Work Papers of the Summer Institute of Linguistics, University of North Dakota Session

Volume 19 Article 2

1975

A new look at new nodes: Scope of predication and surface structure parsing in natural language

Lon Diehl
SIL-UND

A NEW LOOK AT NEW NODES: SCOPE OF PREDICATION AND SURFACE STRUCTURE PARSING IN NATURAL LANGUAGE

Lon Diehl

This paper is an attempt to point out, however sketchily, a striking distribution of order regularities in the non-argument material of strings in natural language surface structure. A tentative attempt is made to provide for this apparently diverse range of related phenomena a unitary characterization.

Among the seven principles which John Kimball sets forth in his characterization of sentence acceptability (as opposed to grammaticality) are the following two:

(1) Principle One (Top-Down): Parsing in natural language proceeds according to a top-down algorithm.

(2) Principle Three (New Nodes): The construction of a new node is signalled by the occurrence of a grammatical function word.[1]

The representations upon which these parsing algorithms operate are of course constituency trees, or phrase structure markers. Abandonment of constituency trees in favor of dependency trees will be the first of several modifications which I suggest will make possible our unitary account of what seems otherwise to be difficult to show to be related. Excluded from our discussion will be the consideration of order among arguments with respect to each other and with respect to non-argument material; for a treatment of argument-related order in natural language see Diehl 1975a. Also excluded will be most matters relating to order internal to arguments, or within NP.

We will not go into detail here to justify the use of dependency trees instead of constituency trees. Instead we will adopt dependency representation without repeating such discussion as can be found in Anderson 1971: 27-31; Diehl 1975: 45-57; and Lyons 1968: 230-231, 234-234, 330-333. While the pertinent value of dependency trees will be clear shortly, the following assumptions, implicit in my use of the trees, should be noted:

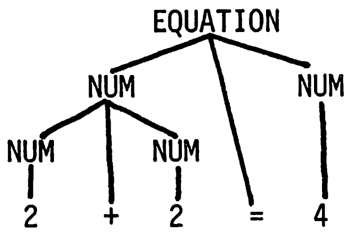
(3) a. non-complex verbs each take at least one argument (IT) and at most two arguments (IT and AT, where AT itself in the case of a verb of motion or transition as opposed to static position or state may manifest as a pathway defined in terms of FM and TO (cf. Jeffrey S. Gruber (1965)'s <u>theme</u> (IT), <u>location</u> (AT), <u>source</u> (FM), and <u>goal</u> (TO)).

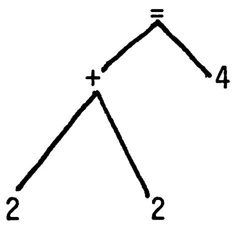
b. there are no non-terminal nodes: the head of the constituent (e.g., V in S or N in NP) is the node from which depend those elements related by it (e.g. the arguments of a predicate).[2]

c. each natural language is either deeply IT-AT or AT-IT; i.e., a natural language is either recursively IT-AT or AT-IT. Clause types are given exemplification in Diehl 1975: 18-19.

Dependency trees together with equivalent linear representation are crudely illustrated in the following contrast with constituency trees and linear bracketing as applied to an elementary arithmetical equation:

(4) CONSTITUENCY TREE DEPENDENCY TREE



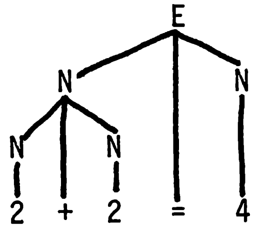


PHRASE BRACKETING LINEARIZED DEPENDENCY

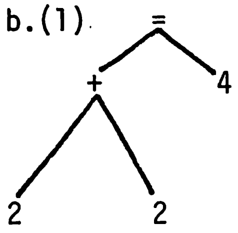
```
[ [ [2]+[2] ] = [ 4 ] ]        = ( + ( 2 , 2 ) , 4 )  OR
E N N N N N N   N   N E        ( ( 2 , 2 ) + , 4 ) =
```

The following difference between constituency representation and dependency representation should be noted: One constituency tree has one corresponding linearization (phrase bracketing), while one dependency tree has two or three corresponding linearizations.

a.(1)

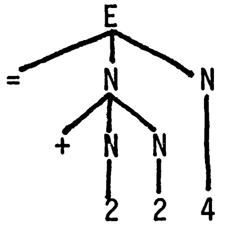


b.(1)

