Parsing Without (Much) Phrase Structure

Michael B. Kac
Department of Linguistics
University of Minnesota
Minneapolis, MN 55455
USA

Alexis Manaster-Ramer
Program in Linguistics
University of Michigan
Ann Arbor, MI 48109
USA

Approaches to NL syntax conform in varying degrees to the older relational/dependency model, (essentially that assumed in traditional grammar), which treats a sentence as a group of words united by various relations, and the newer constituent model. Some modern approaches have nonetheless involved shifts away from essentially constituent-based models of the sort associated with Bloomfield and Chomsky to more relation-based ones (e.g. case grammar, relational grammar, daughter-dependency and word grammar, corepresentational grammar) while some others, notably lexical-functional grammar, have nonetheless continued to rely crucially on certain techniques inherited from constituency-based grammar, particularly context-free grammar. In computational linguistics there is a strong (if not universal) reliance on phrase structure as the medium via which to represent syntactic structure; call this the CONSENSUS VIEW. A significant amount of effort has accordingly been invested in techniques by which to build such a representation efficiently, which has in turn led to considerable work on the formal and computational properties of context-free gramamrs (or natural extensions of them) and of the associated languages. In its strongest form, the consensus view says that the recovery of a fully specified parse tree is an essential step in computational language processing, and would, if correct, provide important support for the constituent model. In this paper, we shall critically examine the rationale for this view, and will sketch (informally) an alternative view which we find more defensible. The actual position we shall take for this discussion, however, is conservative in that we will not argue that there is no place whatever for constituent analysis in parsing or in syntactic analysis generally. What we WILL argue is that phrase structure is at least partly redundant in that a direct leap to the composition of some semantic units is possible from a relatively underspecified syntactic representation (as opposed to a complete parse tree). However, see Rindflesch forthcoming for an approach to parsing which entails a much stronger denial of the consensus view.

The rationale for the consensus view consists of four main propositions: (i) phrase structure analysis is well entrenched in both 'pure' and computational linguistics; (ii) phrase structure grammars are well understood mathematically; (iii) context-free languages are provably computationally tractable; (iv) semantic processing is either impossible, or at best highly nonoptimal, without a complete parse tree to work with (with the possible qualification that syntactic and semantic processing might be interleaved). We will focus on (ii-iv), (i) amounting to nothing more than the identification as such of the consensus view.

Argument (ii) is compelling if, but only if, one is willing to grant certain other assumptions. Since these include the points at issue, namely that phrase structure analysis is in principle adequate to the task at hand, the argument is circular taken by itself. With regard to (iii), note that even if NL's (or large tracts of them) are context-free, that is SUFFICIENT to assure that they are computationally tractable, but not NECESSARY. That is, the tractability of a language or sublanguage implies nothing with regard to context-freeness.[1]

Argument (iv) amounts to saying that the composition of a given semantic unit can be identified only after the corresponding syntactic constituent has been parsed, but this is false. It is possible, both in principle and in fact, to recognize at least some semantic units by operating on an 'impoverished' syntactic representation, i.e. one which does not yet incorporate any information about the syntactic constituents corresponding to the units in question. The following sentences are offered by way of illustration: [1. *John likes Mary* 2. *Mary , John likes* 3. *I think John likes Mary* 4. *Mary, I think John likes*] In these examples, where all the NP's are single words, it is a trivial matter to assign each to one of the following schemata: [1'. NP_1 P NP_2 2'. NP_1 NP_2 P 3'. NP_1 P_1 NP_2 P_2 NP_3 4'. NP_1 NP_2 P_1 NP_3 P_2] The goal in all four cases is to identify a nonlexical predicate consisting of *likes* and *Mary* and a predication consisting of *John* and the afore-mentioned nonlexical predicate. In 3-4, this predication must also be analyzed as a component of a larger one. Under the consensus view, this would require identification of constituents of the categories VP or VP/NP prior to recognition of nonlexical predicates, and the identification of constituents of the categories S or S/NP prior to the recognition of predications. But given just the amount of structure in the schemata shown in 1'-4', we can proceed directly to the semantic units as follows. Assuming that processing starts at the left: (a) in a sequence of the form NP_1 NP_2 P, leave NP_2 unlabelled; (b) in a sequence of the form NP P, label the NP as Subject of the P; (c) if no NP appears to the right of a P requiring an NP Object, associate this function with the nearest unlabelled NP to the left.

