Merge

For Cambridge Handbook of Minimalism
version 1, December 2, 2020, finished March 24, 2021, version 2, November 23, 2021.

Jan-Wouter Zwart

1. Introduction

Merge (Chomsky 1995:226) is the phrase structure rule of minimalism. It describes the formation and composition of syntactic constituents. Incorporating also movement (see [next vignette]), it is the only structure building operation of Narrow Syntax. Applied recursively, Merge generates hierarchical phrase structure, and configurational relations such as c-command can be defined in terms of Merge (Epstein 1999).

Native speakers identify strings of elements as constituents by assigning them an interpretation and by singling them out as targets for syntactic operations (such as movement, coordination, and ellipsis). Any model of the human language faculty must therefore include a formal definition of constituents. Merge is arguably the simplest way to account for these native speaker intuitions of constituency.

In addition, humans can produce and interpret novel and potentially infinitely long expressions, structured along the lines of constituency (the property of 'discrete infinity'). This, too, is achieved by the recursive application of Merge (see Hauser, Chomsky and Fitch 2002 on the central role of recursion in the human faculty of language).

2. Definition

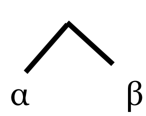
On most definitions, Merge takes two elements and combines them in a set (Chomsky 1995:243; on the need to label the set resulting from Merge, see below, section 3c):

(1) Merge

$$\alpha, \beta \rightarrow \{ \alpha, \beta \}$$

(1) expresses that the syntactic object $\{ \alpha, \beta \}$ is formed by joining the syntactic objects α and β, and that $\{ \alpha, \beta \}$ is composed of α and β. $\{ \alpha, \beta \}$ can therefore be represented as the tree structure in (2), or as the bracketed string in (3).

(2)



(3) $[\ \alpha\ \ \beta\ ]$

Prior to the operation Merge, α and β are independent elements, either lexical items or previously constructed syntactic objects. Movement (also referred to as Move or Internal Merge) can be defined as a subcase of (1), where α is contained within a previously constructed syntactic object β (Kitahara 1994, Chomsky 2000b:13).

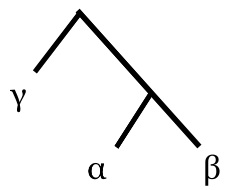
Hierarchical structure results from a second operation of (1), now taking the syntactic object { α, β } as one of the lefthand terms:

(4) Merge

γ, { α, β } → { γ, { α, β } }

The resulting syntactic object { γ, { α, β } } can be represented as in (5) or (6):

(5)



(6) [γ [α β]]

The sequence of operations (1)/(4) defines *αβ* and *γαβ* as constituents.

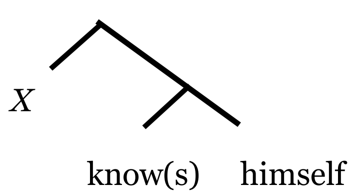
3. Definition of syntactic relations in terms of Merge.

As observed by Epstein (1999), the configurational relation of c-command, describing the domain of scope bearing elements, can be defined in terms of Merge. The relevance of c-command is illustrated by the pair of sentences in (7), where *himself* is in the scope of (c-commanded by) *John*, and can hence be interpreted as referring to *John*, in (7a), but not in (7b).

(7) a. John knows himself
b. *Friends of John know himself

(7) has the structure in (5), illustrated in (8).

(8)



X = *John* in (7a) and *friends of John* in (7b). Traditionally, an element *x* is taken to c-command all elements distinct from *x* that are dominated by the first branching node dominating *x*. While this describes the scope domain of *x* correctly, it was never clear why the first branching node was relevant to the definition. But crucially, *John* is merged with *knows himself* in (7a), but not in (7b), where *John* is contained within *friends of John*. This led Epstein to define c-command in a more principled way, in terms of Merge (cf. Epstein 1999:329), essentially:

(9) α c-commands β iff α is merged with (a node dominating) β

This entailed a shift from a representational to a derivational definition of c-command. In a representational definition of c-command, a subdivision must be made among the nodes in a tree, some of which are, while others are not, c-commanded by a given element. Such a subdivision inevitably raises the question why no other subdivisions apply. But in the derivational definition of c-command proposed by Epstein, the c-command domain of any element *x* is determined at the point in the derivation where *x* is merged (to *y*, say), and at that point *x* c-commands *y* and every element contained in *y*, i.e. all the elements in existence in the structure prior to merger of *x*.

