[From: D.G.Hays (ed.) *Readings in automatic language processing*
(New York: Elsevier), 1966]

10 On the Mechanization of Syntactic Analysis

Sydney M. Lamb

Any language has as part of its structure patterns according to which items are arranged on each of its strata. The term *tactics* is widely used for the analysis and description of arrangements, and the term syntax is traditionally used with reference to arrangements on the morphemic straturn. It is in connection with that stratum that the study of tactics of greatest interest in linguistics.

The items with which syntax is concerned can be of varying kinds, depending upon the school of thought. Some linguists regard the word as the basic unit of syntax; others make no syntax-morphology distinction, and we could apply the term *syntax* here also, with the morpheme as the basic unit. It is also possible to use items which tend to be smaller than words but larger than morphemes, and one unit of this kind is in fact what I prefer. I call it the *lexeme.*[1] But for purposes of this paper, let us think of syntax as being quite general with regard to the choice of the basic unit. The technique of analysis to be discussed applies for any of these kinds of items. After all, if one goes to the trouble of writing a computer program for syntactic analysis, one ought to make it as widely applicable as possible to the needs of different linguists. Indeed, the system as described in this paper, and the accompanying computer program, could also apply to the study of arrangements of phonemes or letters or syllables or morphographernes and perhaps also various nonlanguage phenomena which tend to occur in patterned linear arrangements. In other words, it is really a systern for tactic analysis in general.

At any rate, whatever unit is taken as the basis of the tactic description

Note: Research performed at the University of California, Berkeley. Reprinted from *1961 International Conference on Machine Translation of Languages and Applied Language Analysis* (National Physical Laboratory Symposium No. 13), Vol.II,pp. 673-684, with the permission of the Controller of Her Britannic Majesty's Stationery Office, publisher, and of the author. The paper is somewhat abridged; more recent accounts of the author's theoretical position have superseded that originally included.

(word, lexeme, morpheme, or what-not) will be referred to as an *item* for purposes of this exposition.

The syntax may be completely described by a list of distribution classes of items, with the membership of each, and a list of constructions. A *construction* is characterized by specification of (1) the distribution classes which enter into it and their relative order, (2) the distribution-class membership of the constitutes. Lists of distribution classes of composite forms need not be given in the description (even though they exist), since they are defined by the constructions.

To say that a syntactic description consists of lists of distribution classes and constructions, however, is to specify only its form. There are any number of possible descriptions for a given language which could take this form, but only a few of them are good and only one is the best. It must further be specified, then, what constitutes the best solution. Alternatively, one could specify a procedure which, if followed, leads to the best solution. This latter approach has been popular in linguistic methodology, but it tends to be unnecessarily complicated. In syntax (or tactics in general) we can provide for good analysis and description very easily, by means of a simple definition. Taking for granted that the fundamental requirements of completeness, accuracy, and consistency are met, the best description of the syntax of a language is (naturally) the simplest. Simplicity in this area can be very clearly defined. *The simplest syntactic description is that which makes use of the smallest number of constructions.* It must also be specified that if two solutions have the same number of constructions, the one with simpler constructions is to be preferred. Thus we must now define *simple* with regard to constructions. A construction without discontinuous constituents is simpler than one with such constituents. And among constructions with different numbers of constituents, the simplest is that with the fewest constituents.

Having this simple definition of the best of all possible syntactic descriptions of a language, the analyst can use it either to show that a proposed solution is better than some alternative or, ideally, that it is better than all possible alternatives.

All valid criteria for determining immediate constituents can be deduced from the basic definitions. And most of the criteria which have been put forth by various linguists in recent years are valid in this sense. On the other hand, two principles are worthy of note as having been mentioned at one time or another without being valid. One of these is that constructions should always be binary or that they should always be binary except in the case of coordinate constructions having more than two members. The other, applicable only if items smaller than words, such as lexemes, are taken as the basis of the description, is that words must always be constituents.

Any procedure which arrives at a description satisfying the basic requirements is a valid one. If, therefore, one were expounding on syntactic analysis for the sake of human beings, any remarks added to the above having to do with procedures would serve only pedagogical purposes. On the other hand, if one wants to have a computer do syntactic analysis, it is necessary to specify a procedure in complete detail, since present-day machines are altogether lacking in intuition and ingenuity.

