[From: Allen Kent (ed.) *Information retrieval and machine translation*
(New York: Interscience, 1959)]

CHAPTER 32

Computation of Syntactic Structure*

ARAVIND K. JOSHI

University of Pennsylvania, Philadelphia, Pennsylvania

1. INTRODUCTION

This paper describes a procedure for computing the syntactic structure of a sentence of a language (in particular of English). A sentence of a language is a string of words from the vocabulary of the language. The sentence can be rewritten as a sequence of class-marks where each word of the sentence is replaced by a mark corresponding to a class or classes to which the word belongs. One method of computing the syntactic structure consists of the following activities: 1. Segment the string of class-marks into substrings (first order substrings). Each first order substring has a definite structure. First order substrings do not nest inside another first order substring of the same type. For example, elementary noun phrase, elementary verb phrase, and adjunct phrase. 2. Compute the second order substrings which are fixed sequences of the first order substrings. A second order substring can include one or more second order substrings. For example, second order substrings beginning with which, when, whom, who, etc. 3. Examine the sequence of these substrings to check whether the sentence is well-formed or not. A primitive well-formed sentence is defined in terms of the first order substrings. All second order substrings (except those which act as noun replacers) and certain first order substrings (P$\bar{N}$ phrases, D etc. with the exception of those which are required in the verb object) can be omitted because they do not add to the well-formedness requirements.

Some specific methods and problems of this computation will be discussed in this paper. Only a summary description of the procedure will be given here. The detailed procedures and other theoretical aspects have been described elsewhere.†

*This paper is based on the work done by a group working on the Transformations and Discourse Analysis Project in the Department of Linguistics at the University of Pennsylvania. The project is sponsored by the National Science Foundation.

†Transformations and Discourse Analysis Project at the University of Pennsylvania—Papers 15, 16, 17, 18, and 19.

We will start with a sentence in which the words of the sentence have been replaced by a mark corresponding to a class or classes to which the word belongs* e.g., Sequence:N, Write:V, For:P, Quickly:D, We:R, Plan:N/V (i.e. N or V), The:T, Blue:N/A (i.e. N or A), etc. Thus we have the sentence as a sequence of class-marks with some words bearing two or more class-marks. The computation (or the recognition) of the syntactic structure of the sentence now begins. At first we find that it is possible to isolate substrings of class-marks with certain properties. These are the first-order substrings. Further computation, which consists of the computation of the second-order substrings and the well-formedness, can be done with the help of these substrings and the remaining class-marks.

II. COMPUTATION OF THE FIRST ORDER SUBSTRINGS

A. Here we mark off substrings of class-marks with certain properties. Once these substrings are marked off the detailed structure of these substrings is not required for further analyses. The class-marks which form such a substring participate as a unit in the sentence structure. In these first-order substrings there is one class-mark which is the principal class-mark, and the remaining class-marks bear relation to this class-mark. The domain of their relationship does not extend beyond the substring concerned. Hence, we call these substrings first-order (or local) substrings.

Moreover it is possible to compute (or recognize) these substrings by scanning either in the right to left or in the left to right direction depending on the type of substring under consideration. Since the first-order substrings do not nest inside another first-order substring of the same type it is possible to carry out their computation by a finite state device. This is not possible with the second order substrings because of the possible unlimited nesting (see III).

*We will not describe here the activities of dictionary look-up, the treatment of certain word complexes, and the possible resolution of some multiply classified words. The treatment of certain word complexes has been presented by Lila Gleitman in her paper "The Isolation of Elements for Grammatical Analysis" (Chap. 31 in this volume). The procedure for resolving some multiply classified words is briefly as follows. Suppose a word bears two class marks α and β. First we apply a set of tests which look for environments in which the classification α cannot definitely hold. If we find such an environment then the classification β is accepted. Next we apply another set of tests which look for environments in which the classification β cannot definitely hold. If such an environment is present then the classification α is accepted. If both the sets of tests give a negative result then the multiple classification remains unresolved.

See the paper 16 and 17 of the project for further details of these two activities. Also see the footnote on p. 838.

