CATEGORIAL GRAMMARS FOR STRATA OF NON-CF LANGUAGES AND THEIR PARSERS

Michal P. Chytil

Charles University
Malostranské nám. 25
118 00 Praha 1
Czechoslovakia

Hans Karlgren

KVAL
Södermalmstorg 8
116 45 Stockholm
Sweden

Abstract

We introduce a generalization of categorial grammar extending its descriptive power, and a simple model of categorial grammar parser. Both tools can be adjusted to particular strata of languages via restricting grammatical or computational complexity.

I. Two questions about categorial grammars

In spite of the fascinating formal simplicity and lucidity of categorial grammar as developed by Bar-Hillel [1], Lambek [7] and followers, it has nevertheless never been brought into wide scale use. Why is this so?

We may easily recognize two drawbacks.

1/ Restricted scope of categorial grammars.

It was shown early [1] that the set of languages describable by these grammars is exactly that of context-free languages. Is this restriction inevitable or can a similar type of language description be retained beyond the limit of context-free languages? This is the first question we try to answer.

2/ No realistic model of categorial grammar parsing.

The schematic description of categorial analysis of a given sentence $a_1 \dots$ $\dots a_n$ is sketched in Fig. 1.

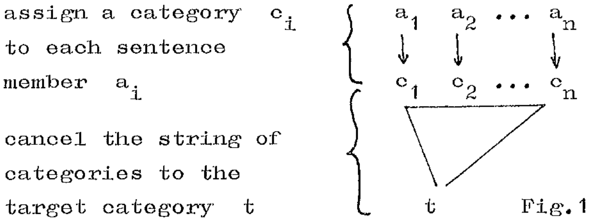
assign a category c_i to each sentence member a_i
a_1 a_2 ... a_n
c_1 c_2 ... c_n
cancel the string of categories to the target category t
t

Fig.1

This abstract scheme cannot serve as a description of a realistic parsing procedure. The suitable assignement appearing here as the first phase is in fact the goal of the parsing. The "brute force" approach following the above scheme, which checks all possible assignements and tries to cancel them is not computationally tractable, since for most grammars the number of all possible assignements grows exponentially with the length of the analysed sentence.

The moral of this observation is that the assignement cannot be separated from the cancellation. Similarly as parsers based on phrase - structure grammars have to make at each point of time an intelligent choice of rule to apply next, the categorial parser must make an intelligent choice out of a list of alternative categories. This necessity to look ahead at cancellation when making the assignement leads to the conclusion [6] that assignement and cancellation must in any actual parser be interwoven. Therefore our second key question reads:

Can this interweaving be grasped by a simple formal model or does it unavoidingly lead to a mess of complicated ad hoc and heuristic techniques?

II. Proposed solution

We introduce in nontechnical language the essence of the proposed generalization of categorial grammars and their parsers. The exact mathematical formulations can be found in [3].

Grammars. The principal difference between the "classical" categorial grammar and the generalized categorial grammar (GCG) is that instead of finite sets of categories corresponding to terminal symbols, GCG allows for *infinite sets* of categories. Each such infinite set, however, can be *generated by a simple procedure*, in fact a procedure based on a finite state generator.

Automata. We offer *list automaton* (LA) as a mathematical model of categorial grammar parsing. List automaton is schematically represented by Fig. 2.

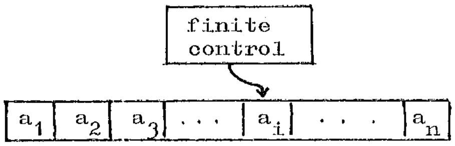

Fig. 2

LA consists of a nondeterministic finite state control unit attached to a finite tape. At the beginning of the computation, the tape contains the analysed string. The automaton can read and rewrite scanned symbols and move the scanning head one tape cell to the left or right analogously as Turing machine. In addition to it, it can *delete* the scanned cell, i.e. cut it out and paste the remaining tape parts together.

In the remainder of the paragraph we list results indicating, as we believe, that the concepts of GCG and LA give satisfactory answers to the above questions.

a/ *Scope and mutual correspondence*. Both GCGs and LA represent exactly *all context -sensitive languages*. Similarly like in the case of CF-grammars and pushdown automata or context-sensitive grammars and linearly bounded automata [5] there exist transformations of GCGs to LA and vice versa: an algorithm A_1, which for each GCG G yields a LA $A_1(G)$ representing the same language and conversely an algorithm A_2 which for each LA M yields an equivalent GCG $A_2(M)$.