Let us now go into some general considerations relating to the application of computers to syntax, after which I will describe part of a specific procedure which I am currently working on.

The machine should use texts as its primary source of information, but it could also be enabled to ask for further information from the informant, just as human linguists do, in order to compensate for the absence of an infinite text. However, the machine will not be quite as dependent upon the informant as humans are, because, taking advantage of its capacity to process data at very high speeds, it will be able to work with much larger amounts of text than would be feasible for the human analyst. By the same token it should be able to do a more detailed analysis than is generally possible.

It need not be required in the initial attempts that the machine program be able to do the entire job of syntactic analysis. Provision can be made for it to admit failure on difficult problems, printing out the relevant data and leaving the solution up to human intelligence. Also, one can keep the initial stages simple by operating only in terms of binary constructions with continuous, obligatory constituents. Consideration of the more complicated types of constructions can be taken up at a later stage of the process.

The program should be designed to do its preliminary analysis on a fairly small portion of text (say around 5,000 items) at first, after which a larger amount can be considered for purposes of more detailed analysis. When the larger portion is brought in, its items can first be classified to the extent possible on the basis of the preliminary analysis, and tentative groupings based on the provisional constructions can be made. The data of the larger portion of text will thus be greatly simplified for the sake of the further analysis, even though some of the provisional conclusions may have to be rescinded.

For the remainder of this brief paper, let us consider just the preliminary analysis that is to be done on the first 5000-item portion of text.

In the course of the analysis, groupings of two kinds will be made. These may be referred to as *horizontal* and *vertical* groupings, or H-groups and V-groups for short. A *vertical grouping* or V-group is a grouping of items (and/or sequences of items) into a distribution class or an approximation to a distribution class. An H-group or *horizontal grouping* is a grouping of constituents of a construction (or tentative construction) into a constitute. Thus a combined horizontal and vertical grouping yields an actual or provisional constitute class. After an H-group or V-group has been made, it can be treated as a unit for the further conduct of the analysis. The term *unit* will be used from here on to refer to any item, V-group, or H-group.

But how is the machine going to make these V-groups and H-groups? Zellig Harris, in his procedure-oriented *Methods in Structural Linguistics*[2], set up distribution classes of morphemes before considering horizontal groupings. To do so in a meaningful way requires that items grouped together be found in identical environments extending several items on either side. It would be futile to attempt such an approach even with a machine because a corpus of truly colossal proportions would be required, and even the computer has limits with regard to the volume of data that can be processed at high speed. One must design the procedure, then, so that the sharing of certain *significant* distributional properties, rather than certain total environments, will be the criterion for combining units into the same V-group. And such an approach requires that a certain amount of horizontal grouping be done first, since it is only in terms of H-groups that we can define significant distributional properties in advance of the completion of the analysis. Now it happens that there is a means of setting up H-groups which are at least usable approximations to constitutes of actual constructions, without the aid of any prior vertical grouping. This method makes use of a concept which I call the *token neighbor ratio*, or T/N ratio for short.

Any specific occurrence of an item may be called a *token* of it. The number of tokens of an item in a text is thus equal to the number of times that item occurs. Any item which occurs adjacent to another item is a *neighbor* of the latter. If two items A and B occur contiguous to each other, A at the left, then A may be called a *left neighbor* (LN) of B, and B may be called a *right neighbor* (RN) of A. The number of tokens of a given item in a text divided by the number of different right neighbors (i.e., RN types) may be called the *token/right-neighbor* (T/RN) *ratio* for that item *in* that text. Similarly, the ratio of the number of tokens to the number of different left neighbors (i.e., LN types) is the *token/left-neighbor* (T/LN) *ratio* for that item in that text. T/RN and T/LN are the two lands of token/neighbor (T/N) ratios.

The first step in the analysis is to compute the two T/N ratios for every different item in the text. For a 5,000-item text, this takes about eight to ten minutes on an IBM 704, depending on the number of item types present. In the course of calculating these ratios for each item, lists of right and left neighbors will be formed but they will not be saved since the aggregate of such lists would soon become very bulky and those individual neighbor lists that will be needed later can be constructed again very rapidly when needed.