Other, more primitive, configurational relations can also be easily defined in terms of Merge.

(10) a. sister
α and β are sisters iff α and β are merged
b. dominate
α dominates β iff α results from an operation Merge involving (γ dominating) β

(It will be recalled that *dominate*, crucial to some definitions in the Barriers-framework of Chomsky 1986, did not receive a formal definition there.)

4. Conditions on Merge

In general, Merge is taken to operate freely (Chomsky 2001:3, Chomsky, Gallego and Ott 2019:237) and the few conditions on Merge that have been formulated explicitly derive from general considerations of simplicity.

a. The Extension Condition

Merge is subject to the Extension Condition, which states that one of the lefthand elements in (1) must be a root (i.e. α and β cannot both be subparts of syntactic objects already construed; Chomsky 1995:248). This has the result that Merge always expands the structure at the root, i.e. builds structure 'from the bottom up'.

The Extension Condition entails that certain movements cannot be formally identified with Merge, most notably head-to-head movement (Chomsky 2001:37), suggesting that it may not be an operation of Narrow Syntax.

The Extension Condition is motivated by the observation that the kind of structures or phenomena that would be the effect of flouting the Extension Condition do not appear to be attested (Chomsky 1995:248; see below for Late Merge).

Jaspers (1998:109) observes an asymmetry between the two elements merging, in the sense that typically one of the elements represents a syntactic object under construction (OUC), while the other element is merged *to* that OUC. The only step where the two elements appear to be truly symmetrical is when both elements merge for the first time ('first Merge'). This suggests an asymmetry in the operation of Merge, allowing for a simpler statement of the Extension Condition, namely that the element merged *to* must be a root (this asymmetry is apparently already implied in Chomsky 1995:248).

Chomsky (2019a:276) rejects the Extension Condition as being incompatible with the general definition of recursion, namely that any object that is generated is accessible for further operations (e.g. a subpart of β in (1) is accessible to Internal Merge, i.e. can be extracted from β; if that is the case, merger to a subpart of β can only be blocked by stipulation, as observed earlier by Van Riemsdijk 1998).

b. Binary Merge

Merge is also taken to be binary, i.e. involving no more than two lefthand elements in (1). This is motivated by a general simplicity argument, given that binary Merge suffices to derive the hierarchical syntactic structures of natural languages (Chomsky, Gallego and Ott 2019:237; for arguments concerning the binary nature of syntactic structure, see Kayne 1984).

If Merge is asymmetric in the sense of Jaspers (1998), binarity follows from the fact that the minimal application of Merge selects just a single element to merge to the OUC. This conception of Merge is sometimes referred to as *unary Merge* (Zwart 2009, De Belder and Van Craenenbroeck 2015).

c. Inclusiveness Condition

Chomsky (1995:228) states that structures formed by Merge should not contain any elements not already present in the items undergoing Merge, hence no bar levels (familiar from X'-theory), indices, etc. In the earliest stage of Minimalism (Chomsky 1992), Merge was taken to be subject to the principles of X'-theory, but this was abandoned with the articulation of the labeling mechanism, ushering in a 'bare phrase structure' theory (Chomsky 1995:249).

Labeling the output of Merge is taken to be necessary to enable interpretation at the interface components [ref to other chapter]. The Inclusiveness Condition entails that any label should be constructed from the two elements merged (Chomsky 1995:244), e.g. α in (1), leading to the notation of the output of Merge (1) as (11), for a constituent *αβ* with label α:

(11) $\{ \alpha, \{ \alpha, \beta \} \}$

Labels are considered a redundant notational device in Chomsky (2000a:135; see also Collins 2002 and Chomsky, Gallego and Ott 2019:247).

5. Types of Merge

a. internal and external Merge

Given the definition of Merge in (1), Merge is said to be *internal* when α is contained in β, and *external* in the absence of a containment relation between α and β.