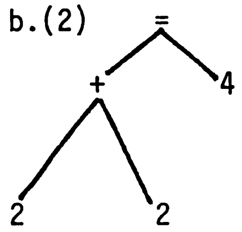


```
[ [ [2] + [2] ] = [ 4 ] ]        ( ( 2 + 2 ) = 4 )
E N N N   N N N   N   N E
```

a.(2)

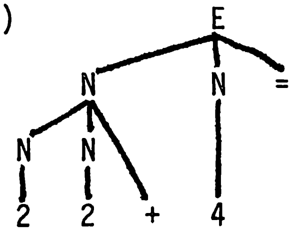


b.(2)

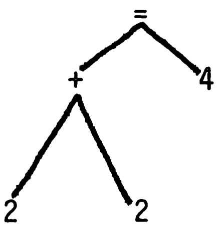


```
[ = [ + [ [2] [2] ] [ 4 ] ]        = ( + ( 2 , 2 ) , 4 )
E   N   N N N N N N N   N E
```

a.(3)



b.(3)



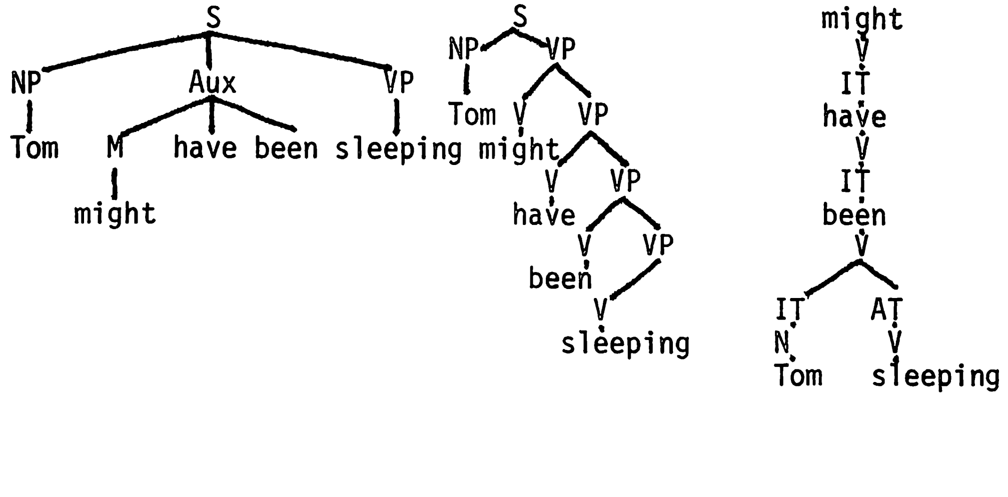
```
[ [ [2] [2] + ] [ 4 ] = ]        ( ( 2 , 2 ) + , 4 ) =
E N N N N N   N N   N   E
```

Dependency tree linearizations will be regarded to be two and not three. Any dependency tree yields both the "Polish notation" version (where each functor precedes everything in its scope) and its mirror image (where each functor follows everything in its scope). These correspond respectively to reading "down" and reading "up" the given dependency tree. The third apparent alternative is to read across the dependency tree ((4)b.(1)). This, however, cannot be taken seriously as a basic linearization; one reason is that it would fail to give any ordering to the one-place predicates with respect to its one argument, forcing either a top-down or bottom-up reading. In other words, since dependency linearizations will always require possible top-down and bottom-up readings only those two are considered basic.

Another difference between constituency and dependency representations is that dependency representation (whether tree or linearized) implies the corresponding constituency representation, while the converse does not hold. In other words, constituency is always recoverable from dependency. For example, S (Prop) may be defined as V (Pred) together with all that depends (or with everything in its scope); NP (Arg) as N together with all that depends.

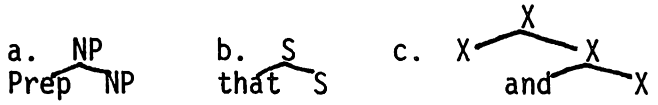
However, the most immediately pertinent property of this dependency representation is that it allows (forces) a non-ambiguous scope hierarchy among all non-argument material in the trees. Below we display two constituency trees (Kimball's (29a-b)) with a dependency tree for the sentence Tom might have been sleeping:

(5) a. Chomsky 1957   b. Ross 1967b   c. (dependency tree)

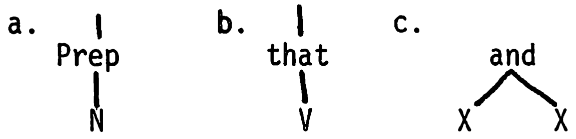


Note that both (5c) and (5b) capture the hierarchy of scope with predicate stacks.

