear as the subject in sentences like *There are some knives in the drawer* but not in strings like **There broke all existing records*. Sag simply allows the syntax to generate structures for strings like the latter. He characterizes them as deviant by assigning to *there* a denotation (namely, an identity function on propositions) that does not allow it to combine with the translation of ordinary VP′s like *broke all existing records*. The VP *are some knives in the drawer* is assigned by the semantic rules a denotation the same as that of the sentence *Some knives are in the drawer*, so *there* combines with it and a sentence meaning is obtained. But *broke all existing records* translates as a property, and no sentence meaning is obtained if it is given *there* as its subject. This is the sort of move that I will refer to as semantic filtering.

A question that seems never to have been considered carefully before is what kind of languages can be defined by providing a CF-PSG plus a set of semantic rules that leave some syntactically generated sentences without a sentence meaning as their denotation. For instance, in a system with a CF-PSG and a denotational semantics, can the set of sentences that get assigned sentence denotations be non-CF?

I am grateful to Len Schubert for pointing out to me that the answer is yes, and providing the following example. Consider the following grammar, composed of syntactic rules paired with semantic translation schemata.

(6)	S --> L R	F(L′(R′))
	L --> C	C′
	R --> C	C′
	C --> a	a′
	C --> b	b′
	C --> aC	G(C′)
	C --> bC	H(C′)

Assume that there are two basic semantic types, *A* and *B*, and that *a*′ and *b*′ are constants denoting entities of types *A* and *B* respectively. *F*, *G*, and *H* are cross-categorial operators. *F*(*X*) has the category of functions from *X*-type things to *B*-type things, *G*(*X*) has the category of functions from *A*-type things to *X*-type things, and *H*(*X*) has the category of functions from *B*-type things to *X*-type things. Given the semantic translation schemata, every different *X* constituent has a unique semantic category; the structure of the string is coded into the structure of its translation. But the first rule only yields a meaning for the S constituent if *L′* and *R′* are of the same category. Whatever semantic category may have been built up for an instance of *L′*, the *F* operator applies to produce a function from things of that type to things of type *B*, and the rule says that this function must be applied to the translation of *R′*. Clearly, if *R′* has exactly the same semantic category as *L′* this will succeed in yielding a *B*-type denotation for S, and under all other circumstances S will fail to be assigned a denotation.

The set of strings of category S that are assigned denotations under these rules is thus

{*xx* | *x* in (*a*, *b*)+}

which is a non-CF language. We know, therefore, that it is possible for semantic filtering of a set of syntactic rules to alter expressive power significantly. We know, in fact, that it would be possible to handle Bambara noun stems in this way and design a set of translation principles that would only allow a string "Noun-*o*-Noun" to be assigned a denotation if the two instances of *N* were stringwise identical. What we do not know is how to formulate with clarity a principle of linguistic theory that adjudicates on the question of whether the resultant description, with its infinite number of distinct semantic categories, is permissible. Despite the efforts of Barbara Hall Partee and other scholars who have written on constraining the Montague semantics framework over the past ten years, questions about permissible power in semantic apparatus are still not very well explored.

One thing that is clear is that Gazdar and others who have claimed or assumed that NL′s are context-free never intended to suggest that the entire mechanism of associating a sentence with a meaning could be carried out by a system equivalent to a pushdown automaton. Even if we take the notion "associating a sentence with a meaning" to be fully clear, which is granting a lot in the way of separating out pragmatic and discourse-related factors, it is obvious that operations beyond the power of a CF-PSG to define are involved. Things like identifying representations to which lambda-conversion can apply, determining whether all variables are bound, checking that every indexed anaphoric element has an antecedent with the same index, verifying that a structure contains no vacuous quantification, and so on, are obviously of non-CF character when regarded as language recognition problems. Indeed, in one case, that of disallowing vacuous quantifiers, it has been conjectured (Partee and Marsh 1984), though not yet proved, that even an indexed grammar does not have the requisite power.

