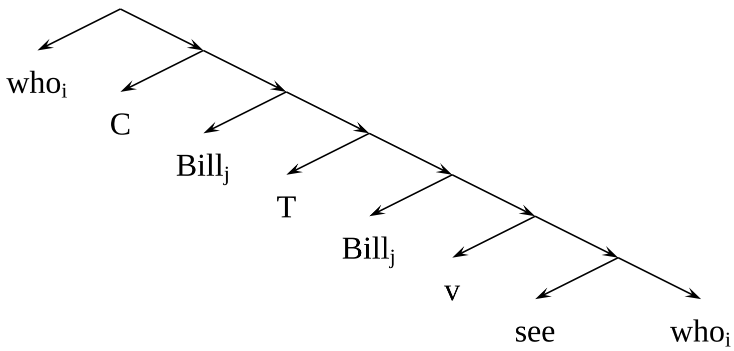
Copies, Traces, Occurrences, and all that

Evidence from Bulgarian multiple *wh*-phenomena[1]

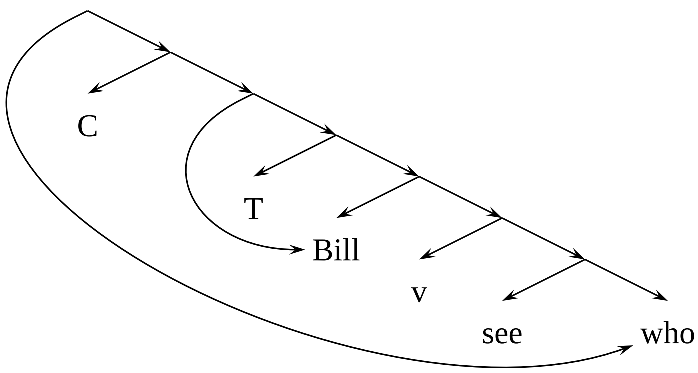
John Frampton
Northeastern University
Second draft, September 2004

One of the guiding principles of Chomsky's "Bare Phrase Structure" (1995) was the idea that syntactic operations do not introduce new symbols into the representation. The notion of a "movement trace" ran counter to this idea. Even with the later "copy theory of movement", some kind of coindexing had to be introduced to tie the trace together with the moved element. One way around this problem was to think of the representation (1a) of *Who did Bill see?* as actually being a stand-in for the representation (1b). (Here and throughout this note, head raising will be ignored. The general framework proposed can be easily generalized to include head raising.[2])

(1) a.



b.



Although (1b) has obvious conceptual advantages, the idea has languished, primarily because it has never been possible to establish empirical differences between (1a) and (1b). Secondarily, particularly in more complex structures, there are typographic obstacles to the use of representations like (1b).[3] In the face of the typographic problems, with no empirical advantages to point to, multiple parent representation have not been worth the trouble. See however Gärtner (2002) for a notable effort to develop this idea. Kracht (2001) also considers multiple-dominance trees.

Richards' recent (2004) note on multiple *wh*-movement in Bulgarian, in which he defends the possibility of lowering operations in syntax, finally provides good empirical evidence for relaxing the single parent condition on syntactic representations. What appears to be lowering from the standpoint of single parent representations turns out to be a simple consequence of the natural form taken by the recursive linearization algorithm for multiple-dominance trees.

1. Linearization of Multi-Dominance Trees

Assume that linearization is by phase, with the construction of CPs triggering phasal linearization. After C is merged and the syntactic C cycle has run its course, the complement of C is linearized. Finally, at the top level, CP is linearized. This is standard derivation by phase theory, with a sparse array of nonfinal phases (only C-complements). Following ideas developed in Frampton, Gutmann, and Legate (forthcoming), I assume also that linearization in each phase is top-down and recursive. The Default Linearization Algorithm (at least for VO languages) is given in (2). The symbol + represents ordered concatenation.

