[*Proceedings of the National Symposium on Machine Translation*, UCLA February 1960]

Session 6: SYNTAX

GROUPING AND DEPENDENCY THEORIES

David G. Hays[1]
The RAND Corporation

Summary

Immediate-constituent analysis and dependency analysis (two theories of syntactic description) are based, respectively, on the topologies of grouping and of trees. A correspondence between structures of the two types is defined, and the two topologies are compared, mainly in terms of their empirical applications.

The two common methods of describing sentence structure (at the syntactic level) are immediate-constituent analysis and dependency analysis. The former, also known as phrase-structure analysis, is most often used by American linguists, but the latter is taught in high school. Phrase-structure theories underlie all MT systems being developed in the United States, except that of The RAND Corporation, while Soviet work on MT uses both theories.[2] The analysis presented in Figure la illustrates one method, that in Figure 1b the other. H. Hiž has recently presented a formal theory of grouping as a basis for immediate-constituent analysis,[3] the present paper defines a correspondence between groupings and the trees which are a basis for dependency analysis. Study of the correspondence reveals some similarities and differences between the methods; each has unique advantages in the study of syntax.

---

[1] I am grateful to Jane Pyne, H. Hiž, A. Madansky, and T. W. Mullikin for their criticisms and suggestions, which have helped substantially in the long, slow development of the material presented here. None is to be blamed for remaining errors.

[2] For examples of Soviet work using dependency theory, see the abstracts by O. S. Kulagina, I.I. Revzin, T. N. Moloshnaya, and Z.M. Volotskaya, Ye. V.Paducheva, I. N. Shelimova, A. L. Shumilina, in Abstracts of the Conference on Machine Translation, (May 15-21, 1958), translated by U. S. Joint Publications Research Service, Washington, D. C. , JPRS/DC-241, July 22, 1958.

[3] H. Hiž, "Steps toward Grammatical Recognition", Preprints of the International Conference for Standards on a Common Language for Machine Searching and Translation, Cleveland, Sept. 6-12, 1959.

Little	John	ate	breakfast
A	N	V	N
	NP	VP	
		S	

Figure la Immediate-constituent analysis

Little John ate breakfast

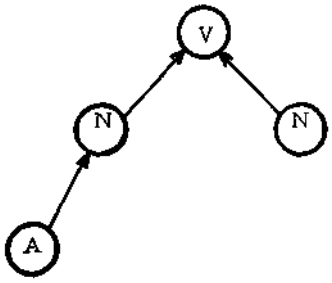

Figure 1b Dependency analysis

FIGURE 1 TWO ANALYSES OF A SIMPLE SENTENCE

Hiž axiomatizes parenthetic expressions (p. e. 's) and sets up an algebra for them. At the level of analysis chosen here, immediate-constituent analysis is described in mathematical terms as the repeated partitioning of a set of objects into equivalence classes. The objects represent the words in a sentence; they are customarily partitioned, first, into two classes: subject and verb phrase. Each of these classes is partitioned again, and so on.

Trees, on the other hand, arise by the introduction of a partial ordering over the elements of a set. Given two elements, either they do not compare at all, or one depends on the other, directly or indirectly; we call indirect dependence derivation--thus, for example, the object of a preposition derives from the word on which the preposition depends, and an adjective modifying the subject of a verb derives from the verb. If there is a unique element from which all others derive, and if no element depends on two others, the partial

ordering can be displayed by a branching diagram, or tree.

Two major properties of language are not discussed in this paper; not reflected in the mathematical structures treated here. Words belong to types or classes: verb, noun, and so on. The elements analyzed here have only the properties given by their positions in p.e.'s or trees. The elements of a sentence are simply ordered, i.e. , can be mapped onto the first several integers. The elements of a parenthetic expression can be ordered, by an apparatus that Hiż introduces, and the elements of a tree can be ordered by an apparatus that we plan to introduce in later papers. For the present, it is best to omit consideration of both topics, making the two types of structures as simple as possible for clarity of comparison.

