

Prospects for top-down derivation

Jan-Wouter Zwart

Merge: the derivation of structure and order
(June 5, 2008, Groningen)

Simplest Merge

Every derivation of syntactic structure needs (a) a set of elements N manipulated in the course of the derivation, called 'numeration', and (b) a procedure establishing relations among the members of N , called 'merge'. Simplicity considerations then demand:

- (1) *Simplicity requirements on the derivational procedure*
 - a. merge manipulates a single element of N at each step of the derivation
 - b. merge manipulates each element from N only once

These requirements are not met in standard conceptions of the derivational procedure, which describe merge as an operation combining two elements (hence manipulating more than one element), and which allow a merged element to be merged again ('internal merge', i.e. movement). While these deviations from the simplicity requirements seem minimal or perhaps unavoidable, it should be pointed out that they introduce stipulations unwanted in a truly minimalist approach.

Thus, a derivational procedure that allows merge to manipulate two elements at a single step in the derivation can disallow merge to manipulate more than two elements only by stipulation (since 2, unlike 1, is not the absolute minimal number). This stipulation is nevertheless needed, if the system is supposed to derive only binary branching structures. Likewise, a procedure that allows a merged element to be moved (remerged) essentially states that one of the two elements manipulated maybe contained within the structure being derived. But then the possibility that the other element being manipulated at that step of the derivation is also contained within the structure being derived can be excluded only by stipulation (the 'extension condition'). Yet this stipulation is needed, if we want the system to be unable to derive bizarre and in fact endlessly looping structures not attested in human language.

Adhering to the simplicity requirements in (1), then, eliminates seemingly inevitable stipulations and yields a closer match between the structures generated by the derivational procedure and the actual phenomena of human language.

Proposal

Concretely, I would like to propose that each step in the derivational procedure turns the numeration N (a set, or, perhaps more appropriately, an array) into the ordered pair $P = \langle x, y \rangle$, where $x \in N$ and $y = (N - x)$. In other words, merge splits N into a pair consisting of a (designated) member of N and its residue in N . The numeration, then, is reduced at each step, and each next step in the derivational procedure targets the residue of the numeration created by the previous step, until N is empty. This procedure meets the requirements in (1): a single element from N is split off from N and further operations target the residue of N .

Structure

The system described here takes the derivation to be a procedure that transforms an unordered or unstructured collection of elements (i.e. a set or an array) into a structure. This differs slightly but significantly from the standard view, which takes merge to involve a transferring process taking elements out of a resource (the numeration) into a structure.

The derivational procedure operates in a top-down fashion, in the sense that the first pair created corresponds to the highest pair in the tree structure representation. This implies that properties of lexical heads (having to do with argument structure or subcategorization features)

play no role in driving the derivation.

Top-down derivations within generative grammar have been proposed and defended earlier, most notably by Phillips (2003) and Chesi (2007). In Phillips's system, merge adjoins material to the most deeply embedded right branch of the structure, essentially splitting that branch by the addition of new material from the numeration. This system differs from the one contemplated here in that it takes merge to be an operation importing material into the structure from some resource, as in the more traditional bottom-up derivational system.

Order

In the system proposed here, ordering is not a function of features of the elements merged, but a function of the derivation itself. More precisely, the circumstance that the steps in the derivation are temporally ordered yields an ordering of the elements affected by these steps.

The relation between the temporal ordering of the steps in a derivation and the linear ordering of the elements involved in the derivation can be made precise in the following way. Adopting the derivational system in (4), we observe that the set of elements merged (i.e. split off from the numeration) grows with each step in the derivation. If we consider only the first two steps, the sets of elements merged at each step are:

- (2) 1. after the first step: { a }
 2. after the second step: { a, b }

The set of sets of elements merged after step 2 is:

- (3) { a, { a, b } }

which is the set-theoretical notation of the ordered pair $\langle a, b \rangle$. It follows that after a sequence of steps we obtain a set of sets of elements merged, which is equivalent to an ordered n-tuple as in (4):

- (4) { a, { a, b }, { a, b, c }, { a, b, c, d }, { a, b, c, d, e } }

The idea of deriving order from nested sets originates with Fortuny (2008) (though details of the implementation differ). As Fortuny (2008) shows, the actual linear order of the words and phrases involved in the derivation, which is established at the interface component dealing with sound, may be derived straightforwardly from the output of the derivational procedure if that output is an ordered n-tuple. The simplest implementation of that idea appears to be (5):

- (5) *Linear Correspondence Axiom (revised from Kayne 1994)*
 $\langle \alpha, \beta \rangle = / \alpha \beta /$

Consequences

This paper discusses a number of consequences of the top-down derivational approach, having to do with the definition of syntactic relations, morphological dependency, and the phenomena commonly described in terms of movement.

References

- Chesi, Cristiano. 2007. Five reasons for building phrase structure top-down from left to right. *Nanzan Linguistics Special Issue 3*, 71-106.
Fortuny, Jordi. 2008. *The emergence of order in syntax*. Amsterdam: Benjamins.
Kayne, Richard. 1994. *The antisymmetry of syntax*. Cambridge: MIT Press.
Phillips, Colin. 2003. Linear order and constituency. *Linguistic Inquiry* 34, 37-90.