B. The substrings with which we will be concerned in this paper are (1) the elementary noun-phrase, which is marked off by [ ], (2) the adjunct, which is marked off by ( ), and (3) the elementary verb-phrase, which is marked off by { }. The first substring is recognized by scanning the sentence in the right to left direction; the second and third, by scanning in the left to right direction. These substrings have to be recognized in the following order, [ ], ( ), { }, because ( ) substrings can include [ ] substrings, and { } substrings can include ( ) substrings.

C. The general procedure for recognizing these substrings is as follows: We scan the sentence from the right (or from the left), and as soon as we recognize a class-mark which appears at the end (or the beginning) of the substring, we close (or open) the bracket and continue reading backwards (or fõrwards) as long as we meet class-marks permitted by a tree (see Fig. 1), which is a structural description of this substring. Sooner or later we end on a terminal branch of this tree. On the terminal branch, we find instructions for opening (or closing) the bracket. After placing one pair of closed and open brackets, we start fresh and repeat the procedure until we reach the beginning (or the end) of the sentence.

D. The elementary noun-phrase (marked off by [ ]; later replaced by the symbol $\bar{N}$): Fig. 1. is a very much simplified tree representation* of an elementary noun-phrase. Such a representation is convenient for recognizing a noun-phrase by a sequential scanning (in the right to left direction) of the sentence. The procedure is as follows. We scan the sentence from right to left. A closing bracket,], is placed as soon as we meet a class-mark which allows us to enter the tree in Fig. 1. The possible entrance points are N and R. Once we

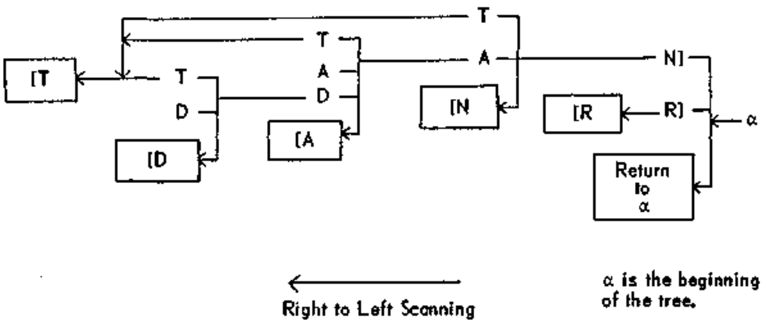

Figure I

*An alternative formulation of a tree representation in terms of functors has been given by H. Hiż in his paper "Steps Towards Grammatical Recognition," in this volume (see Chapter 30).

enter the tree at any one of the entrance points, we follow the branches of the tree according to the class-marks we encounter while scanning from right to left. A branch of the tree can terminate (1) on a class-mark e.g. A, N or (2) on a box which describes the way in which the opening bracket, [, is to be placed. For example

[T . . . .

We terminate on a box only when we meet a class-mark which is not in the set of class-marks which appear vertically above the box.

If we terminate on a class-mark we merely loop back to the place on the tree where that class-mark appeared for the first time.

Consider the sentence

```
                  We hear simultaneous oral translations of the

   class-marks:   R   V        A         A        N      P  T

                            technical papers.

                                A      N
```

Following the tree as described we mark off the following noun-phrases.

[We] hear [simultaneous oral translations] of [the technical papers].

Another example: *

> [The problem] of [inducible enzyme synthesis] is currently receiving [much attention] because of [its obvious value] in [the study] of [nucleic acid formation].

E. Adjuncts (marked off by ( ); later replaced by the symbols Q or O):

After recognizing the possible elementary noun-phrases, we scan the sentence in the left to right direction and look for sequences specified by a tree which is not shown here. During this operation, we are paying attention only to the class-marks which have not been included inside [ ] previously. We also take note of [ ] in order to skip it, but we are no longer interested in the contents of [ ]. e.g. [He] walked (quickly).

```
       (D)
```

*The tree in Fig. 1 will not be adequate for marking off all the noun-phrases in this sentence. A more elaborate version, which is not shown here, is required. This is presented in paper 18 of the project.

Some more examples:

[He] talks (very clearly).
(For [this reason]), [we] have studied [the results]
(of [all the experiments]).

F. The elementary verb-phrase (marked off by { }; later replaced by the symbol V):

After recognizing the elementary noun-phrases and the adjuncts, we now scan the sentence in the left to right direction and look for sequences specified by a tree which is not shown here. During this operation, we are paying attention only to those class-marks which have not been included inside [ ] or ( ) previously. The verb-phrase will include ( ), if we find a class-mark just after ( ) which can be included in { }.

