Mark de Vries
On Multidominance and Linearization
(*Biolinguistics 3.4: 344-403*)

1. OVERVIEW OF MAIN AIMS AND CLAIMS

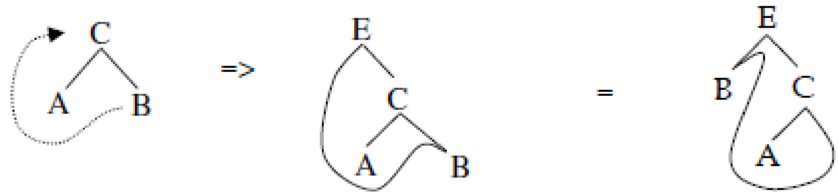
(i) Both movement and structure sharing follow equally and immediately from Merge: they simply correspond to the two available options for Remerge – internal and external. If we have Remerge at all (i.e. merge that takes as input a term of a derived structure), yielding displacement (internal remerge), then we can only exclude external remerge (structure sharing) by stipulation.

(1) For some constituent α embedded in root R:
a. *internal remerge* $=_{def}$ remerge α with R;
b. *external remerge* $=_{def}$ remerge α outside R (i.e., with some root β not included in R).

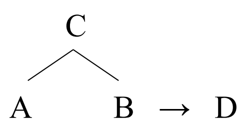
(2) Merge (α, β) → γ constitutes
a. *first-time merge* iff α and β are independent roots before merger;
b. *internal remerge* iff β is a root and α is included in β (or the other way around) before merger;
c. *external remerge* iff β is included in some root δ, and α is an independent root (or the other way around) before merger.
(where 'roots' includes heads introduced from the lexicon/numeration, i.e. 'trivial' roots)

(ii) Both internal and external remerge yield multidominance structures.

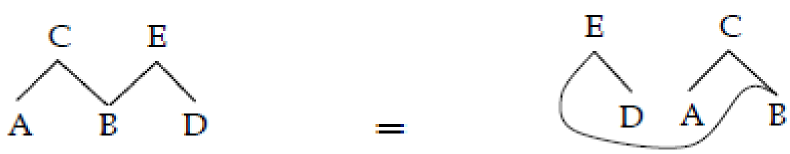
(3) *Internal Remerge*:
Merge (A, B) → C
Merge (B, C) → E



(4) *External Remerge*:
Merge (A, B) → C
Merge (B, D) → E



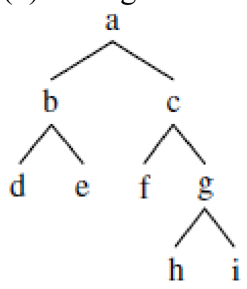
=>



(iii) A simple linearization algorithm (involving recursive top-down tree traversal) can handle any structure that Merge can throw at it (first-merge, internal remerge, external remerge, and all combinations thereof: successive-cyclic movement, roll-up movement, remnant movement, etc.).

(5) *Scan triad* (i.e. mother/parent and two daughters/children, starting at root node)
↓ (← list present node)
↓ select left child; if complex => start *scan triad* of child
↓ (← list present node)
↓ select right child; if complex => start *scan triad* of child
↓ (← list present node)

(6) e.g.:



(i) **Start at root**, *a*; *a* has children (i.e. is complex)
(ii) Select left child, *b*; *b* has children
(iii) Select left child, *d*; *d* has no children (i.e. is not complex) → **list *d***
(iv) Return to *b* and select right child, *e*; *e* has no children → **list *e***
(v) Return to *a* (via *b*); select right child, *c*; *c* has children
(vi) Select left child, *f*; *f* has no children → **list *f***
(vii) Return to *c* and select right child, *g*; *g* has children
(viii) Select left child, *h*; *h* has no children → **list *h***
(ix) Return to *g* and select right child, *i*; *i* has no children → **list *i***
(x) Return to a (via g, c); → **end procedure**

