A PROPOSAL FOR MODIFICATIONS IN THE FORMALISM OF GPSG

James Kilbury
Universität Trier, FB II: LDV
Postfach 3825, D-5500 Trier
Fed. Rep. of Germany

ABSTRACT

Recent investigations show a remarkable convergence among contemporary unification-based formalisms for syntactic description. This convergence is now itself becoming an object of study, and there is an increasing recognition of the need for explicit characterizations of the properties that relate and distinguish similar grammar formalisms. The paper proposes a series of changes in the formalism of Generalized Phrase Structure Grammar that throw light on its relation to Functional Unification Grammar.

The essential contribution is a generalization of cooccurrence restrictions, which become the principal and unifying device of GPSG. Introducing Category Cooccurrence Restrictions (CCRs) for local trees (in analogy to Feature Cooccurrence Restrictions for categories) provides a genuine gain in expressiveness for the formalism. Other devices, such as Feature Instantiation Principles and Linear Precedence Statements can be regarded as special cases of CCRs. The proposals lead to a modified notion of unification itself.

A PROPOSAL FOR MODIFICATIONS IN THE FORMALISM OF GPSG

Recent investigations show a remarkable convergence among contemporary unification-based formalisms for syntactic description (cf Shieber 1985). This convergence is now itself becoming an object of study, and there is an increasing recognition of the need for explicit characterizations of the properties that relate and distinguish similar grammar formalisms. For example, Shieber (1986) describes a compilation from Generalized Phrase Structure Grammar (GPSG; cf Gazdar et alii 1985, henceforth GKPS) to PATR-II. The compilation defines the semantics of GPSG by explicitly relating the two formalisms; at the same time, difficulties in specifying the compilation show that differences between the formalisms transcend variety in notation.

This paper is similar to Shieber's in its aim but differs in the approach. A series of changes in the formalism of GPSG will be proposed that make it look more like the "tool oriented" formalism of Functional Unification Grammar (FUG; cf Kay 1984 and Shieber 1985). This notational transformation has two consequences: the essential and nonessential differences between GPSG and FUG can be made more apparent, and the internal structure of GPSG itself becomes more homogeneous and transparent.

The homogeneity of a formalism is desirable on methodological grounds that amount to Occam's principle of economy: entities should not be multiplied. This is not to suggest that linguistic formalisms can be simplified at our will; on the contrary, they must be complex and expressive enough to capture the complexities inherent in language itself. The burden of proof, however, falls on those who choose more complicated and heterogeneous notational devices.

Despite its restrictiveness in comparison with current transformational theory, GPSG in the GKPS version offers a rich palette of formal devices. It introduces Feature Cooccurrence Restrictions (FCRs) to state Boolean restrictions on the cooccurrence of feature specifications within categories but does not explore the use of analogous restrictions in other parts of the formalism. Immediate Dominance rules, metarules, and lexical rules are clearly distinguished in their form but all serve to capture the phenomenon of subcategorization.

This paper proposes the extension of cooccurrence restrictions in GPSG to express constraints on the cooccurrence of categories within local trees. While presented in Kilbury (1986) as a new descriptive device, such Category Cooccurrence Restrictions (CCRs) are in fact simply a generalization of principles fundamental to GKPS.

The motivation for CCRs is analogous to that for distinguishing Immediate Dominance (ID) and Linear Precedence (LP) rules in GPSG (cf GKPS, pp. 44-50). A context free rule binds information of two kinds in a single statement. By separating this information in ID and LP rules, GPSG is able to state generalizations of the sort "A precedes B in every local tree which contains both as daughters," which cannot be captured in a context free grammar.

Just as ID and LP rules capture generalizations about sets of context free rules (or equivalently, about local trees), CCRs can be seen as stating more abstract generalizations about ID rules, which in turn are equivalent to generalizations of the following sort about local trees:

(1) Any local tree with S as its root must have A as a daughter.

(2) No local tree with C as a daughter also has D as a daughter.

We can state CCRs as expressions of first order predicate logic using two primitive predicates, R(α, t) 'α is the root of local tree t' and D(α, t) 'α is a daughter in local tree t'.

Advantages of CCRs are discussed in Kilbury (1986): The metarules of GPSG can be eliminated as an extra device of the formalism. As noted above, generalizations can be captured that elude the expressive capabilities of GPSG. Moreover, CCRs render the GPSG formalism more homogeneous and establish a parallelism that can be expressed in the traditional notation of an analogy:

(3) FCR : category :: CCR : local tree

Linguistic items (categories and local trees) and restrictions on such items make up the terms of the above analogy. GPSG chooses to represent the items and restrictions as different kinds of object, whereas FUG has only one kind of object, the functional description (FD), which Kay (1984: 76) defines as "a Boolean expression over features" [i.e. GPSG feature specifications]. Thus, a homogeneous formalism for GPSG is easily achieved: just like cooccurrence restrictions, linguistic items can be represented as Boolean expressions, namely, as conjunctions of atomic assertions.