(2) Default Linearization Algorithm (DLA)

If γ is the merger of α and β, then

$$\text{Linearization}(\gamma) = \text{Linearization}(\alpha) + \text{Linearization}(\beta)$$

if α is a specifier or β is a complement; except that

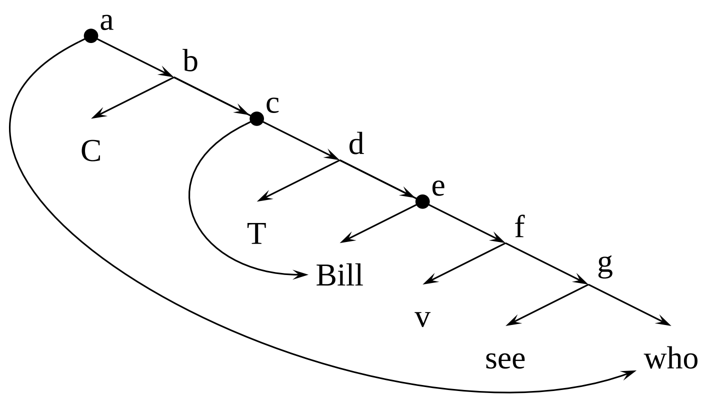
1) if β is a C-complement, then Linearization(β) is taken to be the result of the prior phasal linearization; and

2) the linearization of δ (either α or β) is omitted if δ has a parent outside of γ.

The qualification "default" is intended to imply that variations on default linearization are a possibility. Variations are not relevant to the discussion in this paper.

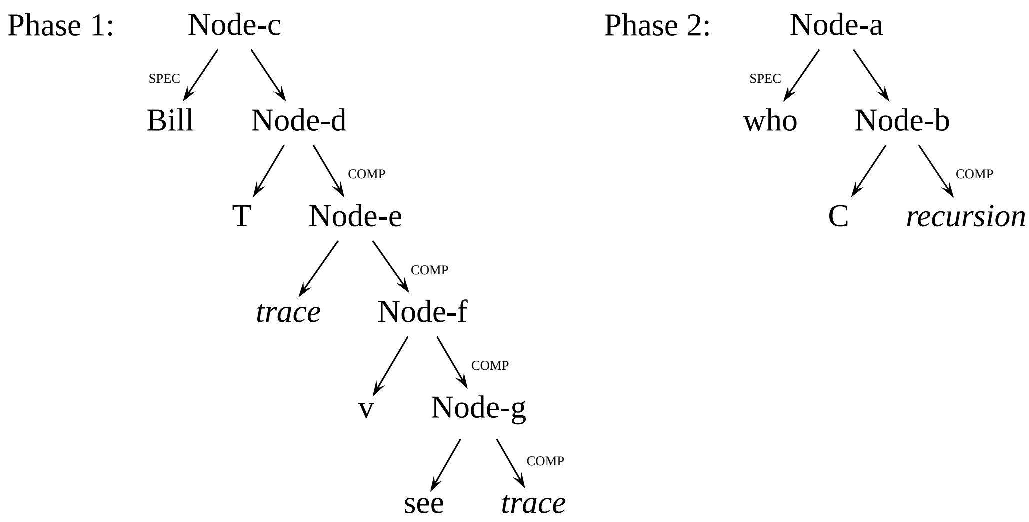
The operation of the DLA on (2b) can now be computed. Nodes with specifiers are indicated by a small black disk in (3) and complements are displayed on the right. I take no position here on how the syntax encodes the complement and specifier relations. Regardless of how it is encoded, this is information that the DLA needs to carry out its computation. Additionally, node labels are given to some of the nodes in (3) so that the DLA computation can be described. This is only for exposition. No labeling is needed in the syntax.

(3)



The recursion tree in (4) is a record of the computation carried out by the DLA operating on (3). It is *not* a syntactic structure.

(4) Recursive linearization of (3b)/(4a)



Specifiers go to the left and complements to the right, as specified by the linearization algorithm. For historical reasons, we use the word "*trace*" to indicate the steps in which the linearization of a node is omitted. The notation "*recursion*" is meant to indicate the result of linearization in the previous phase.