- Obtained string/list of terminals: **/d e f h i/**

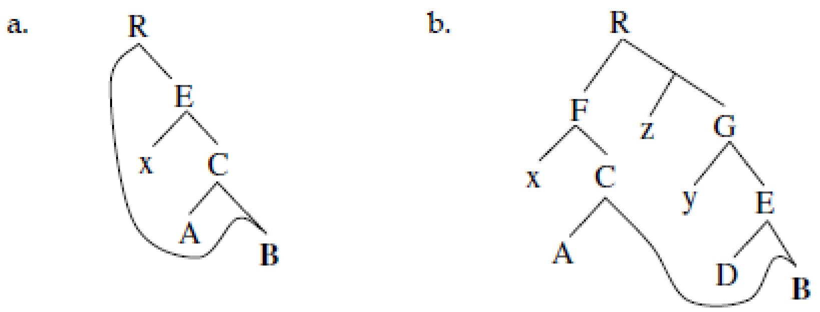
(iv) The proposed linearization procedure, unlike some previous attempts to linearize multidominance structures in the literature, can handle both internal and external remerge with equal aplomb.

- Compare, e.g., Frampton 2004, which only handles internal remerge; Citko 2005, whose system can only handle cases where external remerge (/parallel merge) is followed by internal remerge; and Wilder 1999, which only really deals with external remerge, and even then can only handle cases of external remerge in which the right-periphery condition holds (e.g. RNR). Wilder's system thus cannot be extended to other potential cases of external remerge, such as *cleft-amalgams ('saddle grafts'*, in van Riemsdijk's terminology), where the shared material isn't necessarily right-peripheral:

(7) *Dutch*
Joop heft [ik vermoed dat het *een Bugatti* is] gekocht.
Joop has I presume that it a Bugatti is bought
'Joop bought – I presume it's a Bugatti.'

- More particularly, the proposed linearization procedure derives without further ado the contradictory (non-uniform) spell-out profiles of internal versus external remerge: Internal remerge (movement; (8a)) is 'to the left' (i.e. the highest, leftmost occurrence is spelled out) whereas external remerge (sharing; (8b)) is 'to the right'(the rightmost occurrence is spelled out):

(8) a. *Which violin* should this talented girl purchase ___?
b. The boy only admired ___, but the girl actually bought *this beautiful Stradivarius*.



Terminals encountered by graph traversal in a. (internal remerge): /**B** x A **B**/
Desired string of terminals: /**B** x A/

Terminals encountered by graph traversal in b. (external remerge): /x A **B** z y D **B**/
Desired string of terminals: /x A z y D **B**/

- *Generalizations to be derived by linearization procedure* (p.365):
  Internal remerge of X → overtly realize only the **first** occurrence of X while linearizing the structure.
  External remerge of X → overtly realize only the **last** occurrence of X while linearizing the structure.

De Vries proposes that all we need to add to the Scan Triad procedure in (5) in order to handle remerge (and capture these generalizations) are the following conditions:

(9) *Spell-Out of Remerged Nodes* (p.381)
An α with more than one parent is linearized if and only if
a. the current parent is not dominated by any other parent, *and*
b. (i) every parent has been traversed, *or*
(ii) the current parent dominates every other parent that has not been traversed

… where 'current parent' is defined as follows:

(10) *Current Parent* (p.376)
The current parent of α is the most recently traversed parent during the linearization procedure.

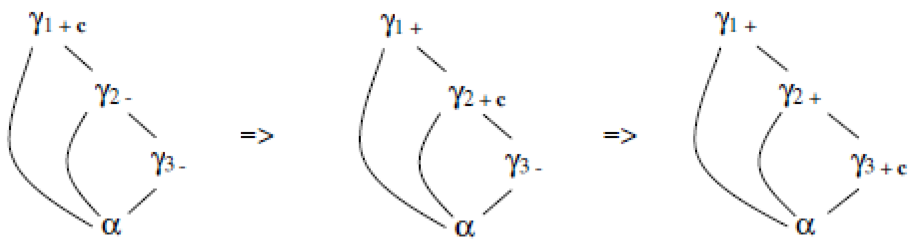
2. SOME COOL TREES (AND THEIR LINEARIZATION)

The effect of (9), essentially, is that what gets spelled out is the highest rightmost occurrence of α whose parent is undominated by any other parent of α – the correct result.

