THE NEW GRAMMER FORMALISMS - A TUTORIAL SURVEY

(Abstract)

Gerald Gazdar

School of Cognitive Sciences, university of Sussex,
Brighton BN1 9ON, United Kingdom

Take two or three upper case letters and glan them together with a "G" on the end, and, presto, you have the name for a new grammar formalism. Or, at least, so it must seem in a decade that has seen the arrival of COG, *CUG,* DOG, COG, FUG, GPsG, HPSG, JPSG, LF*G,* MOG, MLS, RGPSG, UOG, and XG. But the plethora of acronyms (and nonacronyms - PATR H), disguises much commonality, shared properties which set them apart from the mainstream linguistic and computational linguistic activity of the 1970s. The key characteristics of the new grammar formalisms, characteristics exhibited by most of them, are (i) the availability, at least in principle and sometimes in practice, of a declarative semantics for the notation used, (ii) the use of a basically type 2 rule format (single mother, unordered, no explicit context sensitivity) under (iii) a node admissibility rather than a string rewriting interpretation, with (iv) a recursively defined tree or directed acyclic graph based category set, and (v) unification as the primary operation for combining syntactic information. It is primarily the shared properties, rather than the myriad differences of notation, that this survey focusses on, although some of the more interesting differences between them emerge from the historical account of cxnceptual links and antecedents.

Declarative grammar formalisms were not in vogue in the 1970s. In linguistics, the dominant transformational tradition had essentially ceased to use any rule formalism at all in research work (though the textbooks of the period still provided expositions of the rule formalisms used in 1960s research). All that you will see in papers of the time is a widely used informal notation for structural descriptions and a taxonomy of syntactic constructions whose taxa have the names of the grammatical rules proposed by linguists in the 1960s - Raising, Gapping, Egui, Passive, and so on. In natural language processing (NIP), grammars and grammatical rules were equally hard to find. The augmented transition network (AIM) apparatus provided a paradigm for designing programs in which grammar, parser and, often, semantics were inextricably intertwined. Even PIDGIN is far closer to a procedural language for programming Marcus-parsers than it is to a declarative grammar formalism.

The inspirations for the grammar formalisms of the 1980s are manifold. Within linguistics, many of those who had been profoundly impressed by the technical rigour and formal elegance of Montague's work on semantics began to react against the informal but still baroque character of mainstream 1970s work on syntax. In search of simplicity and precision, they looked again at pretransfonnational grammar formalisms such as dependency grammar, phrase structure grammar, and categorial grammar (the latter having been used by Montague in his fragments of English syntax). As a result we see the appearance of Dependency unification Grammar (DUG), Generalized phrase Structure Grammar (GPSG), Combinatory Categorial Grammar (OOG) and their various descendants and hybrids such as HPSG (Head-driven Fhrase Structure Grammar), JPSG (used by the NIP group at IOOT), R-GPSG (from MTT), CUG (Categorial Unification Grammar), MOG (Meta-Categorial Grammar) and UOG (Unification Categorial Grammar).

Within the NIP community, two developments, one European, the other North American, began to converge. The first of these was Prolog. Computational linguistic considerations provided the primary motivation for Colmerauer's design of the language and his Metamorphosis Grammars are thus much more intimately related to Prolog than AINs are to Lisp, say. Metamorphosis Grammar was to be the first of a whole series of logic grammar formalisms, a series that has included extraposition Grammar (XG), Gapping Grammar, Slot Grammar, and Modular Logic Grammar (MUG). Undoubtedly the most influential member of this *group,* however, is the Definite Clause Grammar (DOG) notation. The convergent North American development was the appearance of Kay's Functional Grammar, subsequently renamed Functional unification Grammar (FUG) and the the appearance of the closely related Lexical Functional Grammar (LFG). FUG was directly influential, especially in its advocacy of unification, in many of the grammar types listed in the preceding paragraph. Lf*G,* like GPSG, was intended to embody empirical claims about the nature of all natural languages in the details of its formal design. By contrast, grammar formalisms like DOG, FUG and PAIR II (itself the progeny of DOG, FUG and GPSG) are not intended to make linguistic claims, but merely to provide linguistically natural and computationally tractable languages for writing grammars in. A recent trend has seen compilation from formalisms of the former type into formalisms of the latter type.