The DLA is the simplest linearization algorithm which satisfies the requirement that nodes should be linearized once and only once. The only other simple algorithm is top-down linearization without phases, skipping the linearization of nodes which have previously been linearized in the computation. But linearization by phase makes it impossible to even attempt this approach since in that case some nodes are encountered in low positions before they are encountered in the high position in which they are to be spelled out.

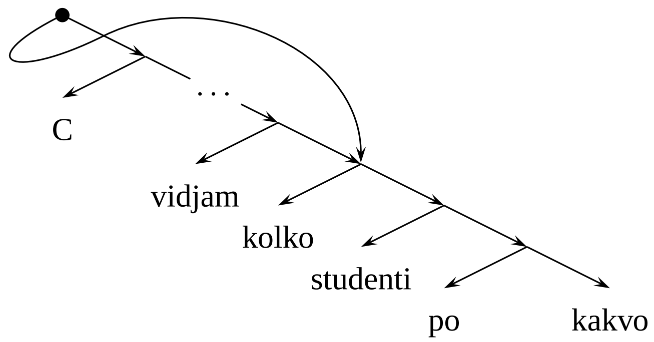
We now consider the involuted *wh*-extraction structures in Bulgarian considered by Richards (2004). The examples in (5) are from his paper.

(5) a. Kolko studenti po kakvo vidja?
how many students of what you saw
How many students of what did you see?

b. Po kakvo kolko studenti vidja?
of what how many students you saw
How many students of what did you see?

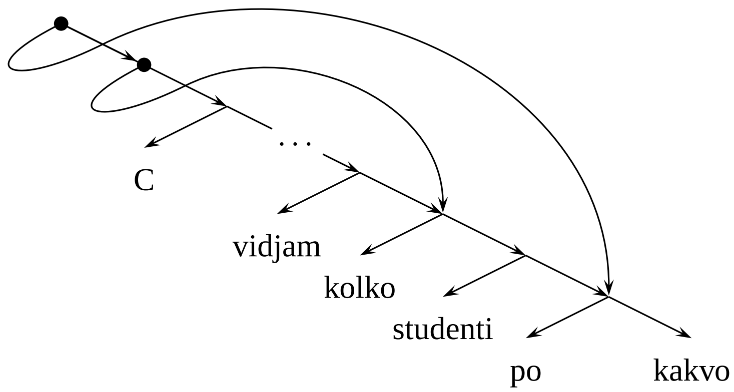
Following Richards, suppose that the higher *wh*-phrase (*kolko studenti po kakvo*) raises first, creating a Spec-CP position:

(6)

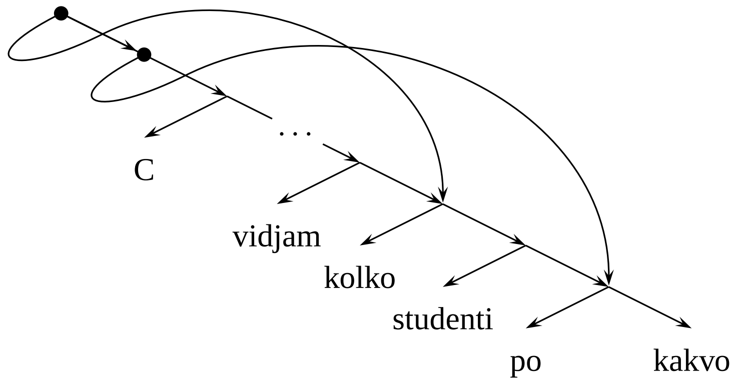


Still following Richards, but adapting his idea to a multi-dominance theory of movement, suppose that the lower *wh*-phrase (*po kakvo*) then raises, either creating a higher Spec-CP, as in (7a), or “tucking into” a lower Spec-CP, as in (7b).

(7) a.