It therefore should not be regarded as surprising that mechanisms devised to handle the sort of tasks involved in assigning meanings to sentences can come to the rescue in cases where a given syntactic framework has insufficient expressive power. Nor should it be surprising that those syntactic theories that build into the syntax a power that amply suffices to achieve a suitable syntax-to-semantics mapping have no trouble accommodating all new sets of facts that turn up. The moment we adopt any mechanisms with greater than, say, context-free power, our problem is that we are faced with a multiplicity of ways to handle almost any descriptive problem.

7. GRAMMARS WITH INFINITE NONTERMINAL VOCABULARIES

Suppose we decide we want to reject the idea of allowing a souped-up semantic rule system do part of the job of defining the membership of the language. What syntactic options are reasonable ones, given the kind of non-context-free languages we think we might have to describe?

There is a large range of theories of grammar definable if we relax the standard requirement that the set *N* of nonterminal vocabulary of the grammar should be finite. Since a finite parser for such a grammar cannot contain an infinite list of nonterminals, if the infinite majority of the nonterminals are not to be useless symbols, the parser must be equipped with some way of parsing representations of nonterminals, i.e. to test arbitrary objects for membership in *N*. If the tests do not guarantee results in finite time, then clearly the device may be of Turing-machine power, and may define an undecidable language. Two particularly interesting types of grammar that do not have this property are the following:

> *Indexed grammars.* If members of N are built up using sequences of indices affixed to a members of a finite set of basic nonterminals, and rules in P are able to add or remove sequence-initial indices, attached to a given basic nonterminal, the expressive power achieved is that of the indexed grammars of Aho (1968). These 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 parsing problem is exponential.
>
> *Unification grammars.* If members of N have internal hierarchical structure and parsing operations are permitted to match hierarchical representations one with another globally to determine whether they *unify* (roughly, whether there is a minimal consistent representation that includes the distinctive properties of both), and if the number of parses for a given sentence is kept to a finite number by requiring that we do not have
>
> $$A \overset{*}{\Longrightarrow} A$$
>
> for any A, then the expressive power seems to be weakly equivalent to the grammars that Joan Bresnan and Ron Kaplan have developed under the name *lexical-functional grammar* (*LFG*; *see Bresnan*, *ed*., *1982*; *cf*. *also* the work of Martin Kay on unification grammars). The LFG languages include some non-indexed languages (Kelly Roach, unpublished work), and apparently have an NP-complete parsing problem (Ron Kaplan, personal communication).

Systems of this sort have an undeniable interest in connection with the study of natural language. Both theories of language structure and computational implementations of grammars can be usefully explored in such terms. My criticism of them would be that it seems to me that the expressive power of these systems is too extreme. Linguistically they are insufficiently restrictive, and computationally they are implausibly wasteful of resources. However, rather than attempt to support this vague prejudice with specific criticisms, I would prefer to use my space here to outline an alternative that seems to me extremely promising.

8. HEAD GRAMMARS AND NATURAL LANGUAGES

In his recent doctoral dissertation, Carl Pollard (1984) has given a detailed exposition and motivation for a class of grammars he terms head grammars. Roach (1984) has proved that the languages generated by head grammars constitute a full AFL, showing all the significant closure properties that characterize the class of CFL's. Head grammars have a greater expressive power, in terms of weak and strong generative capacity, than the CF-PSG's, but only to a very limited extent, as shown by some subtle and suprising results due to Roach (1984). For example, there is a head grammar for

$\{a^n b^n c^n a^n \mid n \geq 0\}$

but not for

$\{a^n b^n c^n d^n a^n \mid n \geq 0\}$

and there is a head grammar for

$\{ww \mid w \text{ is in } (a, b)^*\}$

but not for

$\{www \mid w \text{ is in } (a, b)^*\}$.

The time complexity of the recognition problem for head grammars is also known: a time bound proportional to the seventh power of the length of the input is sufficient to allow for recognition in the worst case on a deterministic Turing machine (Pollard 1984). This clearly places head grammars in the realm of tractable linguistic formalisms.