2.1 Internal remerge (successive-cyclic)

For internal merge, we only want the highest occurrence (top copy) to be realized:

(11)



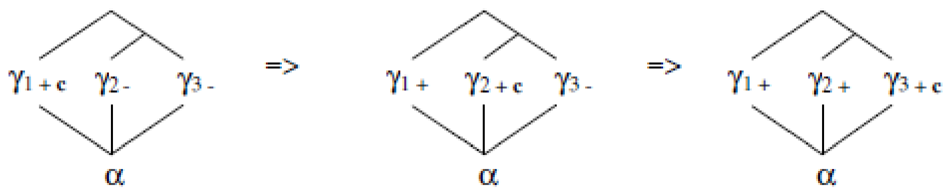
spell out α omit α omit α

- By (9a) and (9b.ii), the highest occurrence (current parent = γ_1) is spelled out: the current parent is not dominated by any other parent and dominates all parents that have yet to be traversed.
- The two lower occurrences are omitted since (9a) doesn't hold of them.

2.2 External remerge

For external remerge, we only want the rightmost occurrence to be spelled out (no matter how many times external remerge applies; cf. *John hates ___, Bill likes ___, Mary admires ___, and Jack detests the president*):

(12)



omit α omit α spell out α

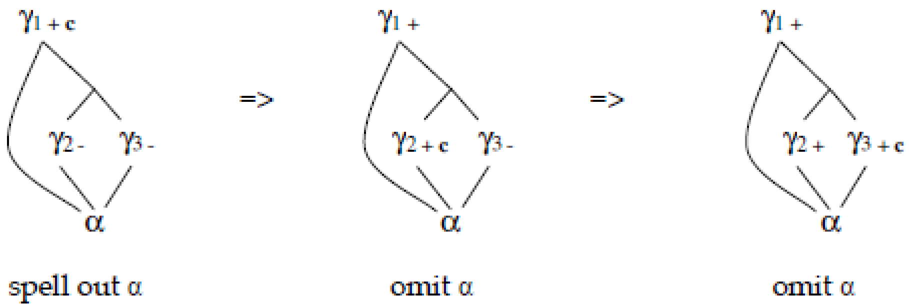
- The first two occurrences encountered (i.e. when current parent = γ_1 and when current parent = γ_2) are omitted: although (9a) holds (they are not dominated by any other parent), neither (9b.i) nor (9b.ii) holds (not every parent has been traversed, nor do either of these parents ($\gamma_1$or γ_2) dominate those that have yet to be traversed).
- The third occurrence (current parent = γ_3) is spelled out, since (9a) and (9b.i) hold.

2.3 External remerge + internal remerge (e.g. ATB)

Here, we want linearization to spell out the highest occurrence, as for internal merge:

- By (9a) and (9b.ii), the highest occurrence (current parent = γ_1) is spelled out: the current parent is not dominated by any other parent and dominates all parents that have yet to be traversed.
- The two lower occurrences are omitted since (9a) doesn't hold of them.

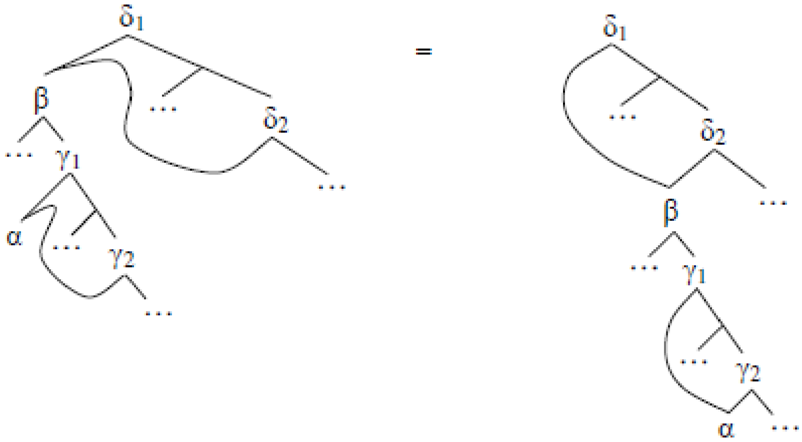
(13)