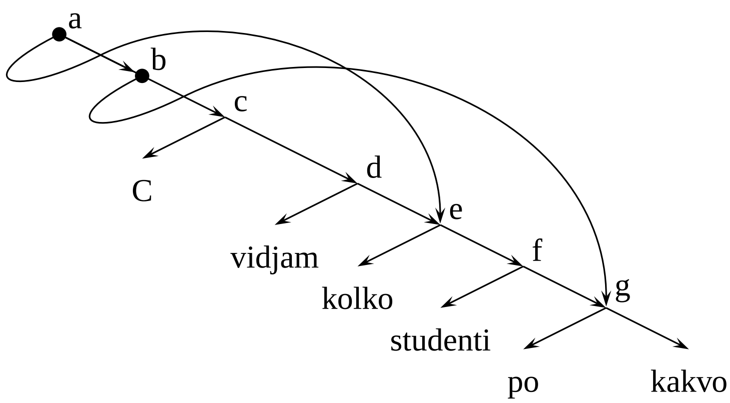


b.



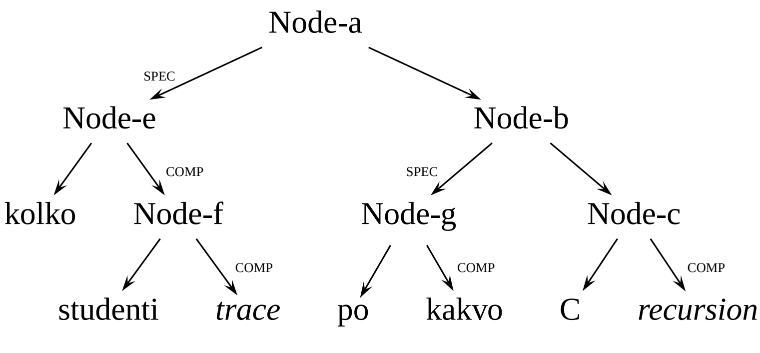
The operation of the DLA will be traced in detail for (7b), which is the kind of example that led Richards to claim that "lowering" operations are permitted. We shall see that nothing more than the DLA is involved. First, we label the nodes in (7b) so that the workings of the linearization algorithm can be traced.

(8)



Recursive linearization can be described by the recursion tree (9).

(9) Top-level recursive linearization of (9a)



This gives (5a), as desired. Note that the "trace" that appears in (9) is the consequence of the movement of *po kakvo* out of *kolko studenti po kakvo* to a position which is lower than the highest attachment point of *kolko studenti po kakvo.* But lowering is not required for this to happen.[4]

Working through the linearization of (7a) is left as an exercise for the reader. You should get:

po kakvo kolko studenti *trace* C *recursion*

Although we will later question this assumption, the fact that the order produced by the linearization algorithm mirrors the linear order in the syntactic representations should be regarded at this point as an artifact of the way we choose to draw the syntactic representations. Other choices might be useful from the standpoint of pedagogy, but are very hard to process without carefully following the recursive logic. What the linearization algorithm needs to know at each point is which branches are specifiers and which branches are complements. The precise way this is encoded is irrelevant.

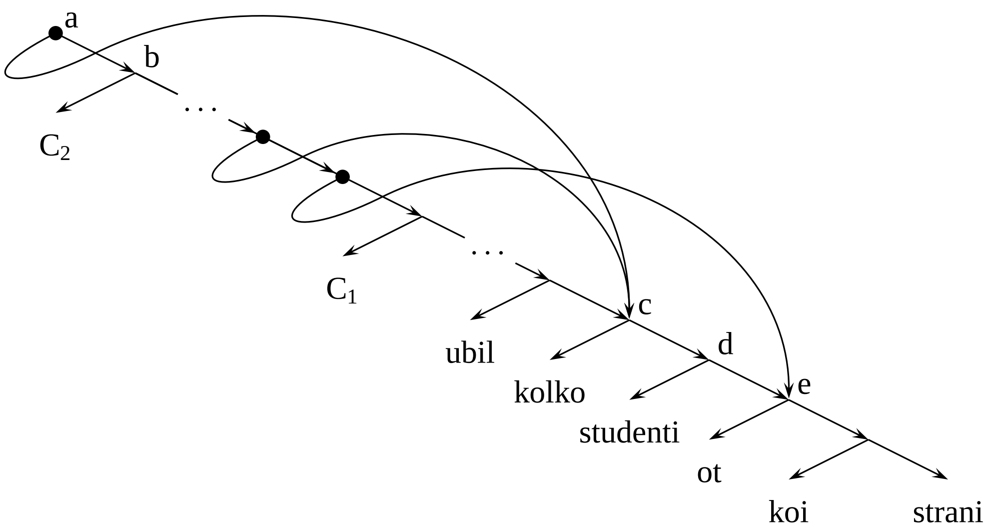
Now consider the other crucial examples given by Richards.