We illustrate with 4. In either case, at the conclusion of the first pass, the predication corresponding to the subordinate clause is fully specified and at least the Subject of the predication corresponding to the main clause is identified. On the second pass, it suffices to search for P's requiring Object complements and to assign this function to any predication whose own P lies to the right of such a predicate. (Discontinuity poses no difficulties, nor is it necessary to make use of auxiliary devices such as empty categories to mark the positions of syntactic gaps.) Further, once a transitive P and its Object have been identified, these may be composed into a larger intransitive predicate.

A second instructive example is provided by the problematical Dutch constructions discussed in Bresnan et al. 1982. The problem, briefly put, is that there is a class of VP's in Dutch which take the form NP^{n-1} V^n but which cannot, apparently, be assigned a center-embedding constituent structure. Using a lexical-functional framework, the authors show that constraints on f-structure can be used as a filter on c-structure which are generable by the (context-free) phrase structure component of the grammar. If one applies this conception seriously to parsing, then it follows that what the parser must construct is functionally annotated parse trees, and yet it is not difficult to see how the functional information could be used, much as it was in the earlier example, to bypass at least some of the steps involved in constructing a c-structure. As an example, consider ... *dat Jan Piet Marie zag helpen zwemmen* 'that John saw Piet help Marie to swim' . One way to look at the problem would be this: imagine that there is a recursive way of constructing complex verbs out of simple verbs such that the complex inherits the arguments of the simplexes, and that the arguments of the complex must appear in a linear order corresponding to the order of the simplexes with which they are associated.   Imagine further that it is possible to

have rules like [5. VP -> V" V; 6. V" -> NP^n V' (UP n OBJ DOWN)] Given a string of Object NP's, we would have each of them bear a different relation to the complex verb: the leftmost would be 1OBJ, the next leftmost 2OBJ, etc. There is now no difficulty coming up with a way to capture the generalization that 1OBJ is the OBJ of the first simplex verb, 2OBJ the OBJ of the second and so on. In regard to parsing, we can now see that as long as there is a way to build up a complex V (we maintain a neutral stance as to how that might be done), then the composition of the semantic unit corresponding to the VP referred to in rule 5--and the relations which obtain within it--can be recovered without actually building the VP constituent of the c-structure. As long as there is a way, somehow, to build up as much structure as is represented in the schema NP NP NP [$_{V'}$ V V] V then the following will yield the desired results: (a) leave the initial NP unlabelled on the first pass; (b) for all $n \geq 2$, label the n th NP n -1OBJ of V$_n$.

In the example under discussion, this will make *Piet* and *Marie* respectively 1OBJ and 2OBJ of the V' *zag-helpen*. The entire predicate can then be identified by composing the rightmost V with the expression consisting of the V' and its arguments; by the same token, the pairings of arguments of the V' with the appropriate daughter V's is easily accomplished. The end result is the recognition of all the f-structures which have to be extracted from the string without prior recognition of either the V" or VP constituents referred to in the rules (5-6).

Our examples are simplified in one respect, namely that they involve no NP's longer than a single word. It is possible that something more like phrase structure analysis is required to handle such units (as well as the V' referred to in the analysis of the Dutch example), though Rindflesch (forthcoming) argues that this is not the case. (See also Hudson 1976, 1984.)

Up to this point, we have been concerned with showing that the case FOR the consensus view is not especially compelling; we now proceed to the arguments AGAINST it. The illustration just given actually amounts to an argument against since it shows that the S- or S/NP - and VP or VP/-NP constituents of a parse tree are inessential to cue the recognition of predications and nonlexical predicates.

The arguments up to this point have been concerned with the output of a syntactic parser; it needs to be noted as well that there are some difficulties associated with the idea that a parser operates with a set of phrase structure rules, or formally similar objects. In Kac et al. 1986, it is argued that there are advantages to parsing in a series of graded stages such that at each stage only a particular, restricted type of structural information (e.g. information about relations of subordination among verbs) is being sought. A variety of different types of information are 'compacted' into phrase structure grammars in a way which makes ·it difficult to isolate a given type and operate with it independently of the others. While there is nothing in principle to prevent this information from being extracted from a set of PS-rules, the overhead imposed by the interpretation process makes this an unattractive option. A preferable stragegy would be to have a highly structured grammar for the parser to refer to, with a hierarchy of different types of information corresponding to the various phases via which the entire structural representation is built up.