2.4 Internal remerge inside internal remerge

This is movement inside a moved constituent. Here, we only want the highest occurrence of α inside the highest occurrence of β to be spelled out:

(14)



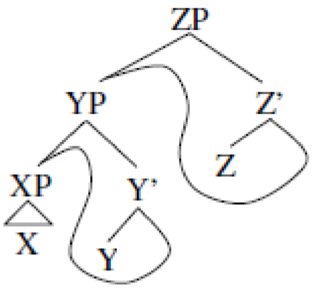
- By (9a) and (9b.ii), the highest occurrence of β (current parent = δ_1) is spelled out: the current parent is not dominated by any other parent and dominates all parents that have yet to be traversed.
- Inside β, the highest occurrence of α (current parent = γ_1) is spelled out, by (9a) and (9b.ii).
- The lower occurrence of β is omitted since (9a) does not hold of it. Therefore, the contents of that lower occurrence are also omitted.

2.5 Roll-up movement

Here, phrases raise successively into the specifier of the next higher phrase, with the result that the spell-out order of terminals should mirror the base order (cf. Brody 2000), i.e. we want /X Y Z/ and not /Z Y X/ in (15).

- By (9a) and (9b.ii), the highest occurrence of YP feeds linearization (its parent, ZP, is not dominated by the other parent, Z', and dominates all parents that have yet to be traversed, i.e. Z').
- Inside YP, the same holds for XP: the highest occurrence (parent = YP) is linearized by (9a) and (9b.ii).

(15)



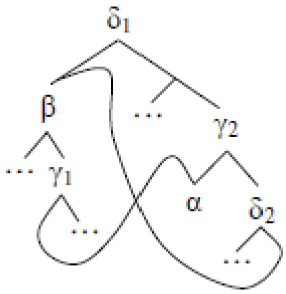
2.6 Remnant movement

Remnant movement can pose a problem for LCA-based accounts, as the highest copy of α (contained in the remnant) is not c-commanded by any other copy and so should arguably feed linearization (i.e. be spelled out). Instead, we want the *second* occurrence of α to be spelled out.

- "[W]hat makes remnant movement special is that an instance of internal remerge gets the structural appearance of external remerge." (p.384)

De Vries's system derives this without further ado: i.e., *remnant movement is identical to external merge of the extracted material (here,* α*) for the purposes of linearization*:

(16)

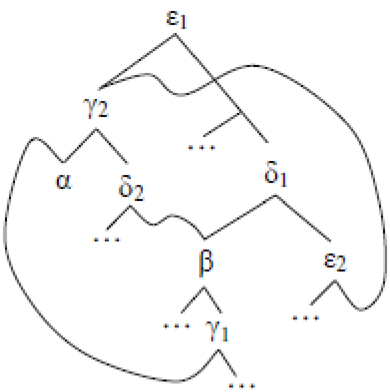


- By (9a) and (9b.ii), the first (highest) occurrence of β is linearized.
- Inside β, however, linearization of α is omitted, as for external remerge: although (9a) holds (parent γ_1 is not dominated by the other parent, γ_2), neither (9b.i) nor (9b.ii) holds (not every parent has been traversed yet, nor does the current parent (γ_1) dominate the parent that has yet to be traversed (γ_2)).
- The second occurrence of α (current parent = γ_2) is spelled out, since (9a) and (9b.i) hold (the current parent, γ_2, is not dominated by any other parent, and every parent has now been traversed).

2.7 Iterated remnant movement

De Vries's "worst case scenario": Here, (16) becomes the input to further remnant movement, e.g. of γ_2 across δ_1. The spell-out positions of α and β are thus reversed: we now want the *first* occurrence of α and the *second* occurrence of β to be spelled out (i.e. it is now β and not α that acts as if it were externally remerged, linearization-wise):

(17)



No problem:

- By (9a) and (9b.ii), the highest (first) occurrence of γ_2 is linearized, as is the occurrence of α inside this (since γ_2 now dominates α's other mother, γ_1).
- By contrast, β inside γ_2 is omitted, as per external remerge: although (9a) holds (parent δ_2 is not dominated by the other parent, δ_1), neither (9b.i) nor (9b.ii) holds (not every parent has been traversed yet, nor does the current parent (δ_2) dominate the parent that has yet to be traversed (δ_1)).
- The second occurrence of β (current parent = δ_1) is spelled out, as per external remerge, since (9a) and (9b.i) hold (the current parent, δ_1, is not dominated by any other parent of β, and every parent has now been traversed).