The highest T/N ratios identify the points of maximum restriction on freedom of combination, insofar as such identification can be made without prior information about the structure of the language.

The process continues with consideration of the item having the largest T/N ratio. This item we may call the *current most restricted unit,* or CMRU. Later the next largest ratio will be considered, and so forth, but various ratios will also be undergoing modification to give effect to horizontal and vertical groupings treated as units, so the second highest may not turn out to be the highest after the first has been dealt with.

Table 1 shows an ordered list of the items ("quasi-lexemes" in this case) having the highest T/N ratios in a particular English text, a selection from the writings of Sir Winston Churchill.[3]

The neighbor class with respect to which the CMRU has the highest ratio may be called the SNC, for *small neighbor class.* It is necessarily small. Moreover, its smallness has significance since the item of which its members are neighbors occurs with relatively high frequency in the text. That is, the

Table 1. Highest T/N Ratios of Items (Quasi-Lexemes) in a 5,000-Item English Text, Excluding Ratios of Punctuation Lexemes

Item	*Token Count*	*LN Count*	*RN Count*	*T/N*
are	82		6	13.67
-s (verbal third singular)	53	4		13.25
-*'s*	85	9		9.44
new	8	1		8.00
have	58		8	7.25
own	6	1		6.00
Adolf	5		1	5.00
the	327	66		4.95
but	9	2		4.50
they	40		9	4.44
seem	4		1	4.00
call	4		1	4.00
Rhineland	4	1		4.00
he	75		21	3.55
be	28	8		3.50
Reichswehr	7	2		3.50
-pl (nominal plural)	223		67	3.33
German	34	11		3.09
force	9		3	3.00
it	24		8	3.00

highest T/N ratio can be the highest only by virtue of the fact that the size of T (number of tokens) is relatively large while the size of N (number of neighbors) is relatively small. It does not necessarily follow, however, that this item (the CMRU) and these neighbors are partners of each other in a construction nor that this small neighbor class constitutes a distribution class.

In designing the procedure, one is faced with alternatives at this point. One could consider the SNC to be a first approximation to a distribution class. In this case, if it has more than one member, it would be necessary to look for the presence of certain relationships of its members to each other. Specifically, it would be necessary to find out whether any of its members can have any members of this same class as neighbors. For all those members which can, separate position classes (left to right) would have to be set up, and it is even possible (though not likely for the first neighbor class studied because of its small size) that more than one set of such classes would be present.

A simpler alternative is to let the machine refrain from making any vertical groupings at this point, waiting until more information is available as a result of the formation of additional H-groups. In general, we will want to combine units into a vertical grouping *only when they are found to share the same partner in H-groups which, in turn, also share the same partner in horizontal groupings of the next higher degree.* For example, if A, B, C, . . . are items, and if AB and AC are H-groups, then that fact alone is not sufficient grounds for grouping B and C together (*cf.*, *John left* and *John Smith).* But if AB-D and AC-D (or D-AB and D-AC) also become H-groups, then B and C will be combined in a V-group. Even the grouping under these circumstances could be incorrect, however, so re-examination of V-groups will be necessary after further analysis has been done.

As soon as the CMRU is obtained, then, it will be combined with each member of the SNC into one or more H-groups.[4] But since such groupings will often be incorrect, there must be provision for re-appraising H-groups at suitable later points, revising as necessary. Let us take an example. As we might expect, frequently occurring prepositions in English have relatively high T/RN ratios. Suppose that the preposition *in* in a text occurs several times, having as different right neighbors *sand, water,* and *the.* The H-groups *in sand, in water,* and *in the* will be formed. Obviously it is necessary that the last of these be rescinded sooner or later. And it will be, as soon as certain V-groups are made. The article *the* has been combined with the preceding *in* simply because the machine does not yet know that the nouns following it belong together in a V-group. (Let us leave adjectives out of the picture, to keep our example simple.) But as the process continues, these nouns will gradually be grouped together, and the resulting V-groups will be treated as units. Then, if re-appraisal of affected H-groups is conducted as each new vertical grouping is made, it will eventually turn out that the T/RN ratio of *the* is higher than the ratio which led to the combining of *the* with *in,* and that incorrect H-group will at that point be dissolved.