We offer one last example which suggests strongly that phrase structure analysis is problematical in some cases. Consider the coordinate sentence *John likes and Bill hates beans*. One immediate observation that we can make is that the sequence *Bill hates beans* would, in isolation, be a sentence, which might in turn lead us to an analysis which, whatever else it might entail, would treat the material to the right of *and* as an S, coordinated by the *and* to some constituent occurring to the left of the

conjunction. An obvious difficulty which stands in the way of this conclusion is that there does not appear prima facie to be any way to treat anything to the left of the *and* as an S, thereby violating the widely assumed principle that only like constituents can be coordinated (the principle of 'categorial harmony'). Four alternatives thus present themselves: abandon the analysis in favor of one in which the right-conjunct belongs to a category other than S; abandon the principle of categorial harmony; modify the principle of categorial harmony; find some way of analyzing the material to the left of *and* as an S.

The first alternative looks initially most attractive, especially when seen in the light of the approach to categories originally proposed by Gazdar (1982) and other expositions of GPSG. We could thus analyze the example as having the structure [$_S$[$_{S/NP}$ [$_{S/NP}$ John likes] and [$_{S/NP}$ Bill hates] beans]. Part of the justification for this analysis is the presence of an intonation break directly after *hates* that is not present when *Bill hates beans* is present in isolation. This move, however, creates two new problems. First of all, it involves a possibly unwarranted intrusion of phonology into syntax. It is one thing to argue that a phrase structure analysis with purely syntactic motivation serves as an accurate predictor of where intonation breaks will fall, quite another to let the phrase structure analysis be dictated by phonological considerations (in which case the predictions are self-fulfilling). There is a more serious difficulty, however, namely that while there is indeed a break after *hates* , it is not the major break (which comes directly after *likes*) despite the fact that the analysis places the major syntactic boundary at this point. Full consistency with the phonological facts would require a syntactic analysis like [$_S$[$_{S/NP}$ John likes] and [$_S$ [Bill hates] beans]] We would then run into problems with the categories, however, since we would again have coordination of unlike constituents. Note, moreover, that it would not be possible to subsume S and S/NP by an 'archicategory' (Sag et al. 1985) since the GPSG treatment of coordinability depends crucially on the categorical impossibility of coordinating X with X/Y (Gazdar 1981).

What we have said so far should be enough to make it clear that finding a way to analyze an example like the one under discussion in phrase structure terms is not as straightforward a matter as it might first have appeared to be. It is conceivable that ways can be found around the difficulties we have mentioned, though one might reasonably ask whether the effort would be of genuine interest or whether it would be more in the nature of a holding action. It is, in any case, possible to handle examples like the ones under discussion in a straightforward manner without attempting a phrase structure analysis (Kac 1985).

Summary:
  1. The rationale for phrase structure analysis is uncompelling on both computational and linguistic grounds.

  2. A fully specified parse tree is partially redundant insofar as structural cues for the recovery of semantic information are concerned.

  3. Phrase structure rules and allied formalisms do not provide the optimal way of representing the grammatical information on which a parser depends.

  4. Phrase structure analysis is problematical in certain cases.

These facts imply that alternatives to the consensus view deserve to be investigated.

Note

1. There is a deeper difficulty here, namely the presumption that NL's must be computationally tractable. There is, to our knowledge, no evidence that this is the case. While it is undeniable that humans parse rapidly and effortlessly most of the time, nothing follows from this fact regarding the computational properties of any NL taken as a whole. At most, it shows an understandable predisposition to communicate via easily parsed structures.

References

Bresnan, J., R.M. Kaplan, S. Peters, and A. Zaenen. 1982. Cross-serial dependencies in Dutch. Linguistic Inquiry 13.613-635.

Gazdar, G. 1981. Unbounded dependencies and coordinate structure. Linguistic Inquiry 12.155-184.

--- 1982. Phrase structure grammar. In P. Jacobson and G.K. Pullum eds., The Nature of Syntactic Representation. Dordrecht: Reidel. 131-186.

Hudson, R.A. 1976. Arguments for a Nontransformational Grammar. Chicago and London: University of Chicago Press.

--- 1984. Word Grammar. Oxford: Basil Blackwell.

Kac, M.B. 1985. Constraints on predicate coordination. Bloomington, IN: Indiana University Linguistics Club.

---, T. Rindflesch and K.L. Ryan. 1986. Reconnaissance-attack Parsing. This volume.

Rindflesch, T. forthcoming. Doctoral dissertation in preparation, University of Minnesota.

Sag, I., G. Gazdar, T. Wasow and S. Weisler. 1985. Coordination and how to distinguish categories. Natural Language and Linguistic Theory 3.117-172.