In sum:

- If Merge can generate it, then (5)/(9) can linearize it (as long as the Single Root Condition is met – Scan Triad (5) relies on there being a single identifiable root node at which to start the traversal procedure). Any combination of Merge operations (including syntactically illicit ones, e.g. Freezing violations, "roll-out movement") can be linearized.

3. FURTHER ISSUES AND PROBLEMS

3.1 Cyclicity, NTC, conditions on Merge, ordering, and islands

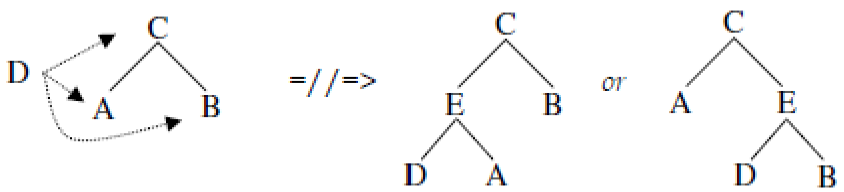
- Trees/graphs, like those depicted above, have no independent theoretical status (p.350). All we have is Merge, which yields sets of relations (dominance, sisterhood). Thus "multidominance graphs ... are only representations of the underlying relations between syntactic objects created by Merge" (p.364).
- Therefore, the only conditions on possible trees/graphs are conditions on Merge – independent graph-theoretical / tree-theoretical well-formedness conditions, such as the Single Mother Condition, Exclusivity, and Non-tangling, have no independent theoretical status and so must be abandoned as stipulations (they do not follow from Merge itself – hence this paper is very much in line with the minimalist, biolinguistic spirit of the SMT); cf. p.372.
- For example, de Vries explicitly rejects (18) (essentially, Nontangling):

(18) *Inheritance of Precedence* (rejected, p.371)
If x directly precedes y, then x and all nodes dominated by x precede y and all nodes dominated by y.

- The trees produced by Remerge, then, such as those in (11)-(17), may be "graphically awkward" (p.347), but they simply represent relations that Merge entails.
  - ➢ External Remerge simply combines two independently needed inputs to Merge: external material from the numeration/lexicon, as needed for first-Merge, and internal material (terms of an existing syntactic object), as needed for movement (internal remerge). Only stipulations could rule out their combination; and various empirical phenomena suggest that we should not do so (RNR, ATB, amalgams, etc., all arguably instantiate external remerge).

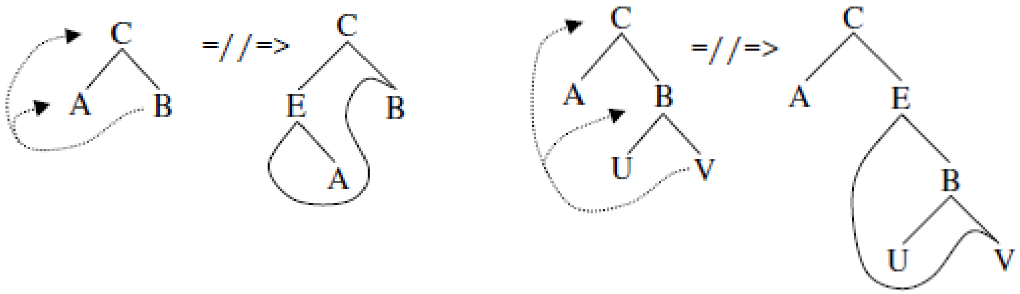
Internal and external remerge should therefore both conform to the usual conditions on Merge – they should respect the Extension Condition and the Non-Tampering Condition (NTC): i.e., remerge that targets a nonroot (thus replacing/destroying that node) should be disallowed:

(19) *NTC holds of external (re)merge, (2a,c)*



➢ NTC/Extension Condition requires that Merge (D, A) in (19) yields a new, additional root, as in (4).