We shall henceforth regard a GPSG category as a conjunction of assertions about the values assigned to features [i.e. FUG attributes]; the assertions assigning these values constitute feature specifications. Unlike FUG, which always allows more information to be added to FDs and hence has no notion of a complete description, GPSG has fully specified categories in which every feature possible for the category is assigned a value. Excluding certain extensions to GPSG for non-context-free phenomena (cf Gazdar and Pullum 1985), GPSG allows only a finite number of categories for a language, while FUG permits infinitely many FDs. Like FDs, GPSG categories do not have a fixed term structure, but this property is nonessential for GPSG while being essential for FUG. It may be added that the modifications to GPSG proposed here leave it nonfunctional in Kay's sense.

FUG as described in Kay (1984) provides for conjunction and disjunction but not for negation in FDs. Karttunen (1984), however, argues for the use of both disjunction and negation in unification-grammar formalisms. GPSG has the full set of logical connectives in FCRs, which are arbitrary Boolean conditions on the cooccurrence of feature specifications within categories; categories themselves, however, are restricted in form to conjunctions of feature specifications. If the formal distinction of GPSG between linguistic items and linguistic restrictions is abandoned in favor of a uniform representation for both as Boolean expressions, we then can in effect use disjunction and negation in the categories as well. Conversely, we may view FCRs as partially instantiated categories, and CCRs correspondingly as partially instantiated local trees.

All Boolean expressions can be written in conjunctive normal form (CNF), i.e. as a conjunction of disjunctions of literals (positive or negated atomic expressions). Expressions in CNF are in turn equivalent to clause sets, i.e. sets of such disjunctions. Given this uniform representation for linguistic items and grammatical statements, it should come as no surprise to see unification, the principal operation of unification grammar, be closely identified with resolution as introduced by Robinson (1965) for automatic theorem proving. Nevertheless, no previous version of unification grammar has to my knowledge taken just this step.

The proposed operation differs somewhat from resolution. While the resolution of the clause sets $\{P\}$ and $\{\sim P \vee Q\}$ yields the resolvent $\{Q\}$, their unification in this sense produces $\{P, Q\}$. Some examples of such resolution-based unification will be useful at this point:

(4) $C_1 = \{f_1{:}v_1, (\sim f_2{:}v_2 \vee f_3{:}v_3)\}$
$C_2 = \{f_2{:}v_2\}$
$C_3 = \{f_3{:}v_3\}$
$C_4 = \{f_2{:}v_2 \vee f_4{:}v_4\}$
$C_5 = \{f_2{:}v_4\}$

$C_1 \sqcup C_2 = \{f_1{:}v_1, f_2{:}v_2, (\sim \text{true} \vee f_3{:}v_3)\}$
$= \{f_1{:}v_1, f_2{:}v_2, f_3{:}v_3\}$

$C_1 \sqcup C_3 = \{f_1{:}v_1, f_3{:}v_3, (\sim f_2{:}v_2 \vee \text{true})\}$
$= \{f_1{:}v_1, f_3{:}v_3\}$

$C_1 \sqcup C_4 = \{f_1{:}v_1, (f_3{:}v_3 \vee f_4{:}v_4)\}$

Note that for any two atomic values a_1 and a_2, the unification $a_1 \sqcup a_2$ succeeds iff $a_1 = a_2$. Given (4) above, if $v_2 \sqcup v_4$ succeeds (whether v_2 and v_4 are atomic or complex), then the unification $C_2 \sqcup C_5 = \{f_2{:}\ (v_2 \sqcup v_4)\}$ succeeds; if $v_2 \sqcup v_4$ fails, then $C_2 \sqcup C_5$ also fails. The unification $C_1 \sqcup C_5$ has three cases:

(5) $C_1 \sqcup C_5 = \{f_1{:}v_1, f_2{:}v_4, (\sim \text{true} \vee f_3{:}v_3)\}$
$= \{f_1{:}v_1, f_2{:}v_4, f_3{:}v_3\}$
if $v_2 \sqcup v_4$ succeeds and v_4 is an *extension* of v_2

$C_1 \sqcup C_5 = \{f_1{:}v_1, f_2{:}v_4, (\sim f_2{:}v_2 \vee f_3{:}v_3)\}$
if $v_2 \sqcup v_4$ succeeds and v_4 is *not* an extension of v_2

$$C_1 \sqcup C_5 = \{f_1{:}v_1,\ f_2{:}v_4,\ (\sim \text{false} \vee f_3{:}v_3)\}$$
$$= \{f_1{:}v_1,\ f_2{:}v_4\}$$
if $v_2 \sqcup v_4$ fails

FUG employs two special values, ANY and NONE, which unify with any and no other value, respectively. With the adoption of negation in the formalism, ANY and NONE emerge in the following dual relationship:

(6) $\sim$ f: ANY $\equiv$ f: NONE (Def.)