(10) a. Kolko studenti se opitvaš da razbereš ot koi strani e ubil Ivan
how many students you-try to find out from which countries AUX kill Ivan?
How many students are you trying to find out from which countries Ivan killed?

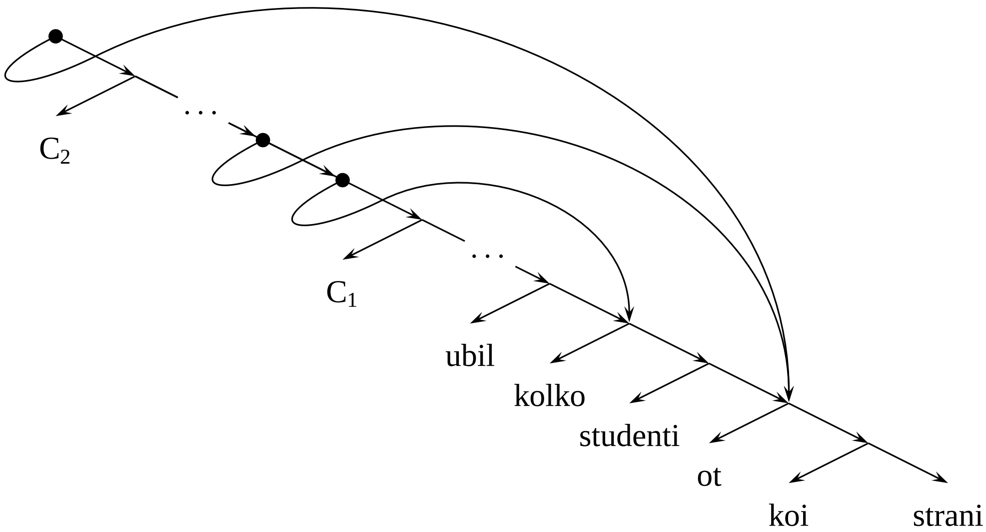
b. Ot koi stani se optivaš da razbereš kolko studenti e ubil Ivan
from which countries you-try to find out how many students AUX kill Ivan?
From which countries are you trying to find out how many students Ivan killed?

The relevant structures, assuming the syntactic possibilities in (7) for the embedded question, are given in (11). In each case, the higher embedded *wh*-phrase raises in the second syntactic C-cycle. Some nodes in (11a) are labeled to help in the explication of the linearization, which will follow shortly.

(11) a. *Kolko studenti* . . . *ot koi strani* . . .

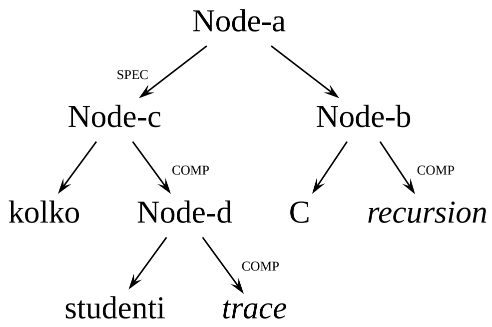


b. *Ot koi strani* . . . *kolko studenti* . . .