(20) *NTC holds of internal Remerge, (2b)*

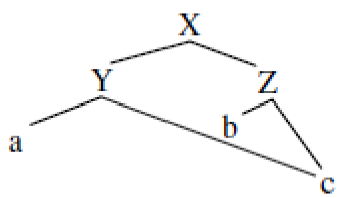


➢ Merge is structure-building, not structure-changing.

<u>Question</u>: Does NTC hold of external remerge (like it should)?

External remerge can yield structures like (21), which *look like* violations of the NTC/Extension Condition:

(21)



- If (21) were formed by *internal* remerge, it would indeed be an NTC/Extension Condition violation like in (20): c merges with a in X's specifier to form the new node Y containing a and c.
- However, external remerge allows a cyclic derivation of (21), i.e. one that respects the NTC/Extension Condition:

(22) a. Merge (a, c) → Y
b. Merge (b, c) → Z
c. Merge (Y, Z) → X

- Thus a multidominance structure might *look* bad (violation of cyclicity/extension condition), but whether it is actually ruled out or not depends on *how* it is derived (through internal remerge or through external remerge).
- Prediction: If structures like (21) exist, they must be the result of external remerge, not internal remerge (p.385). This predicts that it will be the second (rightmost) occurrence (of c in (21)) that will be spelled out.

Dutch examples (23) and (24) illustrate the kind of 'subordinative' RNR/structure-sharing that might instantiate (21).

(23) Het kan moeilijk zijn om syntactische ___ van semantische factoren te onderscheiden.
it can difficult be to syntactic from semantic factors to distinguish
'It can be hard to distinguish syntactic ___ from semantic factors.'

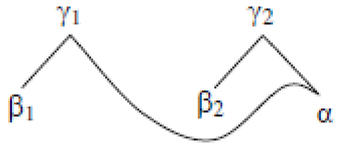
(24) Joop is, ofschoon een schuldbewust gebruiker van ___, niettemin principieel
Joop is although a contrite user of nevertheless principally
gekant tegen de intracontinentale luchtvaart.
opposed against the intracontinental aviation
'Joop is, although a contrite user of ___, nevertheless principally opposed to intracontinental aviation.'

- **No inherent derivational ordering with external remerge**: (22a) and (22b) apply in either order (i.e. it doesn't matter which of these steps is first-Merge and which is Remerge); linearization by (5)/(9) will always spell-out the rightmost occurrence, regardless.
  - Important consequence: **island constraints** will *only hold of* ***internal*** *remerge*, not of external remerge (since the 'sideward link' can be established at any relative point in the two parallel derivations – de Vries calls this a "structural bypass"):

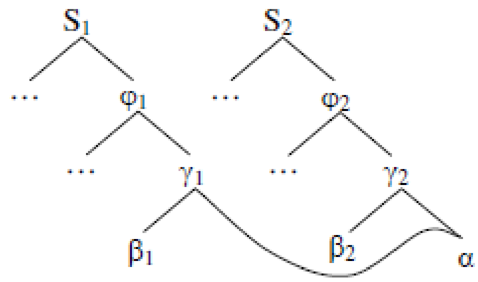
(25) Structural bypass (external remerge in b. [RNR] and c. [cleft-amalgam]):
a. **What* does Mary like [men who sell ___]?
b. Mary likes [men who SELL ___], but she hates [men who BUY *cars*].
c. Joop kuste toen – Piet beweerde dat hij iemand kende die zei dat het
Joop kissed then Piet claimed that he someone knew who said that it
Mieke was.
Mieke was
'Then, Joop kissed – Piet claimed that he knew someone who said that it was Mieke.'

(26) *'Structural bypass' with external remerge*

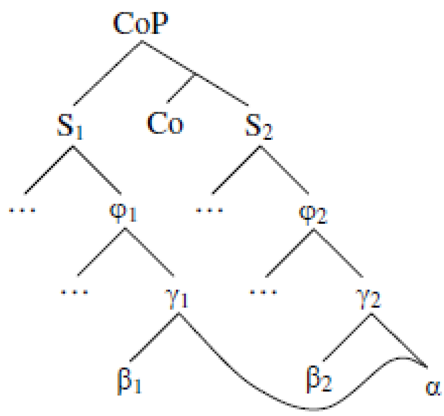
a.



b.



