

"1965 International Conference on
Computational Linguistics"

SOME COMMENTS ON ALGORITHM
AND GRAMMAR IN THE AUTOMATIC PARSING
OF NATURAL LANGUAGES

Paul L. Garvin

The Bunker-Ramo Corporation
8433 Fallbrook Avenue
Canoga Park, California 91304



ABSTRACT

The purpose of this paper is to examine the oft-repeated assertion regarding the efficiency of a "simple parsing algorithm" combinable with a variety of different grammars written in the form of appropriate tables of rules. The paper raises the question of the increasing complexity of the tables when more than the most elementary natural-language conditions are included, as well as the question of the ordering of the rules within such nonelementary tables. Some concrete examples from the field of machine translation will be given in the final version of the paper. Some conclusions are presented.

1. The purpose of this paper is to examine the validity of the frequently repeated contention that the separation of algorithm and grammar is particularly desirable in automatic parsing programs. This examination will be restricted to the area of automatic parsing of natural languages with particular attention to the parsing problems encountered in machine translation.

It must be noted at the outset that in this author's opinion the aim of the automatic parsing component of a machine translation program is the adequate recognition of the boundaries and functions of syntactic units. On the basis of this recognition, automatic translation on a sentency-by-sentence rather than word-by-word basis can be effected.

2. The argument in favor of the separation of algorithm and grammar is roughly the following: many proponents of formal grammar claim that it is possible to construct a single simple parsing algorithm to be used with any of several grammars of a certain type. The type of grammar has to be specified very precisely by means of a grammar rule format. These grammars can be written in the form of tables of rules, and the same algorithm can be used alternatively with several of these grammar tables, provided the rule format is adhered to. The advantage of this approach is supposed to be greater simplicity and easier checkout and updating of the grammar. This is because the algorithm need not be changed every time a correction is made in the grammar: presumably any such correction will be a simple revision of the grammar table.

3. In assessing the usefulness of the separation of grammar and algorithm, it is important to keep in mind the well-known distinction between context-free and context-sensitive grammars. In this author's frame of reference, this distinction can be formulated very simply as follows: a context-free grammar is one in which only the internal structure of a given construction is taken into account; a context-sensitive is one in which both the internal structure and the external functioning are taken into account. This view follows from the conception that internal structure and external functioning are two separate related but not identical functional characteristics of the units of language such as syntactic units.

There are two important considerations which follow from this. One is that very often the internal structure of construction is not adequate to determine its external functioning. The well-known fact must be taken into account that sequences with identical internal structure may have vastly different modes of external functioning and conversely. Examples of this are very common in English, and include many of the frequently cited instances of nesting. The second consideration is that the determination of external functioning by context searching is not a simple one-shot operation. It is not always possible to formulate a particular single context for a particular sequence that is to be examined. Rather, the variety of contextual conditions which may apply to a particular construction may differ from sentence to sentence and the particular conditions that apply can only be determined by a graduated search of a potentially ever extending range of contexts. This means that one can not simply talk of context-sensitivity in a

grammar but one has to talk of degrees of context-sensitivity. In order therefore to parse natural language data adequately, the parsing system has to have, not merely some fixed capability of being sensitive to a certain range of contexts, but it must be capable of modification in terms of increase in context-sensitivity.

The above means that the most significant alterations in grammar rules from the standpoint of natural-language parsing will not be those that affect the formation of particular rules within the same format. Rather those alternatives which will really make a difference in the adequacy of the parsings of natural language sentences will be alteration of the format itself in terms of increasing the degree of context-sensitivity. This is effect means that the simplicity claimed for a separate table of rules with a constant algorithm turns out to be illusory, since the proponents of this concept of simplicity admits that it applies only when the rules are held to the same format.

4. Another point raised in connection with the separation of grammar and algorithm is that the grammar table constitutes a set of input data to the particular algorithm, in a similar way in which the sentences to be parsed constitute input data.

In this author's opinion, this is again an oversimplification.

First of all, it is to be noted that in the view of many programmers, only those data are considered input that are designed to be actually processed. Since the grammar rules are not intended to be subject to processing, but rather to constitute the parameters for processing, they are not input data in any way comparable to the sentences that are to be parsed.

If, on the other hand, the question of processing is to be ignored in deciding what is to be viewed as input data, then another consideration must be taken into account. It is the following: the question as to what constitutes input can not be answered in the absolute, but only relatively. That is, the question is not simply "is it input," but "what is it input to". This means that the answer depends, at least in part, on what portions of the program are previously present in the work space, and what additional portions are inputted subsequently. In a program in which the grammar is written into the algorithm, such as is the case in the approach which this author has taken, the question of whether the grammar constitutes input data can then be viewed as follows: while the grammar does not constitute a separate set of input data, it will nevertheless use separate sets of grammatical input data in the form of a grammar-coded dictionary that is fed into the program from a separate source. Likewise, it is possible to view the executive routine of the algorithm which contains the grammar as the actual parsing algorithm, and to view the remaining portions as forms of input data.

5. Leaving aside the matters of rule format and input data, two further questions can be raised concerning the simplicity that is claimed to result from the separation of grammar and algorithm. These questions are pertinent in the case of a grammar having sufficient context-sensitivity to serve the needs of syntactic recognition adequate for the machine translation of natural languages.

a. Since the table will tend to be increasingly complex because of the requirement of high context-sensitivity, a dictionary-type binary lookup may no longer be sufficient. Rather, it may

become necessary to devise an algorithm for searching the table in such a way that the graduation of contextual conditions is properly taken into account.

b. Revisions of the rules in such a complex table will not be as simple a matter as it seems, because it will no longer be obvious which of the rules is to be modified in a given case, nor will it be obvious where in the table this rule can be found. Likewise, it will not be obvious what contextual conditions will have to be taken into account in order to bring about the desired modification.

6. As can be seen, the argument in favor of the separation of grammar and algorithm is considered far from convincing. It does, however, raise a related question: if the major separation is not to be that between grammar and algorithm, what then are the major components of a parsing program?

The answer which this author has found satisfactory is the well-known one of structuring the parsing program as an executive main routine with appropriate subroutines. This raises the further question of the functions and design of the executive routine and subroutines.

In a parsing program, the function of the executive routine will be to determine what units to look for and where to look for them. The aim of the subroutines will be to provide the means for carrying out the necessary searches.

The design principle for this type of parsing program will be the well-known one of modularization: the subroutines of the program will consist of self-contained and interchangeable modules.

The modules will be of two kinds: analytic modules, the purpose of which will be to perform tasks of linguistic analysis such as the determination of the internal structure and external functioning of the different constructions that are to be recognized; housekeeping modules which are to insure that the program is at all times aware of where it stands. The latter means the following: the program has to know what word it is dealing with, the program has to know at each step how far a given search is allowed to go and what points it is not allowed to go beyond, the program has to be informed at all times of the necessary location information such as sentence boundaries, word positions in the sentence, search distances, etc.

ACKNOWLEDGEMENT

Work on this paper was done under the sponsorship of the Information Processing Laboratory of the Rome Air Development Center of the U. S. Air Force, under Contract AF 30(602)-3506.