The extension Pollard makes in CF-PSG to obtain the head grammars 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 eight new "wrapping" operations to the standard concatenation operation on strings that a CF-PSG can define. For a given ordered pair <B,C> of headed strings there are twelve ways in which strings B and C can be combined to make a constituent A. I give here the descriptions of just two of them which I will use below:

LC1(B,C): concatenate C onto end of B; first argument (B) is head of the result. Mnemonic: Left Concatenation with 1st as new head.

LL2(B,C): wrap B around C, with head of B to the left of C; C is head of the result. Mnemonic: Left wrapping with head to the Right and 2nd as new head.

The full set of operations is given in the chart in figure 1.

A simple and linguistically motivated head grammar can be given for the Swiss German situation mentioned earlier. I will not deal with it here, because in the first place it would take considerable space, and in the second place it is very simple to read off the needed account from Pollard's (1984) treatment of the corresponding situation in Dutch, making the required change in the syntax of case-marking.

In the next section I apply head grammar to cases like that of Bambara noun reduplication.

9. THE RIDDLE OF REDUPLICATION

I have shown in section 6 that the set of Bambara complex nouns of the form "Noun—o-Noun" could be described using semantic filtering of a context-free grammar. Consider now how a head grammar could achieve a description of the same facts. Assume, to simplify the situation, just two noun stems in Bambara, represented here as a and b. The following head grammar generates the language {x o y | x, y are in (a, b)+}:

Figure 1: combinatory operations in head grammar

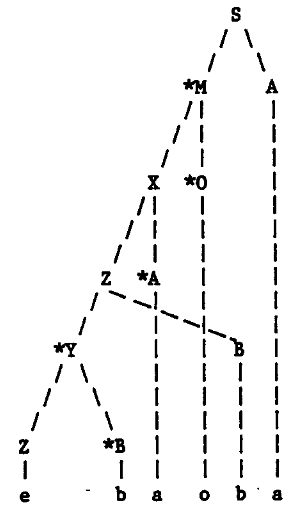
	LC1	LC2	RC1	RC2	LL1	LL2	LR1	LR2	RL1	RL2	RR1	RR2
Leftward or Rightward?	L	L	R	R	L	L	R	R	L	L	R	R
Concatenate, wrap Left, wrap Right?	C	C	C	C	L	L	L	L	R	R	R	R
1 or 2 is head of the result?	1	2	1	2	1	2	1	2	1	2	1	2

(7)

Syntax	Lexicon
S ---> LC1(M, A)	A ---> a
S ---> LC1(N, B)	B ---> b
M ---> LL2(X, O)	O ---> o
N ---> LL2(Y, O)	Z ---> e
X ---> LL2(Z, A)	
Y ---> LL2(Z, B)	
Z ---> LC1(X, A)	
Z ---> LC1(Y, B)	

The structure this grammar assigns to the string ba-o-ba is shown in figure 2 in the form of a tree with crossing branches, using asterisks to indicate heads (or strictly, nodes through which the path from a label to the head of its terminal string passes).

Figure 2: structure of the string "ba-o-ba" according to the gramar in (7).



We know, therefore, that there are at least two options available to us when we consider how a case like Bambara may be described in rigorous and computationally tractable terms: semantic filtering of a CF-PSG, or the use of head grammars. However, I would like to point to certain considerations suggesting that although both of these options are useful as existence proofs and mathematical benchmarks, neither is the right answer for the Bambara case. The semantic filtering account of Bambara complex nouns would imply that every complex noun stem in Bambara was of a different semantic category, for the encoding of the exact repetition of the terminal string of the noun stem would have to be in terms of a unique compositional structure. This seems inherent implausible; "dog-catcher-catcher-catcher" should have the same semantic category as "dog-catcher-catcher" (both should denote properties, I would assume). And the head grammar account of the same facts has two peculiarities. First, it predicts a peculiar structure of word-internal crossing syntactic dependencies (for example, that in dog-catcher-o-dog-catcher, one constituent is dog-dog and another is dog-catcher-o-dog) that seem unmotivated and counter-intuitive. Second, the grammar for the set of complex nouns is profligate in the sense of Pullum (1983): there are inherently and necessarily more nonterminals involved than terminals---and thus more different ad hoc syntactic categories than there are noun stems. Again, this seems abhorrent.