1. ((*) (*) (*) (*))
2. ((*) (*) ((*) (*)))
3. (((*) (*)) ((*) (*)))
4. (((*) (*) (*)) (*))
5. ((((*) (*)) (*)) (*))

FIGURE 2 PARENTHETIC EXPRESSIONS

If a tree and a p. e. are to serve as alternative models for the same sentence, they must have equal numbers of elements. In Figure 2 are presented the five distinct p. e. 's, order being disregarded, that have four elements each. Four distinct trees can be drawn with four nodes each; they are shown in Figure 3. We take it for granted, with types and order disregarded, that the following sentences are all modeled by p. e. 3:

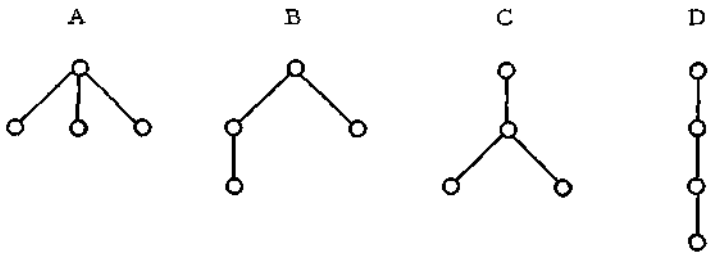

FIGURE 3 TREES CONTAINING 4 NODES

He ate lunch slowly. John ate green apples.

That little boy ate. Very good children eat.

(The underscorings correspond to p. e. 's.) Although, these sentences have the same grouping structure, their trees are different. "He ate lunch slowly" is described by tree A, "John ate green apples" by tree B, "That little boy ate" by tree C, and "Very good children eat" by tree D. Obviously, tree structures capture something of syntax that is lost by grouping.

On the other hand, it is easy to construct a set of sentences with a fixed tree structure and various groupings. For example, these two sentences are described by tree B:

Little John ate breakfast.

He ate his breakfast.

The first has grouping 3, the second grouping 5. Grouping, therefore, captures something of syntax that is lost by tree structures.

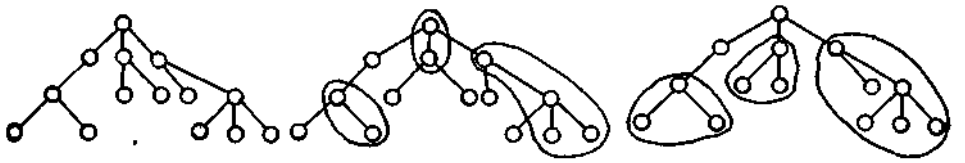

Fig. 4a - A tree Fig. 4b - Subtrees Fig. 4c - Complete subtrees

FIGURE 4 TREES, SUBTREES, AND COMPLETE SUBTREES

Let us formalize the correspondence. A subtree is any connected set of nodes contained in a tree; Figure 4a shows a tree, and Figure 4b shows some of its subtrees. A complete subtree consists of any node together with all nodes that derive from it, as illustrated in Figure 4c. Every complete subtree is a subtree, naturally. Any tree is a complete subtree of itself; a proper complete subtree is a complete subtree that does not contain the unique original node of the parent tree. The correspondence can now be defined.

Definition. A parenthetic expression α corresponds to a tree β (and vice versa) if there exists a one-to-one mapping of the elementary parenthetic expressions in α onto the nodes of β, such that (i) every parenthetic expression contained in α maps onto a subtree of β, and (ii) every complete subtree of β is mapped onto a parenthetic expression in α by the inverse mapping.

The correspondence is illustrated, for trees with 4 nodes, in Table 1. The marginal labels in the table are taken from Figures 2 and 3.