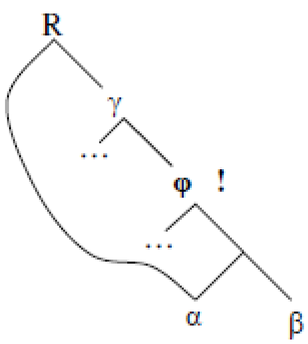
c.



- External remerge can take place before the locality boundaries (island nodes, φ_1, φ_2) are merged into place.
- With *internal* remerge, however, we have a single unified root/tree at the point of the offending (island-violating) remerge operation, not separate subtrees. The offending remerge operation is thus inherently derivationally ordered to take place *after* the locality boundary / island node φ is in place:

(27) *No structural bypass with internal remerge*

*



[Still a problem, though – parasitic gaps, if derived through external remerge (cf. Nunes), as these are still subject to island effects.]

<u>Back to NTC</u>: External remerge is fine by the NTC/Extension Condition if it yields structures as in (4) rather than those in (19).

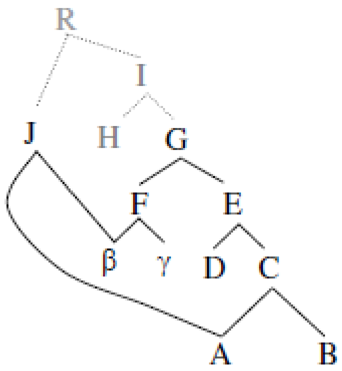
“Quirky” remerge: Structures that go against the ‘spirit’ of the NTC/Extension Condition.

- The structures in (28) and (29) create a new root (i.e. they are structure-building, not structure-changing, and so conform to NTC), but they involve nonroots (and Merge should only target a root node, cf. (1), (2)).

(28) ‘*Quirky internal remerge*’
(Merge (β, γ) → F)

Merge (A, B) → C
Merge (D, C) → E
Merge (F, E) → G
Merge (A, β) → J

Merge (H, G) → I
Merge (J, I) → R



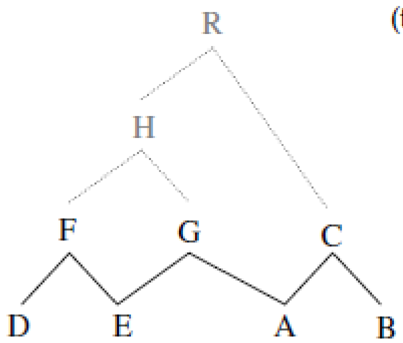
(to be excluded)

➢ Quirky step to be excluded: **Merge (A, β) → J**

(29) ‘*Quirky external remerge*’
(Merge (D, E) → F)

Merge (A, B) → C
Merge (E, A) → G

Merge (F, G) → H
Merge (H, C) → R



(to be excluded)

➢ Quirky step to be excluded: **Merge (E, A) → G**

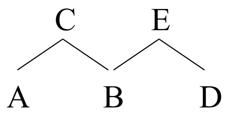
De Vries proposes to exclude these unwanted cases of remerge by the Root Condition in (30), which can itself be derived as in (31).

(30) *Root condition*:
If α and β are selected as input for Merge, then α or β (or both) must be a root.

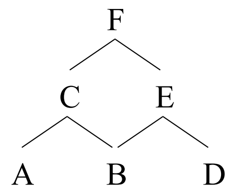
(31) *No proliferation of roots condition*:
If the derivation proceeds from stage i to i+1 through Merge (α, β) → γ,
then $|\{x \in \{\alpha, \beta, \gamma\}: x \text{ is a root at stage } i{+}1\}| \leq |\{x \in \{\alpha, \beta\}: x \text{ is a root at stage } i\}|$.