The next step in our argument is to point out a remarkable feature of the interplay between GCGs and LA.

b/ *Stratification*. The correspondence between GCGs and LA can be observed not only in the whole class of context-sensitive languages, but also on the level of CF-languages and in each of infinitely many strata between CF a CS-languages. The stratification can be defined via two complexity measures.

Grammatical complexity: given a GCG G and a string w , the grammatical complexity of w wrt. G , denoted G(w) , is defined as the length of the longest category used in the analysis wrt. G . (For ambiguous grammars, the complexity is defined for each parse of the string).

Computational complexity: given a LA M and a string w , the computational complexity of w wrt., denoted M(w) , is defined as the maximal number of visits paid to a single square during the accepting computation (ambiguity being treated as before).

In the light of these complexity measures we can reconsider the relation between GCGs and LA determined by the above mentioned algorithms A_1 and A_2 For any GCG G and any sentence w , each grammatical description of w wrt. G is reflected as a computation of $A_1(G)$ accepting w . The grammatical complexity of the description is approximately the same as the computational complexity of the corresponding computation. Analogous result holds for A_2 .

Now, any function f mapping natural numbers on natural numbers determines a stratum S(f) of languages: a language L belongs to the *stratum S(f)* if and only if it can be represented by a GCG G (or equivalently a LA M) such that from each sentence w from L of length n , the complexity G(w) (or M(w)) does not exceed the number f(n) . Our previous considerations show that the algorithms A_1 , A_2 *respect the stratification*. Hence the introduced tools can be adjusted to the investigated languages.

Two examples :

1/ The grammars in the stratum S(const) (determined by constant functions) are exactly Bar-Hillel categorial grammars. "Finite visit" LA appear as their parsers.

2/ The languages in the strata S(f) , where f is any function of order smaller then the function log(log n) belong to "almost context-free languages" (cf. [2]) sharing crucial properties of CF-languages.

c/ Assignement and cancellation interwoven. To show that list automata, besides their simplicity, meet also the above formulated requirement for natural parsers of categorial grammar, we have to examine at least informally in more detail the relationship between a GCG G and its parser $A_1(G)$. When the automaton $A_1(G)$ analyses a string $a_1 \dots a_n$, then during the m-th visit to a square containing originally a symbol a_i , the automaton fixes the m-th symbol in the category belonging to a_i . Thus after m visits , m symbols of the category are determined. Therefore from the (infinite) set of categories assignable to to a_i , only those which agree with the determined symbols remain in play. To determine the next symbol of a category, the automaton can check the environment of the square and take into account possible cancellations. At the moment, when all symbols in a category are fixed, the corresponding square is deleted. In other words, a computation of $A_1(G)$ on a string $a_1 \dots a_n$ evolves dynamically a suitable assignement $c_1 \dots c_n$ of categories. The information used by the parser consists of

- generating mechanism of categories corresponding to particular symbols,
- indications of possible cancelling with neighbour categories.

The computation is completed at the moment when the assignement is found.

III. Open questions

1/ In this brief note we tried to grasp what features of the exact mathematical models described in [3] we consider to be fundamental. We can imagine alternative models differing in technical details but having the same features. Which of the models should be chosen as "canonical" will require more extensive studies.

2/ Our considerations deal with nondeterministic LA, i.e. in fact with "methods" of parsing. The step from "methods" to "algorithms" leads from nondeterministic to deterministic LA. Even a glimpse of the basic stratum S(const) promises interesting results. An observation of T. Hibbard [4] shows that deterministic "finite visit" LA represent a class of languages broader than the class of deterministic context-free languages. It implies that deterministic categorial grammar (in the classical sense) parsing will go beyond the limits of e.g. LR-parsing based on CF-grammars.

References

[1] Y. Bar-Hillel, C.Gaifman, F.Shamir: On categorial and phrase structure grammars, Bull.Res.Council Israel, F9, 1960

[2] M.P.Chytil: Almost context-free languages, to appear in Fundamenta Informaticae,1986

[3] M.P.Chytil, H.Karlgren: Categorial grammars and list automata for strata of non-CF languages, to appear in J.van Benthem, W.Buszkowski, W. Marciszewski (ed.), Categorial grammar, J. Benjamins R.V., Amsterdam - Philadelhia

[4] T.Hibbard: A generalization of context-free determinism, Information and Control 11 (1967), 196 - 238

[5] J.E.Hopcroft, J.D.Ullman: Formal Languages and their relation to automata, Add.-Wesley 1969

[6] H.Karlgren: Categorial grammar calculus, Scriptor, Stockholm 1974

[7] J. Lambek: On the calculus of syntactic types, in Structure of language and its math. aspects, Proc. 12th Symp.Appl. Math. AMS, Providence 1961