TABLE 1

Correspondence matrix:

Parenthetic expressions and trees of 4 elements

Parenthetic Expression	Tree			
	A	B	C	D
Class 1	1	0	0	0
Class 2	1	1	0	0
Class 3	0	1	0	0
Class 4	1	0	1	0
Class 5	1	1	1	1

1: Correspondence

0: Non-correspondence

The effect of condition (ii) is illustrated in Figure 5. First we show, in Figure 5a, a mapping of p. e. 4 onto tree A. Here index numbers are applied to reveal the one-to-one mapping. Next, in Figures 5b and 5c, we show two mappings of the same parenthetic expression onto tree B. In the first, the subtree consisting of nodes 1, 2, and 3 contains a proper complete subtree (nodes 2 and 3); as we

Figure 5a Mapping of a parenthetical expression onto tree A

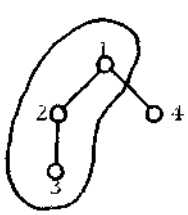

Figure 5b Tree B; first mapping

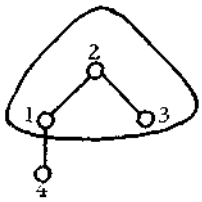

Figure 5c Tree B; second mapping

FIGURE 5 MAPPINGS

might say, there is organization here that is not reflected in the parenthetic expression, whose elements 1, 2, and 3 stand in mutually symmetric relation. In the second mapping (Figure 5c), the connection between nodes 1 and 4 is not agreeably represented by the grouping of the parenthetic expression, since the linguistic application proposed for both structures demands that the sequence of containments in a p. e. run parallel to the partial ordering of nodes in a corresponding tree. In other words, on linguistic grounds, if node 4 depends on node 1, and node 1 on node 2, every p. e. containing nodes 2 and 1 should also contain node 4. For, in Figure 5c, nodes 1 and 3 must represent modifiers or complements of the element represented by node 2, and node 4 represents a modifier of that represented by node 1. In such a situation, the immediate-constituent analyst would always group elements 1 and 4, then built them into a larger structure.

Constructive rules can be given for going from a p. e. to all corresponding trees, and vice versa. First, a given tree, to construct a corresponding p. e. : Embed a tree in a plane, and project each node onto a line in such a manner that no projection line intersects a connection. (See Figure 6.) For each node in the tree, insert parentheses in the line enclosing the projections of all nodes that derive from the given node.

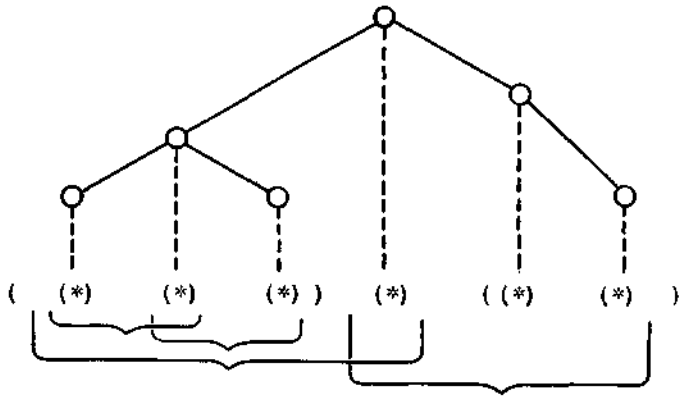

FIGURE 6 CONSTRUCTION OF P. E. FROM A TREE

Make any desired (or all possible) permutations, transposing the p. e.'s within any single p. e. Certain optional parentheses can still be inserted. Consider any p. e. , say a_1 . It consists of n p. e.'s, surrounded by parentheses, a_{i1}, a_{i2}..., a_{in}. If n = 2 , no additional parentheses can be inserted; if n > 2, parentheses can be inserted by the following rule. The p. e. a_i is the projection of a complete subtree, with a unique original node, say X . Now the projection of X is in one of the a_{ij} , say a^1_{ij} . Choose any 2, 3, . . . , n-1 of the a_{ij} , <u>including</u> a^1_{ij} , transpose if necessary to make them contiguous; and set parentheses around them. The operation can be repeated freely on any p. e. containing, after all previous applications of the operation, 3 or more p. e, 's. In Figure 6, brackets below the projection line indicate some p. e. 's that are allowed, but not required, by the partial ordering.