In each case there are three cycles of linearization: C_1 complement, C_2 complement, and C_2P. We sketch the linearization of (11a) and leave the linearization of (11b) as another exercise for the reader. In the C_1-complement cycle of the linearization of (11a), Node-c and Node-e will be traces. In the C_2-complement cycle, Node-c will be a trace, but Node-d will be linearized to the left of the linearization of C_1P. Finally:

(12) Top-level Linearization of (11a)

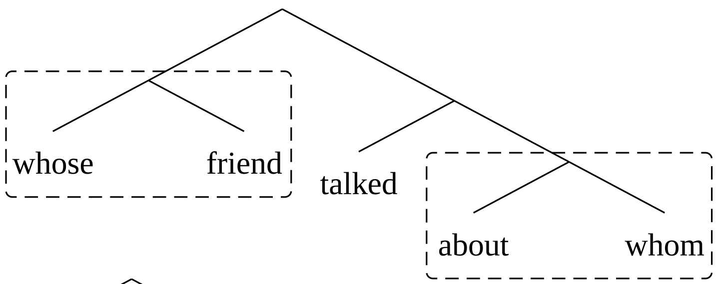


Crucially, Node-e is a Node-d trace. This is true even though the linearization of Node-e is encountered in linearizing the top-level phase and the highest attachment of Node-e is lower in the tree. Again, linearization gives the appearance of lowering.

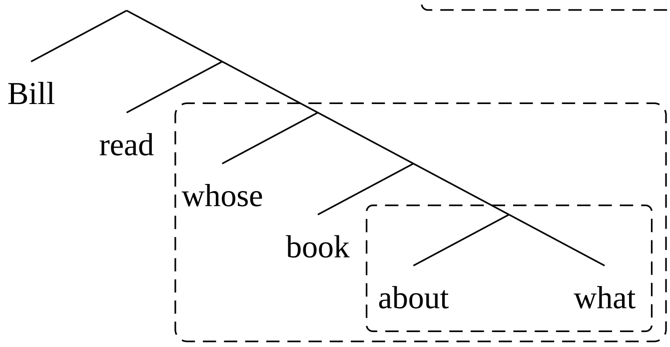
2. Linear order and Superiority

Consider the structures in (13). Imagine Bulgarian, and ignore the absence of functional projections.

(13) a.



b.



The results of *wh*-movement are:

(14) a. [whose friend] [about what] talked?
*[about what] [whose friend] talked?

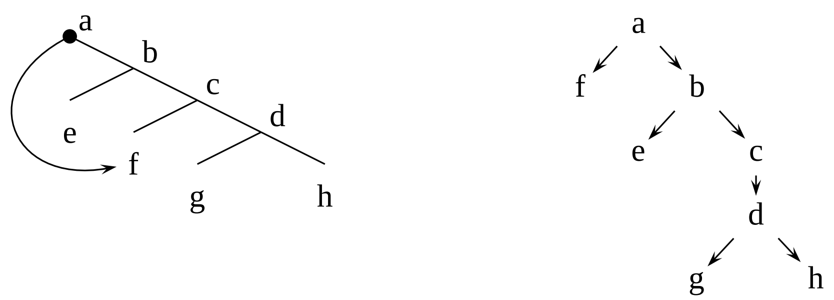
b. [whose book] [about what] Bill read?
[about what] [whose book] Bill read?

Richards develops a very finely calibrated shortest path analysis of the contrast between the two-way variation in (14b) versus the fixed extraction order in (14a). We pursue a different intuition: that the linear order implicit in the representations in (13) in some way represents a syntactic reality, not simply a typographical convenience, and that *wh*-movement is constrained to produce the minimal disruption of the linear order. The *wh*-phrases in (13a) are linearly ordered, but the *wh*-phrases in (13b) are not. In order to make good on this intuition, it will be necessary to jettison the assumption that there is no order in the syntax other than dominance, otherwise there would be no order to disrupt.

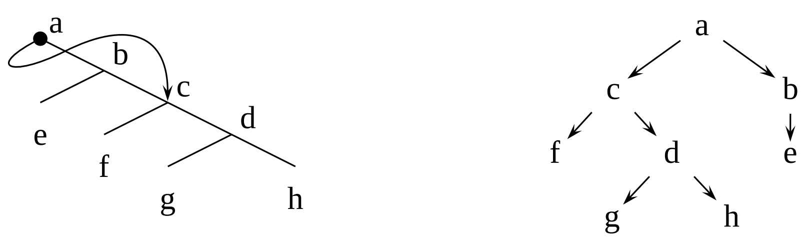
Say that α *completely dominates* β ($\alpha \gg \beta$) if β does not have a parent outside of α and β is a daughter of α, or if $\alpha \gg \beta'$ and $\beta' \gg \beta$. Say that α *precedes* β ($\alpha \prec \beta$) if α and β are daughters of γ which completely dominates them and either α is a specifier or β is a complement; or if $\alpha \prec \beta'$ and $\beta' \gg \beta$.

