

Unification and Some New Grammatical Formalisms

Aravind K. Joshi

Department of Computer and Information Science
University of Pennsylvania

I have organized my comments around some of the questions posed by the panel chair, Fernando Pereira.

The key idea in the unification-based approaches to grammar is that we deal with informational structures (called feature structures) which encode a variety of linguistic information (lexical, syntactic, semantic, discourse, perhaps even cross-linguistic) in a uniform way and then manipulate (combine) these structures by means of a few (one, if possible) well-defined operations (unification being the primary one). The feature structures consist of features and associated values, which can be atomic or complex i.e., feature structures themselves. In other words, the values can be from a structured set. The unification operation builds new structures and together with some string combining operation (concatenation being the primary one) pairs the feature structures with strings (Schieber, 1986).

How does the unification formalism differ from the standard context-free grammar formalism?

In a pure CFG one has only a finite number of nonterminals, which are the category symbols. In a CFG based grammar one associates with each category symbol a complex of features that are exploited by the grammar in a variety of ways. In the unification formalism there is really no such separation between the category symbols and the features. Feature structures are the only elements to deal with. Of course, the traditional category symbols show up as values of a feature (cat) in the feature structures. The notion of nonterminal symbol is flexible now. If we multiply out all features and their values down to the atomic values, we will have a very large number (even infinite under certain circumstances) of nonterminal symbols. Of course, this means trouble for parsing. Clearly, the standard parsing algorithms for parsing CFG's cannot be extended to unification formalism because of the exponential blowup of

computational complexity, including the possibility of nontermination. One could focus only on parts of feature structures, not necessarily the same parts for different feature structures, and thereby, have a flexible notion of nonterminal on the one hand and, perhaps, control the computational complexity on the other hand. This aspect of unification formalism has not received much attention yet, except in the very interesting work of Schieber (1985).

To what extent has the unification formalism been motivated by processing considerations?

First of all, we should distinguish at least two meanings of *processing considerations*. One has to do with the efficiency of computation and the other has to do with computational formalisms, which are well-defined and whose semantics (i.e., the semantics of the formalism) also can be well-defined. Although the unification formalism has been developed largely by researchers who are, no doubt, interested in the efficiency of computation, the primary motivation for the formalism has to do with the second meaning of *processing considerations*. The standard CFG based formalisms (augmented in a variety of ways) can do all the computations that a unification based formalism can do and vice-versa, however, the semantics of the formalism (not of the language described by the grammar) is not always well understood. The same is, of course, true of the ATN formalism. The unification formalism does give an opportunity to provide a well-defined semantics because of its algebraic characterization (Pereira and Schieber, 1984). How this understanding can be cashed into efficient algorithms for processing is still very much an open question. Good engineering is based on good theory - therein lies the hope.

Are we converging to some class of formalisms that are relevant to processing and, if so, how can this class be characterized in a theoretical manner?

Most of the grammatical formalisms, especially those of the so-called nontransformational flavor, have been motivated, at least in part, by processing considerations, for example, parsing complexity. We could say that these formalisms are converging if convergence is defined along several dimensions. GPSG, LFG, HG, HPSG¹ all have a context-free grammar explicitly or

¹CFG: context-free grammar, GPSG: generalized phrase structure grammar, LFG: lexical functional grammar, HG: head grammar, TAG: tree adjoining grammar, HPSG: head driven phrase structure grammar, FUG: functional unification grammar, CG: categorial grammar, PATR: parsing and translation

implicitly, use feature structures of some sort or another, and the lexicon. Unification formalism by itself is not a grammatical theory but a formalism in which different grammatical theories can be instantiated. Some of these grammatical theories explicitly incorporate unification formalism as one component of the grammar (e.g., GPSG, LFG, HPSG, FUG, PATR based grammars, etc.), while some others (e.g. TAG, HG, CG, etc.) do not explicitly incorporate unification formalism, as the feature checking component is not explicitly specified in these grammars as they are formulated at present. The unification formalism is a nice way of incorporating this feature checking component in these grammars, in fact, the string combining operations (in HG and CG) and the tree combining operation (in TAG) can themselves be formulated within the unification formalism generating feature structures in an appropriate manner. In fact, these different grammatical theories differ with respect to the *domain of locality* over which the *unifications* (a la Schieber), i.e., a set of constraints across a set of feature structures, are defined. For example, for a CFG based unification formalism, the domain of locality are the context-free rules, e.g., $X_0 \rightarrow X_1 X_2$. The *unifications* are defined over feature structures associated with X_0 , X_1 , and X_2 . For a tree adjoining grammar, the domain of locality are the elementary trees (structures, in the general case), both initial and auxiliary. These domains of locality define the unifications across the feature structures associated with the components of the domain, and thereby, determine how information flows among these feature structures. These domains also determine the kinds of word order patterns describable by these different grammatical formalisms. In this sense, all these grammatical formalisms could be said to converge. This is not surprising as the unification formalism is a very powerful formalism, in fact, equivalent to a Turing machine. As far as I can see, any reasonable grammatical formalism can be instantiated in the unification formalism, as it is unconstrained in the sense described above. The particular constraints come from the particular grammatical formalism that is being instantiated.

