[From: *Information Processing 1965*: Proceedings of the IFIP Congress 65 (Washington, D.C.: Spartan Books, 1965)]

ALGORITHMS OF MECHANICAL TRANSLATION

O. S. KULAGINA

Steklov Mathematical Institute
Academy of Science of USSR
Moscow, USSR

THE PROBLEM

In developing AT (automatic translation) algorithms, the following closely connected tasks come up:

- The elaboration of a notional system for the description of language;
- A description of the chosen languages in terms of those notions;
- The creation of algorithms providing for the transition from the text in a certain language to its description and the reverse.

The first and last task cited above are the concern of mathematicians, while the second task is is the concern of linguists, provided there is close and constant cooperation between the former and the latter. In practice the three tasks are hardly ever distinctly formulated, and very often they are tackled without separation; that is, the system of description is worked out along with the creation of an algorithm.

The solution of the first task comes down to the choice of a set of elements for the system of description; their classification; the definition of relations among them; and the definition of rules for constructing more complicated units out of the simpler ones. The description of a language in terms of such a system amounts to setting up correspondences between language elements (words, phrases, texts, etc.) and elements of the system of description (the information ascribed to a word, the structure of a phrase, etc.). Thereby, facts and regularities observed in the language are formalized in the system of description in the relation among its elements. In that way the definition of a correct phrase structure in the given language is formed.

It is worthy of notice that every description of a language is just an approximation which is more or less crude; in every description a certain number of possibilities are left out of the picture. An adequate description seems to be equivalent in volume to the language described, and this solution of the problem is obviously senseless.

We can think of a number of levels of description: usually one considers the levels of morphology, syntax and semantics, but generally speaking, we can set up a system with a greater number of levels where every subsequent level will be related to the previous one as semantics to syntax, because every subsequent level ties up the elements of the previous level with something lying beyond it.

The transition from a text to its description is analysis, while the reverse process is synthesis. AT algorithms may be classified according to depth, that is, the chosen level of description which we reach in analyzing the text and from which we start from in synthesizing it. At present, analysis in the majority of cases does not go any deeper than the syntactic structure of a phrase with certain elements of meaning.

THE LEVELS OF ANALYSIS

By fixing the depth of the final results we do not yet solve in any way the issue of a reasonable division into levels of the process of reaching this result. It is desirable to go down to the deeper levels gradually, so as not to overload the algorithm.

Research tools should harmonize with the object. Superficial levels may be studied with more or less primitive tools; the deeper we go, the finer should be the system of features and theoretical considerations. The confusion of finer and cruder considerations at the same level results in an unjustifiable padding of the algorithm.

Apparently, the task consists in providing every level with a method of work of its own. The use of fine methods at a superficial level is as pernicious as the attempt to force deep things into a crude methodological framework.

There is good reason to suppose that the combination of a number of essentially different approaches, not necessarily too fine, will be more successful than the attempt to develop and refine over and over again one single approach. The difficult thing is that well developed methods are available only for a few superficial levels. The deeper we go the less we know and the greater are the difficulties.

The question of where to draw the boundary between the levels comes up. It may turn out to be reasonable to set up as distinct levels various stages of analysis: the division of the text into parts (segments); analysis of the internal relations among the parts of the initial segments; the combination of initial segments into larger ones and the analysis of the relations among those larger units, etc., until the text itself emerges as a single unit. Another possible solution is to set up as many levels as there are classes of syntactic connections (strong connections, weaker ones, still weaker ones. etc.). until all the connections are established.

ON THE INADEQUACY OF DESCRIPTION TO LANGUAGE AND ALGORITHM TO DESCRIPTION

For every level one should solve the same standard tasks: to choose the system of description of a definite level, to describe the regularities of the given level for the chosen language and to work out an algorithm of transition from the previous to the given level. At every level the following question comes up. Suppose we have formulated the notion of a *correct structure* for the given level. We shall say that a structure suits a phrase if the linguist regards it as corresponding to the given phrase. As every description is merely a rough approximation to the object, the sets of correct and suitable structures do not coincide (the inadequacy of description to language. The problem to be clarified is how they are related.

The question of a sensible division into levels has not yet been sufficiently investigated; ordinarily, the transition from the morphological to the syntactic structure is considered as a whole (syntactic analysis).