$\sim$ f: NONE $\equiv$ f: ANY (Def.)

ANY and NONE may be used in GPSG to express the condition that a feature must or may not receive a value. Shieber (1985: 32) notes that ANY constitutes a nonmonotonic device in the formalism, since final representations must not contain occurrences of ANY. In our terms, final representations must not contain negation or disjunction, i.e., they must be sets of unit clauses, each of which is a nonnegated literal. Since the logic upon which this formalism is based is monotonic, however, the essential monotonicity of the formalism is preserved.

GPSG goes a step further and introduces Feature Specification Defaults (FSDs), which are a patently nonmonotonic device based on default logic. This paper proposes banning them from the formalism for the time being. Some of the particular FSDs formulated in GKPS for English appear questionable under different analyses (cf Kilbury 1986). This is not to deny that default statements may capture significant generalizations about language. But why, then, should defaults be confined to the statement of restrictions on categories? It may be methodologically advantageous to first develop a more homogeneous and coherent formalism for GPSG without strongly nonmonotonic devices. If default logic later still appears desirable on theoretical linguistic grounds, then it can be reintroduced in a more principled fashion allowing default statements at all levels of linguistic description where it is useful.

The position of Linear Precedence (LP) statements in this formalism must now be clarified. It was stated above that CCRs are formulated using the two primitive predicates R(α, t) 'α is the root of local tree t' and D(α, t) 'α is a daughter in local tree t'. This is not quite adequate since different daughters in a local tree may be tokens of the same category. Let us replace D(α, t) with D(α, i, t), interpreted as 'α is the i-th daughter in local tree t'. A local tree t with VP as root and V, NP, and NP as daughters (in that order) can now be represented with the following set of unit clauses:

(7) $\{$R(VP,t), D(V,1,t), D(NP,2,t), D(NP,3,t)$\}$

Likewise, the LP statement $\alpha < \beta$ 'ß may not precede α in any local tree t' (where '<' denotes the LP relationship) may be reformulated in a logical expression (using '<' for arithmetic comparison) as follows:

(8) $\forall t$: (D(α, i, t) $\wedge$ D(ß, j, t)) $\supset$ i<j

This, in turn, can be represented as a set containing one clause:

(9) $\{(\sim D(\alpha, i, t) \vee \sim D(\beta, j, t) \vee (i<j))\}$

If arithmetic comparison '<' is now added to the primitive predicates allowed in CCRs, then LP statements become simply a special case of CCRs; they are applied to local trees by resolution-based unification with the representations of the latter.

The principle of cooccurrence restrictions can be further generalized in a final step. GPSG describes linguistic items and their distributions. Local trees are arrangements of categories, which in turn are arrangements of feature specifications; the latter are themselves items consisting of a feature name and a feature value in an arrangement. The formal devices already introduced allow us to state cooccurrence restrictions governing the combination of features and values in feature specifications; the definition of the value range of a feature can thus be regarded as another special case of cooccurrence restriction.

In summary, the essential contribution of this paper lies in its generalization of the notion of cooccurrence restriction. Many of the distinct formal devices of GPSG as presented in GKPS can be eliminated without an apparent loss of expressive power, and the resulting formalism gains both in simplicity and homogeneity while preserving essential properties of the GKPS formalism. Likewise, the uniform representation of cooccurrence restrictions and linguistic items allows a new interpretation of unification which is promising in its own right and which should facilitate the comparison of GPSG with other unification-based grammar formalisms. Parallels to other linguistic approaches, both more and less distant, should be evident. Similarities to American structuralism are neither accidental nor unintentional. In regard to his own proposals for unification, Karttunen (1984: 31) remarks that "the problems that arise in this connection are very similar to those that come up in logic programming." Indeed, many questions involving the equivalence of notations and of computational problems are raised that must be addressed in future studies.

REFERENCES

Gazdar, G. / E. Klein /G. Pullum / I. Sag (1985): *Generalized Phrase Structure Grammar.* Blackwell: Oxford.

Gazdar, G. / G. K. Pullum (1985): "Computationally Relevant Properties of Natural Languages and their Grammars," *New Generation Computing* 3: 273-306.

Karttunen, L. (1984): "Features and Values," *Proceedings of COLING 84*, 28-33.

Kay, M. (1984): "Functional Unification Grammar: A Formalism for Machine Translation," *Proceedings of COLING 84*, 75-78.

Kilbury, J. (1986): "Category Cooccurrence Restrictions and the Elimination of Metarules," *Proceedings of COLING 86*, 50-55.

Robinson, J. A. (1965): "A Machine Oriented Logic Based on the Resolution Principle," *Journal of the ACM* 25: 23-41.

Shieber, S. M. (1985): *An Introduction to Unification-Based Approaches to Grammar.* CSLI: Stanford, California.

Shieber, S. M. (1986): "A Simple Reconstruction of GPSG," *Proceedings of COLING 86*, 211-215.