Second, given a parenthetic expression, to construct a corresponding tree: Delete the first and last parentheses; the p. e. is thus converted into 2 or more p. e.'s. Choose one of the resulting p. e.'s

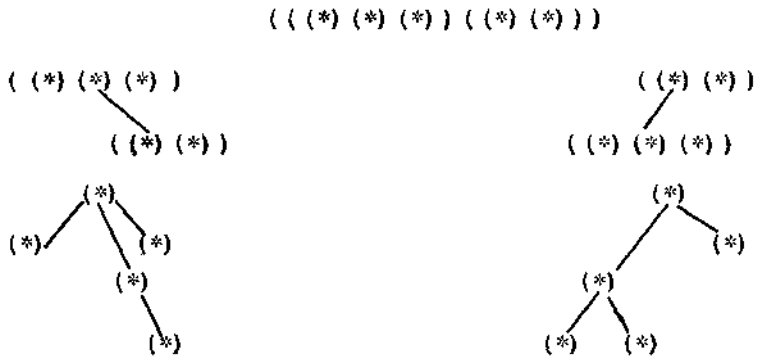

FIGURE 7 CONSTRUCTION OF TREES FROM A P.E.

and construct a tree whose unique original node corresponds to the chosen p. e. All the other nodes depend on the first, and correspond to the remaining p. e. 's. If every node in the tree corresponds to an elementary p.e. , the operation is complete. Otherwise, choose any node and examine the corresponding (non-elementary) p. e. Delete the first and last parentheses, choose one of the resulting p. e. 's and maintain its correspondence with the chosen node. Add to the tree a

node corresponding to each remaining p. e. , letting the new node depend on the chosen node. This operation is continued until every node corresponds to an elementary p. e. Each sequence of choices leads to a tree, but not all the trees are distinct.

We can now state summarily what trees and p. e. 's capture uniquely. In a tree, a node can have several dependents; its relations to them are equivalent. In the corresponding p.e. 's their relations can be ordered by degree of closeness. On the other hand, a p. e. consists of several p. e. 's; in a tree, one of these p. e. 's dominates the rest.

Let us conclude with some examples from natural language. In a sentence such as "Children love candy", whose form is N-V-N, immediate-constituent analysis groups verb and following noun into a verb phrase, rendering the sentence as N-VP. Dependency analysis makes the two nouns dependents of the verb. A passive transform, "Candy is loved by children", with the form N-is V (ed)-by N, would be grouped into N-VP. Note that the groupings reflect grammatic properties clearly enough, but disregard meaning; "candy" goes into VP one time but not the other. Constancy of meaning behind the grammatic transformation is reflected more clearly, as we believe, by two trees, in which "children" and "candy" are dependents of "love" in both active and passive forms of the sentence.

Again, consider the naming of phrases. An adjective plus a noun form a noun phrase, and an adverb plus an adjective form an adjective phrase. The naming singles out an element of each phrase, as does the topology of a tree. Grouping - e. g., ((A)(N)) - does not.

Neither parenthetic expressions nor trees capture all that the linguist wants to say about sentences. Beginning with either, he requires ancillary apparatus to complete his description. What is natural and inherent in one theory has to be appended to the other; immediate-constituent analysis introduces phrase names[4] to handle a property of language that is reflected in inherent properties of

[4] Several MT systems have been projected in which sequences of sentence elements of given types are replaced by phrase units of given types, until the sentence is reduced to N-VP = S. Cf. Victor A. Oswald and Stuart L. Fletcher, "Proposals for the Mechanical Resolution of German Syntax Patterns", Modern Language Forum, vol. 36, No. 3-4, 1951, pp. 1-24.

trees, whereas dependency analysis must introduce a treatment of linear order which is considerably more complicated than Hiž's treatment. If the present paper emphasizes the advantages of dependency theory, it is because the pre-eminence of immediate-constituent analysis in American linguistics has made its virtues widely known; the two methods deserve more penetrating comparison than they are given in current text books.