In order to see how this works in multi-dominance trees, two examples are given below. On the left is the syntactic structure. On the right, the corresponding complete dominance relations are represented by arrows.

(15)



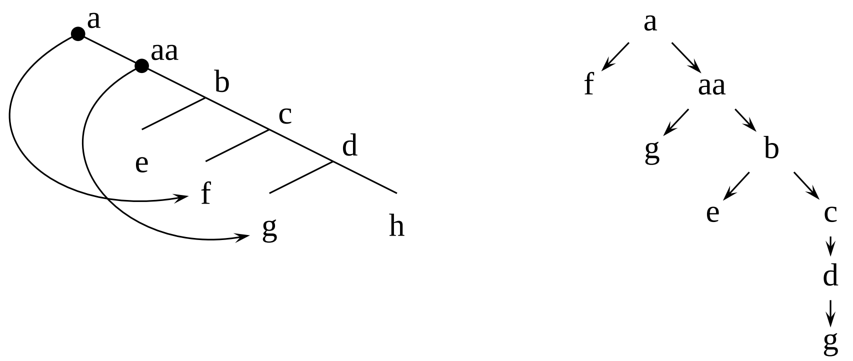
(16)



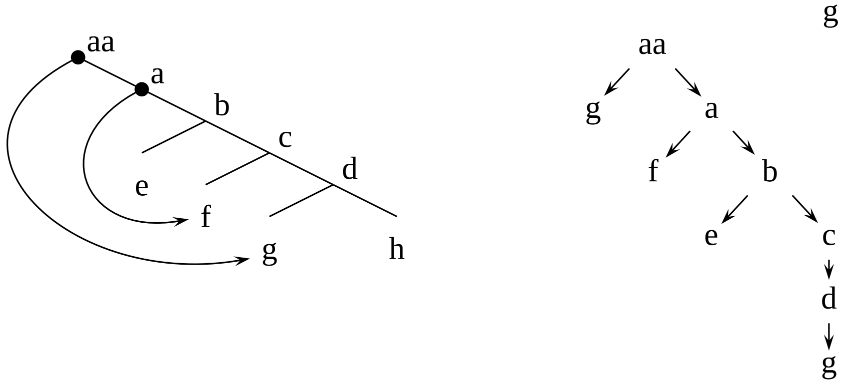
Since the dominance trees are displayed with specifiers on the left and complements on the right, the precedence relation can easily be read off the dominance trees. $x \prec y$ if y is to the right of x and dominated by the parent of x. So in (15), f $\prec$ b, e, c, d, g, h ; e $\prec$ c, d, g, h ; and g $\prec$ h. In (16), f $\prec$ d, g, h ; g $\prec$ h ; and c $\prec$ b, e . These listings are exhaustive.

We now consider how movement affects complete dominance and precedence relations. Consider first two potential transformations of (15), with (17a) corresponding to what Richards has called "tucking in".

(17) a.



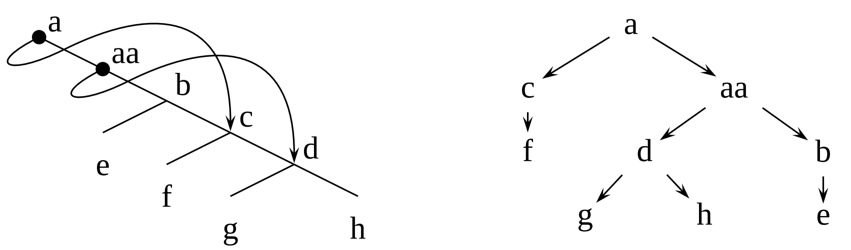
b.