- Merge either reduces the number of roots in the workspace by one (external merge, first-Merge) or else maintains the same number of roots (remerge, external or internal).
- The 'quirky' cases of remerge in (28), (29), on the other hand, *increase* the number of roots by one. This is what (31) now excludes.
- Functionally, the 'point' of merge is to take a set of lexical items (numeration) and combine them (in binary fashion) into a single-rooted structure (i.e. one that is interpretable/linearizable at PF) – i.e., Merge tends towards a single-rooted structure. Hence the *Single Root Condition* may still plausibly hold of Merge and the structures it generates: A double-rooted structure (such as that initially created by external remerge, as in (32a)) is still 'unfinished': it has the potential for at least one more Merge operation (even if it's just Merge(C, E) here, yielding (32b)).

(32) a.



b.



<u>However</u>: ***Problem*** *(not noted by de Vries)*

- Due to the lack of derivational ordering with external remerge (cf. above), it is impossible/incoherent to impose the derivational sequences specified in (28) and (29).
    - Therefore, it is perfectly possible to derive the 'unwanted' structures in (28) and (29) in a manner that conforms with (30)/(31) by simply starting with the 'illicit' remerge step (i.e. Merge (A, β) → J in (28); Merge (E, A) → G in (29)).

(33) *Licit derivation of (28)*:

Merge (A, β) → J
Merge (A, B) → C [= remerge of A to root B, thus conforming to (30)/(31)]
Merge (β, γ) → F [= remerge of β to root γ, thus conforming to (30)/(31)]
Merge (D, C) → E
Merge (F, E) → G
Merge (H, G) → I
Merge (J, I) → R

- At no step is the number of roots increased; every Merge step involves at least one root.

(34) *Licit derivation of (29)*:

Merge (E, A) → G
Merge (D, E) → F [= remerge of E to root D, thus conforming to (30)/(31)]
Merge (A, B) → C [= remerge of A to root B, thus conforming to (30)/(31)]
Merge (F, G) → H
Merge (H, C) → R

- At no step is the number of roots increased; every Merge step involves at least one root.

- Hoisted by his own "ordering doesn't matter" petard? Or might we simply want to allow structures like (28) and (29) anyway (after all, they already conform to NTC/Extension Condition)?

3.2 *Extra*: What Chomsky says

Chomsky explicitly rejects shared structures of the kind that result from external remerge:

(35) a. "Merge cannot create objects in which some object O is shared by the merged elements X, Y." (Chomsky 2007: 8)
b. "[Such Merge] requires new operations and conditions on what counts as a copy." (Chomsky 2007: 8, fn. 10)

- Criticism (35a) is aimed at Citko/Parallel Merge, and holds insofar as structures like (4) are created in a single application of Merge (i.e. if Parallel Merge = merger of two roots via a term of one of them), but it does not hold of de Vries's external remerge or indeed any approach to structure sharing that derives such structures in a licit two-step derivation (as given in (4); this is surely also what Citko intends). Thus the (il)licitness of an operation should be not be conflated with the (il)licitness of the structure.
- Criticism (35b) is arguably true of Nunes/sideward movement, which relies on a distinct copying mechanism applying in addition to Merge. (It would also be true of Citko's parallel merge *if* this were indeed a "new operation", deriving (4) as a supposed instance of ParallelMerge(C, E) (which is *not* what Citko seems to intend).) But de Vries's external remerge neither appeals to a copying mechanism (it uses remerge/multidominance, thus attributes no properties to 'copies' as such), nor is it a new operation (no new Merge operations are required; it is simply external merge applied a second time to a given syntactic object, much as it is for Citko). Only at PF (i.e. for linearization) might (35b) be true of de Vries's approach (i.e. (9) sets new conditions on which occurrences count for spell-out).

→ Chomsky's objections, as they stand, perhaps do not rule out external remerge (and the structures it generates) after all, at least in the syntax.

ADDITIONAL REFERENCES

Brody, M. (2000) Mirror Theory: Syntactic Representation in Perfect Syntax. *Linguistic Inquiry* 31: 29-56.
Chomsky, N. (2007) Approaching UG from below. In Uli Sauerland & Hans-Martin Gärtner (eds.) *Interfaces + Recursion = Language? Chomsky's Minimalism and the View from Syntax–Semantics*, 1–29. Berlin: Mouton de Gruyter.

.