In comparison with earlier stages of generative grammar, External Merge represents the rewrite rules of the base component (Chomsky 1957:26f), including generalized transformations (Chomsky 1965:132f), whereas Internal Merge represents the movement rules of the transformational component (Chomsky 1957:44), including the generic Move α rule of the pre-minimalist Principles and Parameters framework (Chomsky 1981, 1982). However, since Merge operates freely, External Merge and Internal Merge are not assigned to ordered blocs (such as the earlier base and transformational components), eliminating the basis for a distinction between D-structure (deep structure) and S-structure (surface structure). This is consistent with the minimalist desideratum that the model of grammar should articulate no levels other than the interface levels (Chomsky 2000b:10).

See [the next vignette] on further aspects of Internal Merge.

b. set Merge and pair Merge

Chomsky (1995:248) proposes to distinguish set Merge (illustrated in (1), repeated with label in (12)) from pair Merge, originally (*loc. cit.*) defined with a label that is an ordered pair (13a), later (Chomsky 2000a:134) as Merge resulting in an ordered pair with a simple label (13b).

(12) set Merge

$\alpha, \beta \rightarrow \{ \alpha, \{ \alpha, \beta \} \}$

(13) pair Merge

a. $\alpha, \beta \rightarrow \{ \langle \alpha, \alpha \rangle, \{ \alpha, \beta \} \}$

b. $\alpha, \beta \rightarrow \{ \alpha, \langle \alpha, \beta \rangle \}$

Set Merge was taken to represent substitution, pair Merge adjunction. The intuition underlying (13b) was that adjunction yields an asymmetric pair of elements.

However, the circumstance that one of α, β provides the label for {α, β} makes the output of set Merge inherently asymmetric (Chomsky 1995:246, cf. Zwart 2011; in fact, {{α}, {α, β}} is the set-theoretical notation of ⟨α, β⟩, see Langendoen 2003:310, going back to Kuratowski 1921). It is not clear, then, that there is a difference between substitution and adjunction in terms of the (a)symmetry of the constituent resulting from Merge. It has been argued that the simplest conception of Merge is that it invariably involves adjunction (Epstein, Kitahara and Seely 2014, Zwart 2011:103).

The distinction between set Merge and pair Merge did not prove productive and is deprecated in Chomsky, Gallego and Ott (2019:249), but revived in Chomsky (2019b, lecture #4) to account for unbounded coordination.

c. late Merge

Late Merge (adopted in Chomsky 1995:205 from a pre-minimalist analysis by Lebeaux 1994) is an instance of countercyclic adjunction (i.e. violating the Extension Condition) to account for unexpected suspension of Principle C-effects in cases like (14b).

(14) a. Which claim that John was asleep was he willing to discuss? (*John* ≠ *he*)
b. Which claim that John made was he willing to discuss? (*John* = or ≠ *he*)

The idea is that the adjunct clause in (14b) *that John made* is externally merged with *claim* after *which claim* has been fronted (which is not an option in (14a) where *that John was asleep* is a complement clause). Consequently *John* is not in the scope of *he* at any point in the derivation, obviating the expected Principle C-effect.

Late Merge is frequently resorted to in analyses of the interaction of movement and scope (e.g. Sauerland 1998, Fox 1999, Takahashi and Hulsey 2009), in spite of questions about its propriety within minimalism (e.g. Epstein, Kitahara and Seely 2014, Sportiche 2019, Chomsky, Gallego and Ott 2019).

d. self-Merge

Self-Merge arises where the elements to be merged are identical. It was proposed in Guimarães (2000) to allow vacuous (non-branching projection), and further developed in Kayne (2011:332f), Adger (2013:19f). Formally, it takes an element α and yields the singleton set {α}:

(15) Self-Merge

$$\alpha \rightarrow \{ \alpha \}$$

In Adger (2013), recursive Self-Merge turns acategorial roots into interpretable syntactic objects (characterized by the categorial and functional features associated with the self-Merged object after each step), allowing him to dispense with functional heads (as the source of these features) and functional projections in the analysis of syntactic structure.

e. parallel Merge

Parallel Merge (Citko 2000) is a variant of Internal Merge, where α, a subpart of β, is not merged with β but with a third, independent element γ (a process also referred to as 'sideward movement', cf. Nunes 2004). [See the next vignette for a further exposition.]

f. bottom-up and top-down ('split') Merge

Merge is standardly taken to be a bottom-up structure building operation, as illustrated above. Top-down structure building approaches have been proposed by Phillips (2003) and Zwart (2009).