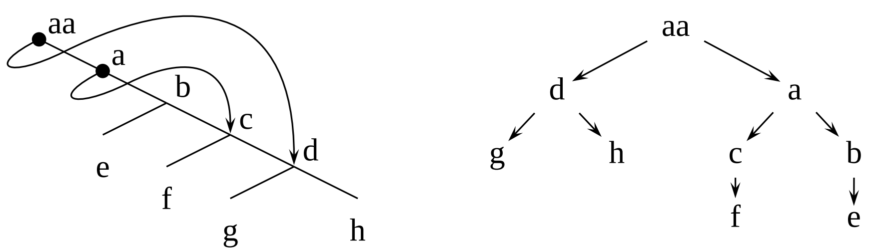
In (15) and (17a), $f \prec g$. In (17b), $g \prec f$. If both (17a) and (17b) are otherwise allowed, but movement favors minimal disruption of precedence, then (17a) is ruled out and (17b) forced.

Now consider the relevant possibilities for (16).

(18) a.



b.



In (18a), c $\prec$ d and in (18b), d $\prec$ c. Note, however, that in the input structure (16), there is no precedence relation between c and d. A condition favoring minimal disruption of precedence does not choose between (18a) and (18b).

We now turn to the specifics of *wh*-movement. Assume, along with Richards, that the multiple *wh*-extraction in Bulgarian is sequential, and that standard "attract closest" assumptions imply, for example, that *whose friend* moves first in (13a) and that *whose book about what* moves first in (13b). The issue is the landing site of the second *wh*-phrase. We ignore, as does Richards, the problem of specifying precisely how the *wh*-phrases are targeted for movement rather than simply the *wh*-word within the phrases (pied-piping). We also ignore, as does Richards, the problems in developing an agreement based theory of movement which allows looking inside a *wh*-phrase for a second embedded *wh*-feature without encountering minimal link obstacles. All that is offered here is a way to understand Superiority in terms of precedence that avoids the delicacy of Richards' shortest path analysis.

The crucial assumptions we add are that there is an option of moving to either a higher or lower Spec-CP position in multiple *wh*-phrase movement (after the initial *wh*-phrase has moved), but that minimal disruption of precedence relations is favored.

Notes

1. I have discussed these matters with Sam Gutmann for almost ten years. He shares my fascination with data structures and sense of the crucial role they play in computation. Arguably, he should be a coauthor. Thanks to him and to Hans-Martin Gärtner, Jim McCloskey, and Norvin Richards for comments on an earlier draft, which were much appreciated.

2. See Frampton (2004) for some discussion.

3. In an attempt to help overcome this problem, the Tex source code for the examples in this paper is available at *http://www.math.neu.edu/ling/tex.* In the authors' opinion, typographical limitations have had a very negative effect in linguistic research, biasing researchers towards the implicit assumption that mental computation tends to make use of data structures which are easy to represent on two dimensional surfaces. The widespread inattention to autosegmental considerations in phonology is a good example.

4. Richards, in effect, recognizes the possibility of something along these lines in fn. 10 in considering the attraction of "the 'original copy' of the embedded *wh*-phrase out of the trace of *wh*-movement of the embedding *wh*-phrase." He calls it a "disturbing alternative derivation." Abandoning the copy theory of traces and admitting multiple-dominance tree representations allows the basic idea to be recast in a much more attractive form.

References

Chomsky, Noam. 1995. Bare phrase structure. In *Government and binding theory and the minimalist program*, ed. G. Webelhuth, 383-439. Oxford: Blackwell.
Frampton, John. 2004. Some Notes on Synthetic/Periphrastic Alternation. At *http://www.math.neu.edu/ling/*.
Gärtner, Hans-Martin. 2002. *Generalized Transformations and Beyond: Reflections on Minimalist Syntax*. Berlin: Akademie-Verlag.
Kracht, Marcus. 2001. Syntax in Chains. *Linguistics and Philosophy* 24: 467–529.
Richards, Norvin. 2004. Against Bans on Lowering. *Linguistic Inquiry* 35.3: 453–463.