What is the correct description? My analytical intuition (which of course, I do not ask others to accept unquestioningly) is that we need a direct reference to the reduplication of the surface string, and this is missing in both accounts. Somehow I think the grammatical rules should reflect the notion "repeat the morpheme-string" directly, and by the same token the parsing process should directly recognize the reduplication of the noun stem rather than happen indirectly to guarantee it.

I even think there is evidence from English that offers support for such an idea. There is a construction illustrated by phrases like Tracy hit it and hit it and hit it. that was discussed by Browne (1964), an unpublished paper that is summarized by Lakoff and Peters (1969, 121-122, note 8). It involves reduplication of a constituent (here, a verb phrase). One of the curious features of this construction is that if the reduplicated phrase is an adjective phrase in the comparative degree, the expression of the comparative degree must be identical throughout, down to the morphological and phonological level:

(8)a. Kim got lonelier and lonelier and lonelier.
b. Kim got more and more and more lonely.
c. *Kim got lonelier and more lonely and lonelier.

This is a problem even under transformational conceptions of grammar, since at the levels where syntactic transformations apply, lonelier and more lonely are generally agreed to be indistinguishable. The symmetry must be preserved at the phonological level. I suggest that again a primitive syntactic operation "repeat the morpheme-string" is called for. I have no idea at this stage how it would be appropriate to formalize such an operation and give it a place in syntactic theory.

10. CONCLUSION

The arguments originally given at the start of the era of generative grammar were correct in their conclusion that NL's 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 NL's were CFL's were dismissed rather too hastily; English was never shown to be outside the class of CFL's or even the DCFL's (the latter question never even having been raised), and for other languages the first apparently valid arguments for non-CFL status are only now being framed. If we are going to employ supra-CFL mechanisms in the characterizing and processing of NL's, there are a host of items in the catalog for us to choose among. I have shown that semantic filtering is capable of enhancing the power of a CF-PSG, and so, in many different ways, is relaxing the finiteness condition on the nonterminal vocabulary. Both of these moves are likely to inflate expressive power quite dramatically, it seems to me. One of the most modest extensions of CF-PSG being explored is Pollard's head grammar, which has enough expressive power to handle the cases that seem likely to arise, but I have suggested that even so, it does not seem to be the right formalism to cover the case of the complex nouns in the lexicon of Bambara. Something different is needed, and it is not quite clear what.

This is a familiar situation in linguistics. Description of facts gets easier as the expressive power of one's mechanisms is enhanced, but choosing among alternatives, of course, get harder. What I would offer as a closing suggestion is that until we are able to encode different theoretical proposals (head grammar, string transformations, LFG, unification grammar, definite clause grammar, indexed grammars, semantic filtering) in a single, implemented, well-understood formalism, our efforts to be sure we have shown one proposal to be better than another will be, in Gerald Gazdar's scathing phrase, "about as sensible as claims to the effect that Turing machines which employ narrow grey tape are less powerful than ones employing wide orange tape" (1982, 131). In this connection, the aims of the PATR project at SRI International seem particularly helpful. If the designers of PATR can demonstrate that it has enough flexibility to encode rival descriptions of NL's like English, Bambara, Engenni, Dutch, Swedish, and Swiss German, and to do this in a neutral way, there may be some hope in the future (as there has not been in the past, as far as I can see) of evaluating alternative linguistic theories and descriptions as rigorously as computer scientists evaluate alternative sorting algorithms or LISP implementations.

REFERENCES

Bermudez, Manuel (1984) Regular Lookahead and Lookback in LR Parsers. PhD thesis, University of California, Santa Cruz.

Bresnan, Joan W., ed. (1982) The Mental Representation of Grammatical Relations. MIT Press, Cambridge, MA.

Browne, Wayles (1964) "On adjectival comparison and reduplication in English." Unpublished paper.