Some examples:

[I] {went}.
[Those papers] {may have been published} (in [a hurry]).
[He] {may have gone fishing}.
[I] {may (soon) go}.

III. COMPUTATION OF THE SECOND-ORDER SUBSTRINGS AND THE WELL-FORMEDNESS OF THE SENTENCE

A. The computation of the second-order substrings and the well-formedness of the sentence can be done at the same time because the computation of the well-formedness turns out to be a special case of the computation of the second-order substrings.

B. A second-order substring is a suitable sequence of (a) a second-order substring head e.g., which, who, whom, while, etc., (b) a fixed sequence of first-order substrings, (c) any omittable first order substrings e.g., adjunct phrases, and (d) zero or more second-order substrings.

Examples:

1. In The man whom I saw on the street was my teacher, whom I saw on the street is a second-order substring where whom is a second-order substring head, I is a first-order substring $\bar{N}$, saw is a first-order substring $\bar{V}$, and on the street is a first-order substring P$\bar{N}$.

2. In The poet whom the people who lived in that town admired was a friend of mine, whom the people . . . admired is a second order substring in which another second order substring—who lived in that town is nested.

On account of the possible unlimited nesting of the second-order substrings the computation procedure differs from the one for the first-order substrings because while computing a second order substring K_1 we might meet another second-order substring K_2 whose computation must be finished first (unless we meet a third second-order substring K_3, etc.) before continuing the computation of K_1.

Second-order substrings will be marked off by < >.

B. A primitive well-formed sentence is a special case of a second-order substring in the sense that it is a fixed sequence of the first-order substrings but there is no substring head. In general there will be a set of primitive well-formed sentences. The most frequent one which we encounter (and which is the only one that will be treated in this short paper) is the sequence $\bar{N}$ $\bar{V}_i$ +i where $\bar{N}$ is a first-order substring (elementary noun-phrase) or a second-order substring which acts as a noun replacer e.g. < whatever [you] {said} > {is} (not) [true]; $\bar{V}$ is a first-order substring (elementary verb phrase), and the subscript i denotes a subset corresponding to $\bar{V}$ (a particular $\bar{V}$ may belong to one or more subsets), and +i means the necessary object corresponding to the subset i.

Examples:

sleep	V_{01}	object is zero—I sleep.
color	V_{04}	object is $\bar{N}$A—I color the kite red.
want	V_{11}	object is $\bar{N}$ to V +i—I want him to go.
attend	V_{21}	object is to $\bar{N}$— . . . attend to something . . .
base	V_{46}	object is $\bar{N}$ on $\bar{N}$— . . . base the conclusions on facts . . .

etc. *Other primitive well-formed sentence types are rare. One example is* +i $\bar{N}$ $\bar{V}_i$

D. After the computation of the first-order substrings we proceed as follows: We first replace the first-order substrings by single symbols e.g. the elementary noun-phrases by $\bar{N}$, the elementary verb-phrases by $\bar{V}$, P$\bar{N}$ phrases by Q and other adjunct phrases by O. The sentence now appears as a sequence of first-order substring symbols and some dictionary symbols (e.g. second-order substring heads) which were not included in any first-order substrings. We now start at the beginning of the sentence and proceed in a left to right direction. The second-order substrings are computed in the order in which they appear. If we come to a nested second-order substring we compute this first and then return to the original second-order substring. First order substrings $\bar{N}$ and $\bar{V}$ which are not a part of any second-order substring will be called free $\bar{N}$ or $\bar{V}$. A primitive well-formed sentence is $\bar{N}$ $\bar{V}_i$ +i. A well-formed sentence consists of a primitive well-formed sentence with possibly one or more second-order substrings (excluding those which act as noun-replacers and which were possibly required for the primitive well-formed sentence e.g. [I] {remember} <whatever [you] {did}>. Here the second-order substring <whatever [you] {did}> serves as the required object for the verb remember) and certain first order substrings (P$\bar{N}$ phrases, D etc.).