The application of New Nodes to constituents in English is illustrated in part by the following grammatical functors: (a) prepositions, (b) complementizers, (c) conjunctions ((6) = Kimball's (24)):

(6)


The equivalent relations in dependency trees are shown in (7):

(7)


Kimball, speaking in terms of his seven principles as they operate upon constituency trees, makes the following observations, hedging, in effect, or weakening New Nodes (p. 33):

> The question of New Nodes in SOV languages needs further examination. In such languages grammatical formatives typically follow those constituents to which they are attached.... For cases where the constituent is a simple NP with a post-position, the principle could be operative, as this NP could be stored until a look ahead to the post-position gave clue to its syntactic status. For large constituents such as S's with following complementizers, New Nodes simply is inoperative. Note, however, that such cases are not counter examples; New Nodes has the logical form of a conditional: If a grammatical function word occurs it signals construction of a new phrase.

Note the following:

(8) a. (boku wa) John ga baka da to omou (cf. Kuno 1973:5(14))
I fool is that think

b. I think that John is a fool.

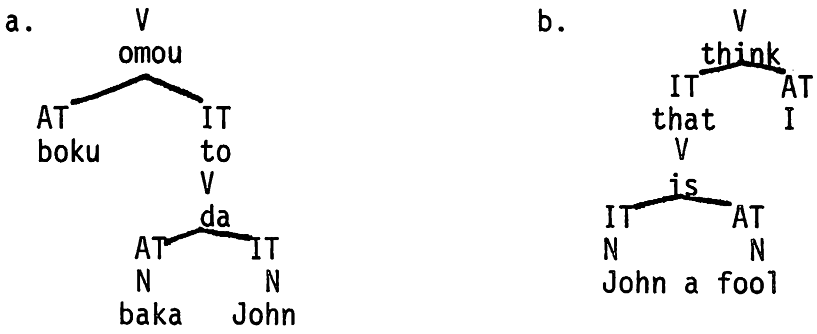
In (9) we see the respective linearizations; the direction of linearization will follow that found in the respective surface structures of Japanese and English, except that scope representation will require that non-predicate arguments (i.e., nonsentential NP) appear "out of place".

(9) a. (boku, (baka,John da) omou
I fool is think

b. think (is (John, a fool) , I)

Ignoring the order of argument material, we see that, while the linear orders of the predicate sequences in Japanese and English in the sentences above are in a mirror-image relationship, they share an *identical scope hierarchy*, i.e., the same dependency configuration. Furthermore the complementizers *to* and *that* are equivalently ordered.

(10)



Starting with the above pair of trees and linearizing back into the respective surface orders we immediately see that the difference can be seen as the result of performing linearization on essentially the same hierarchy, in the one case going bottom-up and in the other top-down. Even the complementizers *to/that* come out ordered correctly, which is suggestive of the status of New Nodes as now being subsumed under the notion Node Marker, which, in top-down languages must of course have the effect of "introducing" constituents and likewise in bottom-up languages cannot do other than "follow" the constituent named and bounded by it.

Although the above examples are extremely simple (as well as few), I strongly suspect that the principle they illustrate has wide application over a diverse range of ramifications.

One example of pushing this notion is its application not only to all sorts of stacks of lexical verbs (including auxiliaries) but also to the components of "complex" predicates such as *kill* when analyzed as follows:

(11) CAUSE((COME(BE(X,DEAD))),Y)
CAUSE((BECOME(DEAD,X)),Y)
CAUSE((DIE(X)),Y)
KILL((X),Y)

If our generalization about "universal scope hierarchies" is really correct, then it would be nice if such alternations of lexical incorporation (granting the analysis) showed that the different "components" behaved as though they too were stacked according to scope hierarchy. The English given in (11) above suggests that a bottom-up language should find the lexicalized version of CAUSE after any of the other elements involved. In Jinghpaw (spoken throughout NE Burma) we have fairly strict SOV order, suggesting that, if there are any differentiated incorporations equivalent to any two of the representations in (11), the bottom-up routine should give us the final position for the CAUSE equivalent. In fact, sat is equivalent to kill, and equivalent to CAUSE(DIE) we have si shangun where si means "die" and shangun translates as "to cause to"; the Japanese equivalent (now morphologized) works the same way.