It will be noted that although the procedure begins by considering immediate environments only, wider environments automatically come into consideration as horizontal groups are made.

A detailed summary of the first stage of the process follows:

Definitions

Item: ultimate constituent.
Unit: item, H-group, or V-group.
H-group: horizontal grouping; i.e., constitute of a construction or of an approximation to a construction.
V-group: vertical grouping; i.e., provisional distribution class.
CH-group: complex H-group; i.e., H-group in which at least one partner is itself an H-group.
CMRU: current most restricted unit; i.e., the unit currently having the highest T/N ratio.
NC: neighbor class; i.e., the set of units which are neighbors (right or left) of a given unit in a given text.
SNC: small neighbor class; i.e., the NC with respect to which the CMRU has the highest T/N ratio.

Main Routine

I. Perform *A* on every different item in the text

II. Get the CMRU and for each member of the SNC as partner form a new H-group. For each new H-group, (1) record its membership in reference list; (2) replace it in the text (each occurrence) by a unit symbol for the group (reference list permits restoration in case of later revision); (3) if it is a CH-group, go to *B,* specifying which partner is complex (if both are complex go to *B* twice). Perform *A* for each new H-group and for all units affected by the new groupings (replacing previous information now obsolete), namely (1) units occurring as neighbors of the new H-groups and (2) those members of the SNC which still have occurrence apart from the new H-groups.

III. *Switch,* having the values *plus* and *minus.* (Starts as *minus,* can be set *plus* by *B* and is reset *minus* by *IV.)* If *minus* return to *II*; if *plus* go to *IV.*

IV. Reset switch *III* to *minus*. Form new V-group(s) as indicated by *B*. For each, (1) record its membership in reference list; (2) in text, replace tokens of members by symbol for the group. Perform *A* for each new V-group and for all other units affected by the new grouping. Re-appraise all affected H-groups, revising as needed; upon revision, reappraise any affected V-groups, revising as needed. Return to *II*.

Subroutines

A. Determine the T/N ratios of the specified unit.

B. Split the specified complex partner into its constituents and add the CH-groups (in this form) to the list of CH-groups; let the other partner be called Other Partner. If Other Partner and either constituent of the complex partner match the two members of corresponding posi tion of any other CH-group in the list, set the switch (*III*) *plus;* the third (nonmatching) constituents are to be combined as a V-group.

At the tune of writing, the process is operational on the computer only up to the point at which proper justification is found for making the first vertical grouping. In performing the analysis on some newspaper text from the Associated Press which had kindly been furnished by the MT group at the Massachusetts Institute of Technology, the machine reached that point after forming 31 H-groups, three of which were complex. In this text, capitalization of the following letter was everywhere segmented as a separate item by the M.I.T. group, so much of the horizontal grouping involved combining proper names (such as Poland, Gomulka, and Egypt) with their preceding capitalization. The first vertical grouping consisted of *united* and *mrs.* Both had been combined with preceding capitalization, and each of the two resulting H-groups was found to have capitalization as its only right neighbors

This is, of course, only a beginning. But it is the beginning of a system which may eventually be able to reduce the time required for analyzing the structure of a language from several years down to a few months or even weeks.

References

1. Even though it is defined somewhat differently from the lexeme of Bernard Bloch and Charles F. Hockett; *cf.,* Hockett's *A Course in Modern Linguistics* New York (1958). Chapter 19.

2. University of Chicago Press, Chicago (1951).

3. This text consists of the first 5,000 quasi-lexemes in the first chapter of the *Life Magazine* edition of *The Second World War,* New York, (1959). Quasi-lexemes, for this text, are the items arrived at by segmenting (1) at spaces, (2) punctuation lexemes (including capitalization at the beginnings of sentences only), (3) certain nominal (-pl, -'s) and verbal

(-s, -ed, -en, -ing) suffixes, and (4) -n't and -'ll; where such segmented forms are written so that their morphemic identity in different environments is preserved, regardless of variation which might be present in a graphemic representation.

4. The procedure has been revised so that the CMRU is combined only with that member of the SNC which occurs most frequently as neighbor of the CMRU.

5. When the revised procedure (cf. preceding note) was used on the Winston Churchill text on which Tables 1 is based, the first vertical grouping formed consisted of *he* and *they*.