*Some second-order substrings have no recognizable head** e.g. The

*For the details of other types of second order substrings and their computation see paper 19 of the project.

book I bought disappeared. Here I bought is a second-order substring. In order to compute such second-order substrings we have therefore to keep a count of free $\bar{N}$s. Thus in The book I bought disappeared we have a free $\bar{N}$ followed by another free $\bar{N}$ and then a free $\bar{V}$ followed by another free $\bar{V}$. We see that the second $\bar{N}$ and the first $\bar{V}$ (minus its object) form a second-order substring and hence can be marked off if we keep the count of free $\bar{N}$s and $\bar{V}$s.

It is convenient to represent the computation in a tabular form as shown below. The computation of the second-order substrings and well-formedness begins at the top of column m. If we run into a nested substring we move to column m-1 and to m-2 if there is another nesting and so forth. In any given column m-i when the computation of that substring is finished we return to the column m-i + 1 and so forth.

Examples

1. [We] {will describe} [these results] (in [our next paper]).

Writing symbols for the first order substrings the sentence becomes

$$\bar{N}\ \bar{V}\ \bar{N}\ Q$$

We write this in column 1.

1	m-2	m-1	m	Remarks
$\bar{N}$			$\bar{N}$	First free $\bar{N}$
$\bar{V}$			$\bar{V}$	First free $\bar{V}$
$\bar{N}$			+	End of the object of $\bar{V}$
Q				Omittable Q
$				End of sentence; the sentence is well-formed.

2. [Those]<who {read} [newspapers] > {waste} [their time].

1	m-2	m-1	m	Remarks
$\bar{N}$			$\bar{N}$	First free $\bar{N}$
K 1		K 1		Substring head who
$\bar{V}$		$\bar{V}$		Free $\bar{V}$ inside the substring
$\bar{N}$		$\bar{N}$		End of object of $\bar{V}$ and of substring K1*
$\bar{V}$			$\bar{V}$	Free $\bar{V}$
$\bar{N}$			+	End of object of $\bar{V}$
$				End of sentence; the sentence is well-formed.

*Most second-order substrings end in $\bar{V}$ and hence the end of such a second-order substring coincides with the end of the object corresponding to the $\bar{V}$.

3. [They]{will consider} [these issues] (in [the next meeting]).

1	m-2	m-1	m	Remarks
$\bar{N}$			$\bar{N}$	First free $\bar{N}$
$\bar{V}$			$\bar{V}$	Free $\bar{V}$; required object types: $\bar{N}$ or $\bar{N}$ $\bar{N}$
$\bar{N}$			+	End of object type $\bar{N}$, since the next symbol is Q the other object type is not satisfied.
Q				Omittable
$				End of sentence; the sentence is well-formed.

If both object types were satisfied then we will have to carry out the computation for both the readings and see which one yields a well-formed sentence. If neither reading yields a well-formed sentence then the sentence will be a case of a nonwell-formed sentence.

4. [The conviction] {was based} (on [evidence]).

1	m-2	m-1	m	Remarks
$\bar{N}$			$\bar{N}$	Free $\bar{N}$
$\bar{V}$ (passive)			$\bar{V}$	Free $\bar{V}$ (passive) required object: $\bar{N}$ on $\bar{N}$; because $\bar{V}$ is in passive the object will be short of one $\bar{N}$.
Q			+	Q is required in the object; end of object.
$				End of sentence; the sentence is well-formed.

IV. THE QUESTION OF NONUNIQUE DECISIONS WHICH ARISE AT THE VARIOUS LEVELS OF COMPUTATION

The computation procedure described above appears to be too simple because we have so far avoided the set of possible nonunique decisions which arise at the various levels of computation.* We follow

*The nonunique decisions which arise in the activities of dictionary look-up, in the treatment of certain word complexes and in the possible resolution of some multiply classified words are briefly as follows. (a) In the dictionary some words bear two or more classifications. For example, study: N/V etc. (b) Certain word complexes (e.g. because of) are given the new classification (in this case P) as well as the old word-for-word classification (in this case C P). See Lila Gleitman's paper (Chapter 31) for further details. (c) See the footnote on p. 832 for the possible resolution of some multiply classified words and the nonunique decisions which arise in that activity.

the principle viz., that whenever we are faced with a set of nonunique decisions we follow a particular path and leave behind an indication of the alternative paths. The particular path which we will take depends on what level of computation we are at and the extent of the environment which we can survey at this level. Thus throughout the computation we are following a preferred path hoping that this path will result in a well-formed sentence. If the sentence fails to be so then we have to follow all the other paths one by one.* Even if the preferred path results in a well-formed sentence we still must exhaust all the other paths because it is likely (though rare) that the sentence will have another reading which is also well-formed.* In this case the sentence under consideration has a permanent ambiguity.