Carlson, Greg (1983) "Marking constituents," in Frank Heny, ed., Linguistic Categories: Auxiliaries and Related Puzzles; vol. 1: Categories, 69-98. D. Reidel, Dordrecht.

Chomsky, Noam (1957) Syntactic Structures. Mouton, The Hague.

Church, Kenneth (1980) On Memory Limitations in Natural Language Processing. M.Sc. thesis, MIT. Published by Indiana University Linguistics Club, Bloomington IN.

Culy, Christopher (forthcoming) "The complexity of the vocabulary of Bambara."

Daly, R. T. (1974) Applications of the Mathematical Theory of Linguistics. Mouton, The Hague.

Gazdar, Gerald (1981a) "Unbounded dependencies and coordinate structure. Linguistic Inquiry 12, 155-184.

Gazdar, Gerald (1981b) "On syntactic categories." Philosophical Transactions of the Royal Society (Series B) 295, 267-283.

Gazdar, Gerald (1982) "Phrase structure grammar," in Jacobson and Pullum, eds., 131-186.

Gazdar, Gerald; Pullum, Geoffrey K.; and Sag, Ivan A. (1982) "Auxiliaries and related phenomena in a restrictive theory of grammar," Language 58, 591-638.

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

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

Jacobson, Pauline, and Pullum, Geoffrey K., eds. (1982) The Nature of Syntactic Representation. D. Reidel, Dordrecht, Holland.

Lakoff, George, and Peters, Stanley (1969) "Phrasal conjunction and symmetric predicates," in David A. Reibel and Sanford A. Schane, eds., Modern Studies in English. Prentice-Hall, Englewood Cliffs.

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

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

Langendoen, D. Terence, and Postal, Paul M. (1984) "English and the class of context-free languages," unpublished paper.

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

Marcus, Mitchell (1980) A Theory of Syntactic Recognition for Natural Language. MIT Press, Cambridge MA.

Manaster-Ramer, Alexis (1983) "The soft formal underbelly of theoretical syntax," in Papers from the Nineteenth Regional Meeting, Chicago Linguistic Society, Chicago IL.

Nozick, Robert (1974) Anarchy, State, and Utopia. Basic Books, New York.

Partee, Barbara, and William Marsh (1984) "How non-context-free is variable binding?" Presented at the Third West Coast Conference on Formal Linguistics, University of California, Santa Cruz.

Pereira, Fernando (1984) "A new characterization of attachment preferences," in D. R. Dowty, L. Karttunen, and A. M. Zwicky, eds., Natural Language Processing: Psycholinguistic, Computational and Theoretical Perspectives. Cambridge University Press, New York NY.

Pollard, Carl J. (1984) Generalized Phrase Structure Grammars, Head Grammars, and Natural Languages. Ph.D. thesis, Stanford University.

Pullum, Geoffrey K. (1979) Rule Interaction and the Organization of a Grammar. Garland, New York.

Pullum, Geoffrey K. (1983) "Context-freeness and the computer processing of human languages," in 21st Annual Meeting of the Assocation for Computational Linguistics: Proceedings of the Conference, 1-6. ACL, Menlo Park CA.

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

Rich, Elaine (1983) Artificial Intelligence. McGraw-Hill, New York NY.

Roach, Kelly (1984) "Formal properties of head grammars." Unpublished paper, Xerox Palo Alto Research Center, Palo Alto CA.

Sag, Ivan A. (1982) "A semantic analysis of ´NP-movement´ dependencies in English." In Jacobson and Pullum, eds., 427-466.

Shieber, Stuart (1983) "Evidence against the context-freeness of natural language." Unpublished paper. SRI International, Menlo Park CA, and Center for the Study of Language and Information, Stanford CA.

Shieber, Stuart (1983) "Sentence disambiguation by a shift-reduce parsing technique," in 21st Annual Meeting of the Assocation for Computational Linguistics: Proceedings of the Conference, 113-118. ACL, Menlo Park CA.

Thomas, E. (1978) A Grammatical Description of the Engenni Language. SIL Publication no. 60. Summer Institute of Linguistics, Arlington TX.