Phillips essentially proposes to derive (5) by the sequence of operations Merge in (16), expanding the tree at the bottom rather than at the root.

(16) top-down Merge (Phillips) [where e.g. α = *loves*, β = *Mary*, and γ = *John*]
1. γ, α → { γ, α } [yielding *John loves*]
2. α, β → { α, β } [yielding *loves Mary*]

This allows Phillips to derive Right Node Raising phenomena like (17) in a straightforward way, the required coordination taking place between steps 1 and 2 in (16).

(17) John loves and Bill hates Mary

An analysis along these lines was further developed by Chesi (2007).

Zwart (2009), following a suggestion by Fortuny, proposed to derive (5) by subsequently splitting off elements from an initial, unordered set, yielding ordered pairs with each step:

(18) split Merge (Zwart)

a. $\{ \gamma, \alpha, \beta \} \rightarrow \langle \gamma, \{ \alpha, \beta \} \rangle$
b. $\{ \alpha, \beta \} \rightarrow \langle \alpha, \{ \beta \} \rangle$

As shown by Zwart (2017), split Merge is formally a case of Internal Merge. Assuming that the element split off from the initial set leaves a copy, the output of each split Merge operation is an ordered pair by the set-theoretical definition of ordered pairs of Kuratowski (1921):

(19) split Merge as Internal Merge (Zwart 2017)

$\{ \alpha, \beta \} \rightarrow \{ \alpha, \{ \alpha, \beta \}\} \equiv \langle \alpha, \beta \rangle$

This would eliminate External Merge, leaving Internal Merge as the only possible instantiation of Merge (see also Chomsky 2019b, lecture #2).

6. Merge and the architecture of Narrow Syntax

a. Merge in the model of grammar

The function of Merge in the minimalist model of grammar is to turn an unordered set of elements into a structured object that is interpretable to processes of the interface components dealing with sound (including morphological shape and linear order) and meaning. It is therefore the key operation of Narrow Syntax, and the definining property of the Faculty of Language in the Narrow sense (FLN) of Hauser, Chomsky and Fitch (2002).

Since the output of Merge is to be further processed by the interface components, there must be an operation Transfer connecting Narrow Syntax to these interface components. This transfer takes place after a certain sequence of operations Merge called a Phase (Chomsky 2001), though other scenarios have been argued for as well (Uriagereka 1999, Epstein and Seely 2002).

The starting point of each derivation is typically taken to be a (random) selection of elements, a set or an array referred to as the Numeration (Chomsky 1995:225).

b. Numeration/workspace

The Numeration was initially (Chomsky 1995:225) taken to be a set of pairs (LI, *i*), where LI is an item of the lexicon, and *i* its index, essentially a counter tracking the number of times LI is merged. It was crucial in upholding a version of global economy, where the most economical derivation could be selected from the set of possible derivations based on a particular Numeration (instead of from all possible derivations, which would be intractible). This concept of Numeration is now no longer entertained (cf. Chomsky, Gallego and Ott 2019:237), and replaced by the general conception of a "workspace" containing syntactic objects over which Merge operates (ibid: 236).

c. Merge and the Numeration/workspace

The earliest conception of the relation between Narrow Syntax and the Numeration was that Merge selects objects from the Numeration and creates a new object, which we may call the object under construction (OUC), in Narrow Syntax. This required an operation Transfer (from Numeration to Narrow Syntax) as being inherent to External Merge.

This conception of the relation between Narrow Syntax and the Numeration was criticized by Bobaljik (1995), arguing that Merge does not transfer anything out of the Numeration but merely defines relations among objects in the Numeration (which we may now call the workspace), creating new objects in the process. Thus, in our example (5), the workspace consists of { α, β, γ }, and Merge defines a relation between α and β, yielding a new object (say, A), and a relation between γ and A next. (It will be seen that split Merge does the exact same thing, defining a relation between γ and { α, β } first, and a relation between α and β next.)

On this conception of Merge, the workspace is enriched with each step, since Merge creates in the workspace a new element { α, β } out of α and β, but α and β remain available in the workspace for further operations, such as internal Merge (in fact remerge of another occurrence of α or β).

d. derivational complexity

The question what kind of elements are in the Numeration/workspace (and why) is not generally explicitly addressed (but see Zwart 2015). A useful starting point may be the principle in (20).