The two principal ways of describing syntactic structure which are in use at present are dependency trees and constituent trees (equivalent forms being disregarded); there are a number of papers concerned with clarifying their interrelations.[1,2]

The algorithm of syntactic analysis is an algorithm of transition from a sequence of information ascribed to words to a certain correct syntactic structure. The question arises, how many correct structures should be put into correspondence to a phrase (provided the case is unambiguous for a linguist). The point is, that even within the chosen system of description the algorithm may be inadequate to the description (although adequacy is in principle possible); that is, for a phrase whose suitable structure is correct, the algorithm may yield apart from the suitable structure some other correct structures, or else it may not yield the suitable structure at all. The combination of both of these cases is also possible. The establishment of a precise interrelation between the description constructed and the algorithm is a mathematical task (similar to the ones solved in Greibach[3]), while the clarification of the interrelations between a certain language and its description (the revelation of insufficiency of the description) is a linguistic task. Unfortunately, in the majority of cases the question of these interrelations is not explicitly solved; usually one does not go beyond merely pointing out the cases when the algorithm yields an unsuitable structure.

THE METHODS OF DETERMINING SYNTACTIC STRUCTURE

After the final aim of syntactic analysis is defined, i.e., the notion of a correct structure is established, and after the strategy is laid out, i.e., it is determined whether we allow the obtaining of more than one correct structure for a phrase, whether the correct suitable structure may be lost, it is still necessary to decide upon a certain tactical plan. There are a great variety of ways of determining structures with two principal approaches: sequential analysis[4-9] and the filter method.[10-12] In sequential analysis, as far as possible, an attempt is made to avoid getting at intermediate stages redundant structures or parts of structures which may turn out to be incorrect; correspondingly, in processing every word, an effort is

made to use to the utmost the data on other words and their syntactic connections. If the question of the syntactic connections of a given word with other words has no unique solution, the most probable solution is chosen (as in Moloshnaya's algorithm[5] directed at yielding one final structure), or else the process ramifies (as in Oettinger's algorithm[8]), and all the different ramifications are scanned to the end one by one or expanded simultaneously.

Under the second approach, one sets up, in a way which is simple and general enough to be applicable to various languages, a set of structures among which there are many incorrect ones, and then picks out the correct structures. This method is effective when the number of intermediate structures is not very large; otherwise it will take the machine too much time to perform the work. Therefore, a sensible combination of the two methods, that is, finding sufficiently simple limitations on the structures that are being established, should be preferred. It is important to find a reasonable balance of the information used in establishing structures and the information used in picking out the correct ones.

THE STAGES OF SYNTHESIS

The problem of independent synthesis has received much less attention than the problem of independent analysis. For that reason I shall speak in greater detail about the algorithms of synthesis rather than those of analysis. Similarly to analysis, synthesis may be divided into semantic, syntactic and morphological stages. The questions of semantic synthesis have hardly been tackled at all, and therefore I shall have to leave them out of this discussion. On the contrary, algorithms of morphological synthesis have been intensively developed, and some such algorithms have been published.

I shall dwell below on some particulars of the stage of syntactic synthesis, that is, the stage of transition from the syntactic tree of a phrase to a string of information which makes it possible to construct actual word forms. This stage consists in the determination of the grammatical data sufficient to define the word form, and the determination of the word order.

Synthesis differs from analysis in that it is, as a rule, non-unique. While an unambiguous phrase has only one suitable structure, there are ordinarily very many ways of synthesizing a phrase with a given meaning. Even in simple cases the number of variants may amount to a thousand or more.

That is why the task may be formulated in different ways:

- To obtain one correct variant;
- To obtain a number of correct variants (in the extreme case, to obtain them all);
- To obtain a number of correct variants and pick out the one which is optimal in one sense or the other.

Accordingly, the algorithm may be organized in different ways.

SYNTHESIS WITHOUT ESTIMATES

Algorithms constructing a single variant for a phrase have been constructed as part of a translation system (in all groups where AT algorithms were built), as well as independent systems.[13,14]

The operation of such an algorithm results in obtaining, on the basis of the phrase tree, a string of information to words, and that marks the end of this stage of synthesis under the first formulation. Under the second formulation one proceeds to constructing other strings. Within the algorithm no means are provided for estimating the strings, regardless of whether one or more strings are formed.

SYNTHESIS WITH ESTIMATES

Under the last formulation the process of synthesis does not end with the formation of a certain correct string or strings.

I shall dwell in some detail on the scheme of an algorithm meant to solve the task of Russian synthesis under the last of the above formulations.

In this algorithm we intend, in the case of a compound or complex sentence, to construct the clauses separately and devise a special routine for assembling the larger unit.