There is another sense of convergence we can talk about. Here we are concerned with the weak generative capacity, strong generative capacity, parsing complexity, and other formal language and automata theoretic properties. It appears that a proper subclass of indexed grammars with at least the following properties may characterize adequately a class of grammars suitable for describing natural language structures, a class called "mildly context-sensitive" in Joshi (1985), (MCG: mildly context-sensitive grammars, MCL: languages of MCGs). The properties are: 1) context-free languages are properly contained in MCL, 2) languages in MCL

can be parsed in polynomial time, 3) MCG's capture only certain kinds of dependencies, e.g. nested dependencies and certain limited kinds of crossing dependencies (e.g. in the subordinate clause constructions in Dutch, but not in the so-called MIX (or Bach) language, which consists of equal number of a's, b's, and c's in any order), and 4) languages in MCL have constant growth property, i.e., if the strings of the language are arranged in increasing order of length then any two consecutive lengths do not differ by arbitrarily large amounts, in fact, any given length can be described as a linear combination of a finite set of fixed lengths. These properties do not precisely define MCG, but rather give only a rough characterization, as the properties are only necessary conditions and further, some of the properties are properties of structural descriptions rather than the languages, hence, difficult to characterize precisely. TAG, HG, some restricted IG², and certain types of CG all appear to belong to this class. Moreover, certain equivalences have been established between these grammars, for example, between TAG and HG (Vijay-Shanker, Weir, and Joshi, 1986). Some natural extensions of TAG also seem to belong to this class. The processing implications of this convergence are not at all clear, because the polynomial time complexity, first of all, is only a worst case measure, and secondly, it has to be considered along with the constant of proportionality, which depends on the grammar.

Do processing considerations and results show that such systems when implemented can be neutral between analysis and production?

The pure unification formalism (i.e., with unification as the only operation and no non-monotonic aspects in the feature structures) is bidirectional, in the sense that the order in which unifications are performed does not matter. In this sense, they can be considered neutral between analysis and production. However, as soon as one adds operators that are not commutative or associative or add values to feature structure which exhibit non-monotonic behavior, we no longer have this bidirectionality (and also, perhaps, disallowing the possibility of giving well-defined semantics). The proponents of unification formalism hope to keep these amendments under control. How successfully this can be done is very much an open problem.

To the extent a formalism is declarative (and this applies equally well to the particular grammatical theories instantiated in a unification formalism) it can be neutral between analysis

²IG: indexed grammar

and production. The processes which manipulate these formalisms may or may not differ for analysis and production. Neutrality between analysis and production is a property shared by a variety of grammatical formalisms. This kind of neutrality is not the key selling point for unification formalism, in my judgement.

Is it a real advance or just a Hollywood term?

We have already stated the difference between a CFG based formalism using feature complexes in a variety of ways and the unification based formalis. A well-defined formalism whose mathematical properties (syntactic, semantic, and computational) are well understood is always an advance, even though some earlier theories may have used the same pieces of information in some informal manner. Clearly, before the advent of the CFG formalism, people had worked with related ideas (e.g., immediate constituent analysis, even part of Panini's grammar are in a CFG style!); however, no one would say that CFG is just a Hollywood term (or a Broadway term, given the location where CFG's were born). The mathematical and computational insights that CFG has provided has immensely helped linguistics as well as computational linguistics. The unification formalism shows similar possibilities although the mathematical or computational results are not yet at the level corresponding to the CFG formalism. So in this sense, it is not a Hollywood term, it is an advance. How big an advance? We will have to wait for an answer to this question until we know more about its mathematical and computational properties. Personally, I would like to see some results on some constrained unification formalisms, in the sense that the flow of information between feature structures is constrained in some systematic manner. Such results, if obtainable, could give us more insights into the computational properties of these formalisms and their suitability (not just their adequacy) for describing natural language structures.

Acknowledgements

This work is partially supported by DARPA grants NOOO14-85-K-0018 and NOOO14-85-K-0807, NSF grants MCS8219196-CER, MCS-82-07294, 1 RO1-HL-29985-01, U.S. Army grants DAA6-29-84-K-0061, DAAB07-84-K-F077, U.S. Air Force grant 82-NM-299, AI Center grants NSF-MCS-83-05221.

References

Joshi, A.K. 1985. Tree Adjoining Grammars: How much context sensitivity is required to provide reasonable structural descriptions? In *Natural Language Processing* (Eds. D. Dowty, L. Karttunen, and A. Zwicky), Cambridge University Press, Cambridge.

Pereira, F.C.N. and Schieber, S.M. 1984. "The semantics of grammar formalisms seen as computer languages". In *Proceedings of the Tenth International Conference on Computational Linguistics*, Stanford University, Stanford, CA, August.

Schieber, S.M. 1985. "Using restrictions to extend parsing algorithms for complex-feature-based formalisms". In *Proceedings of the 22nd Annual Meeting of the Association for Computational Linguistics*, University of Chicago, Chicago, Illinois, June.

Schieber, S.M. 1986. *An Introduction to Unification-Based Approaches to Grammar*. Center for the Study of Language and Information, Stanford University, Stanford, CA.

Vijay-Shanker, K., Weir, D., and Joshi, A.K. 1986. "Tree adjoining and head wrapping", in *Proceedings of the International Conference on Computational Linguistics (COLING)* Bonn, August 1986.