A. During the computation of the first order substrings nonunique decisions arise in many situations.† We will give only three illustrations.

1. On account of the analysis being local, it is sometimes not possible to state uniquely whether the opening noun-phrase bracket is to be placed to the left or to the right of a given word. For example, We proved that [impure preparation] was responsible for the failure of the experiment, and We used [that impure preparation] for the experiment. In the first example the opening bracket is to the right of the word that and in the second it is to the left of the word that. Since the decision whether to place the bracket to the right or to the left of that depends on the structure of the whole sentence and since this information is not available to us at this level we have to take one of the two decisions and keep a record of the other. The ambiguity will be resolved at a higher level of computation.

2. Another such situation in the computation of the noun-phrase arises when we come across a word bearing one or more class-marks. We follow the same general principle of choosing a preferred path and keeping a record of the alternative paths. The preferred path is indicated by the following consideration viz., that if a word one of whose class-marks, say α, would fit into [ ], occurs adjacent to [ ], then it is the α class-mark that will most probably hold for this occurrence of the word. We accept this reading and extend [ ] beyond this word i.e. we accept the value which maximizes [ ]. For example, In... $\underset{V/A}{\text{cool}}\ \underset{N}{\text{water}}$], we extend the noun-phrase beyond the word cool.

3. When we meet two words in succession bearing the class mark N then we are not sure at this level whether we have two noun-phrases ($\overline{N}$ strings) or one noun-phrase (one $\overline{N}$ string—NN i.e. a compound noun). Here again we prefer the path which gives the second reading (i.e. one $\overline{N}$ string) and keep a record of the alternative readings. This is consistent with the principle stated in the second illustration described above.

*Actually it is not required to follow each path completely. Some paths are ruled out at a very early stage of the computation.

†The detailed procedures of treating these various situations have been discussed in the papers 18 and 19 of the project.

B. During the computation of the second-order substrings we also have many situations where a set of nonunique decisions arise. The important case is the decision about the end of the substring. As most of the second-order substrings contain verbs, this becomes a question of where the verb object ends, e.g. the type and position of the substring permits the verbs to have either its full object or the short object.* If there is a possibility of satisfying both the requirements in a given sentence we choose the full object as the preferred reading but the short object must also be tried in the alternative reading. For example, [He] {was reading} [a book], and <while [he] {was reading}> [papers] {were flying} (everywhere).

We have a similar situation when a verb has more than one object type. If in a particular sentence we find that more than one of the object types can be satisfied we have to mark the end of each possible object type. The well-formedness computation must be carried out for each one of these readings (see example 3 in IIID).

C. Following is an example of the fact that certain nonunique decisions which arise during the computation of the first-order substrings can be resolved while computing second order substrings. For example to class can be a P$\bar{N}$ string as in [He] {went} (from [class]) (to [class]) or it can be a $\bar{V}$ string as in <{To class} [the manuscripts]> {is} (not) (always) [easy]. In the second example it is possible to consider the string to class as a $\bar{V}$ string because we find the necessary object ($\bar{N}$ string—the manuscripts) after it. This decision however can only be taken while computing second-order substrings. Incidently, the second-order substring in the second example is also a noun-replacer.

V. CONCLUSION

We have described briefly a procedure for computing the syntactic structure of a sentence of English. The activities can be summarized as follows: (a) Assign one or more class-marks to each word of the sentence; (b) assign a new classification for certain word complexes; (c) resolve some multiply classified words in favour of one of the classifications; (d) compute the first order substrings which are class-mark sequences with a definite structure for each type of the substring; (e) compute the second-order substrings which are fixed sequences of the first-order substrings and finally (f) compute the well-formedness of the sentence and decide whether or not the sentence is a case of a well-formed sentence.

*E.g. read has two object types viz., zero object and $\bar{N}$. The first one is the short object and the second one is the full object.