The syntactic synthesis of a simple sentence is effected in three steps. The first step consists in forming the so-called initial word groups (IWG). Each IWG consists of a head and a number of its adjuncts (dependents). IWG may be of four different types depending on the class of the main word (noun, verb, adjective or adverb). The place of the main word within the group is fixed once and for all. The place and the form of every other member of the group are fully determined by the type of relation between this member and the main word and are independent of the other members of the same group or other groups. In subsequent stages of synthesis every IWG functions as a single whole; in particular, it may make part of another word group.

As a result of the first step of synthesis, we obtain a tree of the IWG's, the governing member for the head of an IWG being the governing member for the whole group.

The second step consists in combining IWG's into the so-called terminal word groups (TWG). The TWG's which are not governed by the predicate are built into groups of the corresponding governing members. Two cases are possible: (1) The IWG is a member of another group; then it is inserted at a definite place into the group of the governor; (2) The IWG should be placed to the right (or the left) of its governor. If there are several groups to be placed on one side of the governor, the question of their relative order is settled with a view to observing the grammatical, stylistic and semantic norms of the output language.

As a result of the second step we obtain a set of TWG of the finite verb, the subject and the objects, the adverbial modifiers, the nominal and infinite parts of the compound predicate.

The third step consists in arranging the TWG's. Unlike the word order within IWG's and the order of IWG's within TWG's, the order of TWG's cannot be uniquely stated on the basis of the properties of two connected TWG's and the type of relation between them. To arrive at the right word order, it is necessary to take into account the whole set of TWG's that are being arranged, and to consider a number of factors interacting in a complicated manner with one another.

The arrangement of groups proceeds by degrees. At the beginning a tentative arrangement is effected. It is based on a certain norm of order which is correct if no word of the phrase carries any special logical stress, all the groups have approximately the same length, there are no clashes between the groups producing ambiguity, and so on.

In different tentative arrangements the places of different components are fixed to a greater or lesser degree. The places of some components are strictly fixed and cannot be changed under any circumstances. The places of some other components may be changed only under the influence of very powerful factors, such as the logical stress. Still other components can be easily shifted. To take account of all these factors we assign a weight to every component in the given arrangement, that is a number characterizing the degree of rigidity of a given component at a given place in a given arrangement (these numbers are chosen empirically).The greater the weight of a component, the more difficult its shifting. For the factors considered below we point out the weight they can *cope with,* and the weights the components will have in a new arrangement.

At the second stage of arrangement the logical stress is considered (it is presumed that the necessary data are obtained in analysis). At this stage we either shift certain groups, or insert emphatic particles; the rearrangement of groups necessary in interrogative sentences is also effected at this stage.

The third stage consists in substituting pronouns for certain groups.

The fourth stage consists in evaluating the resulting sequence of TWG's from the point of view of a number of criteria (about 20) each of which singles out an undesirable property. For instance, such properties are: (1) a very short group not carrying any stress is placed nearer the end of the sentence than a longer group; (2) there is no group to the left of the predicate group while to the right of it there are more than two groups; (3) there are three or more neighboring adverbial modifier groups; (4) there is an ambiguity; (5) an adverbial modifier group separates the predicate group from the object group, etc. Each of these properties is assigned a conventional *mark*—some negative number. The output sentence is assigned a mark which is the sum of all the marks assigned to the undesirable properties contained in it.

Then we test in turn all possible rearrangements of the groups whose weight does not exceed a fixed number, in order to get rid of as many as possible undesirable phenomena and thus raise the sum-total of the sentence. But it is not always possible to avoid all undesirable phenomena: a rearrangement cancelling one bad construction may entail another. Still, such a rearrangement may be helpful if the sentence's total rises: while admitting some undesirable phenomena we eliminate a more undesirable one.

The result is either a sentence with the highest mark, zero *(sentence without drawbacks),* or a number of sentences with negative marks from which the one with the maximum mark, i.e., with minimum drawbacks, is chosen as the best.

If, however, several sentences with equal marks are produced, the best one may be chosen by means of preference rules. Should these fail to select a single sentence as the best, several variants considered equivalent in quality are sent to the output.

Failure by the algorithm to produce a zero mark output sentence means that the respective meaning cannot be expressed smoothly enough by means of what it has at its disposal. Some radical modification of the input syntactical tree (breaking it down

into several independent sentence trees and replacing the vocabulary, i.e., the nodes, involved) may be needed. We are proposing to develop a special device capable of making such modifications.