The payoff for adopting dependency trees and linearizations is that we gain a directed highly restrictive representation of how non-argument material is linearized in a natural language: one (or a systematic combination e.g., German) of two possible linearizations of scope: "inward" (top-down) or "outward" (bottom-up). This representation, taken together with the restriction to two linearization routines, permits us to capture in a direct way a generalization which relates the following phenomena with an automatic (forced) characterization.

(12) a. direction of predicate scope linearization:
(1) inward (=top-down) vs. (2) outward (=bottom-up)

b. relative order of verb and object:
(1) VO vs. (2) OV

c. position of complementizer with respect to embedded clause (S):
(1) before vs. (2) after

d. position of coordinate conjunctions with respect to conjuncts:
(1) before vs. (2) after

e. position of subordinating conjunctions with respect to subordinate clauses:
(1) before vs. (2) after

f. position of deep case marking (see Diehl 1975) with respect to the NP marked (either pre- vs. post-positional particles or as focus of productive inflectional morphology):
(1) before vs. (2) after

It is claimed then that all these phenomena are correctly and automatically predicted in terms of reading up or down on the same dependency tree, and that finding any one of (12)a-f in any natural language should enable the field linguist to correctly predict all the others. Implications for transformational theory (e.g., the relatively high acceptability (or eased processing) of EXTRAPOSITION output in top-down languages) will be held for later.

The principle we have been discussing, functor scope iconicity (FSI), a derivative of linkage iconicity (Diehl 1975) as LI applies to logical structure, relates (derives?) the following Greenberg Universals:

3, 4, 9, 11, 12a., and 16.

Finally, we combine functor scope iconicity with the two basic argument scope orders (IT-AT and AT-IT, or IA and AI, see Diehl 1975), to generate a typology of natural languages in terms of ordering routines:

(13)	A.	B.	C.	D.
	TD (top-down	ID	BU (bottom-up)	BU
	AI	IA	AI	IA

Type A is represented by languages such as Chinese; type B by languages such as English, French, Thai, etc.; type C by Japanese, Burmese, and other SOV languages. As far as I have been able to find out, type D is not represented among natural languages. That this should be the case is automatically predicted by the interaction of the independently motivated universal (because cognitive principles described in Diehl 1975 (i.e., MS taken together with the iconicities). Specifically, any Bu (bottom-up) language is an OV language, and the principles specify NSIOV as the only possible basic order for any OV language, automatically excluding the IA possibility.

NOTES

1. There is a traditional distinction in the discussion of the parts of speech between what are called content words (nouns, verbs, adjectives, etc.). In the literature of transformational grammar, this distinction surfaces in terms of the difference between lexical formatives and grammatical formatives. For the time being I will focus on just prepositions, wh-words, (e.g., what, where, who, how, when, why, etc.) conjunctions, and complementizers (that, for-to, and pos-ing)...

   There is syntactic evidence that grammatical formatives are Chomsky adjoined on surface structure (cf. Ross, 1967). (The assumption that this is the case is in fact not necessary to the correct operation of New Nodes, but I shall maintain the assumption in that which follows.) Thus, what is traditionally called a prepositional phrase is in fact a NP, as in (24a), and the complementizers and conjunctions appear on the surface structure as in (24b,c).

   Kimball, p. 29.

2. For analyses of complex clause types, see Diehl 1975: 18-19.

BIBLIOGRAPHY

Anderson, John M. 1971. The Grammar of Case: Towards a Localistic Theory. Cambridge Studies in Linguistics 4. Cambridge, England: The University Press.

Diehl, Lon. 1975. "Space Case: Some Principles and their Implications for Linear Order in Natural Language". (coursework paper, Indiana University).

Greenberg, Joseph H. 1963. "Some Universals of Grammar with particular reference to the order of meaningful elements". J. H. Greenberg (ed.) Universals of Language. Cambridge, Massachusetts: MIT Press. also in Keiler (ed.) 1966. A Reader in Historical and Comparative Linguistics. New York: Holt, Rinehart, and Winston.

Kimball, John. 1974. "Seven Principles of surface structure parsing in natural language". Cognition 2(1) pp. 15-47.

Kuno Susumu. 1973. The Structure of the Japanese Language. Current Studies in Linguistics Series No. 3. Cambridge, Massachusetts MIT Press.

Langacker, Ronald. 1974. "Movement Rules in Functional Perspective". Language Vol. 30 No. 4(1) pp. 630-664.

Lyons, John. 1968. Introduction to Theoretical Linguistics, Cambridge, England: The University Press.

Mathias, Gerald. 1975. personal communication.