In synthesis as well as in the analysis it is necessary to keep down the number of possible scannings in searching for the optimum variant.

The two approaches in synthesis (with or without estimates) are in a certain respect similar to the two approaches to analysis, the sequential analysis and the filter method. Under the filter method and estimate synthesis we deal with the notion of a correct structure (good phrase) defined *statically* regardless of the process by which it was obtained. The use of filters of estimates is a criterion by which we test the correspondence of the object produced by the algorithm to this notion. Under sequential analysis and synthesis without estimates, the problem of testing for quality the object produced by the algorithm is not handled within the algorithm.

THE USE OF MACHINES IN CONSTRUCTING AT ALGORITHMS

There exists one more class of problems which I shall merely touch upon. I refer to the problems of the use of computers in constructing algorithms, in collecting language data for the construction of algorithms and in correcting and improving the algorithms.[15-18]

By the character of problems AT is an empirical science and the need for experiment in this field is as great as in biology, chemistry or physics.

However, AT experiments are specific in that they amount to modelling by a computer. Computers have various functions in AT and correspondingly there are various ways of staging experiments with their help.

- The testing of a ready algorithm.
- The programming of an algorithm.
- The collection of language data, table construction, data classification, statistics, etc.
- The correction of an algorithm by means of testing it on a computer, comparing the result with a certain standard. The mistakes are either marked off and classified, or rectified, or else hypotheses are formulated concerning the possible source of these mistakes and propositions to eliminate them (on condition that the hypothesis is accepted, that is, if certain conditions are satisfied).
- The incorporation into the algorithm of various corrections, formulated by the human operator, and the testing of the consistency of the resulting algorithm.
- The comparison of experimental results not with a certain standard made up in advance, but against a certain set of conditions to be met, and the correction of the algorithm if it yields results incompatible with the given conditions.

It is extremely important that the role of computers in the development of AT research should constantly grow.

REFERENCES

1. Е. В., Падучева, "О способах представления синтаксической структуры прелдложения," *Вопросы язкознания, 2,* 1964.
2. D. G. Hays, "Grouping and Dependency Theories," *Proceedings of the National Symposium on Machine Translation,* London, 1961.
3. S. Greibach, *Inverses of Phrase Structure Generators,* Math. Ling. and Automatic Translation, Report NSF-11, Harvard Univ., 1963.
4. И. А. Мельчук, "Автоматичуский анализ текстов (на маериале русского языка), Славянское языкознание, Moscow, 1963.
5. Т. Н. Молошная, "Алгорим перевода с английского язка на русский," *Проблемы кибернетики,* 3, 1960.
6. D. G. Hays and T. Ziehe, *Studies in Machine Translation. 10.Russian Sentence Structure Determination,* U.S. Air Force, RM-2538, 1960.
7. M. Corbe and R. Tolbory, "Introduction to an Automatic English Syntax (by fragmentation," *Proc. of the 1961 International Conference on MT of Languages and Applied Language Analysis,* London,1962.
8. A. G. Oettinger, "A New Theory of Translation and Its Applications," *Proc. of the National Symposium on Machine Translation,* London, 1961.
9. I. Rhodes, *A New Approach to the Mechanical Translation of Russian,* National Bureau of Standards Report No. 6295, 1959.
10. F. Lecerf, "Programme des Conflicts, Modèle des Conflicts," *Traduction Automatique,* 4, 1960.
11. F. Lecerf, "L'Adressage Intrinséque en Traduction Automatique," *Traduction Automatique,* 2-3,1961.

12. Л. Н. иорданская, "Свойсва правильной синаксической структуры и алготм ее обнаружения (на материале русского языка)," *Проблемы кибернетики*, III 1964.
13. Л. Н. засорина, "Порядок слов при синезе русского предложения," *Материалы по математической лингвистике и маш.переводу, 2,* 1963.
14. М. И. Откупшикова, *Позиционнй этап синтеза русского предложения при машинном переводе,* НТИ, II , 1963.
15. О. С. Кулагина, "Об использовании машин при состлении алгоритмов анализа текста," *Проблемы кибернетики,* 7, 1962.
16. V. Giuliano, *A Formula Finder for Automatic Synthesis of Translation Algorithms,* Mechanical translation, 6, 1961.
17. K. E. Harper and D. G. Hays, *The Use of Machines in the Construction of a Grammar and Computer Program for Structural Analysis,* Rand Corporation Report P-1624, 1959.
18. *Report No. 16. Final report,* Linguistics Research Center, The University of Texas, 1963