(20) Every hierarchically structured element must be derived by Merge

Since words have hierarchical structure, they must be derived by Merge, too. It follows that subparts of words (morphemes, including formal features) may be included in the workspace (at the very least, those workspaces involved in the derivation of words). One possible position, then, would be that the workspace invariably consists of morphemes (cf. Boeckx 2015), or even phonemes (as in Kayne 2019).

On the other hand, (20) is also consistent with a model in which words (and also phrases and clauses) may be generated in separate derivations, yielding packets of form and meaning which may be included in a workspace as well (e.g. the 'multiple spell-out' approach of Uriagereka 1999, or the 'layered derivation' approach of Zwart 2009).

It will be observed that there is a trade-off between the conception of the workspace as being homogeneous (only morphemes) or heterogeneous (any kind of linguistic object) on the one hand, and the complexity of the derivation on the other. If the workspace is homogeneous, higher order linguistic elements such as words, phrases and clauses must be generated in the course of the derivation, necessitating parallel tracks [see next vignette]. On the other hand, if the workspace may contain higher order elements such as words, phrases and clauses (generated in separate derivations feeding the workspace), in addition to morphemes, Merge can proceed in the straightforward fashion described in section 2 (see Zwart 2015 for discussion).

e. MERGE

Chomsky (2019b, lecture #3) introduces 'capital merge' (notated *MERGE*) as a different operation from Merge as discussed above. In its simplest form, MERGE operates on a workspace containing two elements P, Q (notated in straight brackets) and creates a new workspace containing just the set {P, Q}:

(21) MERGE
[P, Q] → [{ P, Q }]

This operation derives exocentric structures, such as the subject-predicate combination.

Crucially, the new workspace that is the output of MERGE contains only the set {P, Q}, not in addition P and Q as individual elements. In this sense MERGE is different from Merge as discussed above (see section c.).

The concept of MERGE addresses the problem of how to engage with previously construed constituents while keeping the derivation maximally simple (i.e. without parallel Merge, sideways movement, late Merge, etc.). The solution of MERGE is to reduce the number of elements available for further syntactic operations. (This is different from ordinary Merge, which is not an operation on a workspace, but on elements within a workspace, which, after having been merged, remain available in the workspace for another operation Merge, effectively giving us Internal Merge.)

7. The trigger for Merge

Merge is justified as the simplest operation that turns an unordered set (the Numeration or workspace) into a hierarchically structured object, thereby serving interpretability at the conceptual-intentional interface. Viewed this way, Merge has a natural trigger (the absence of hierarchical relations among elements in the Numeration/workspace) and a natural end point (when all elements in the Numeration/workspace have been hierarchically ordered).

In early minimalism, this led to the idea that External Merge is cost-free, and hence preferred over Internal Merge, which was in need of an additional trigger (or should be procrastinated as much as possible, 'Merge over Move', e.g. Chomsky 1995:346). Such a trigger was found in the strength of inflectional features in need of checking (Chomsky 1995:233), a notion later abandoned (Chomsky 2000a:132), and replaced by a generic EPP-feature residing in phase heads (C and v) attracting elements to the phase edge (specifier of CP and vP)(Chomsky 2001:12). More recently, Chomsky has argued that Internal Merge takes place to remove a 'problem of projection', when labeling of a constituent could not take place unless one of the daughters of the constituent move (Chomsky 2013); as Chomsky shows, this provides an elegant new account of successive cyclic movement, for which the question of what triggers Internal Merge was particularly perplexing.

Quite apart from this is the question of what motivates the selection of elements to include in a Numeration/workspace, preceding any operation Merge. This creative aspect of human language lies beyond current attempts to model the faculty of language (Chomsky 1995:227, 2019b lecture #4).

References.

Adger, David. 2013. *A syntax of substance*. Cambridge: MIT Press.

Bobaljik, Jonathan D. 1995. In terms of Merge. *MIT Working Papers in Linguistics* 27, 41-64.

Boeckx, Cedric. 2015. *Elementary syntactic structures*. Cambridge: Cambridge University Press.

Chesi, Christiano. 2007. An introduction to phase-based minimalist grammars: why move is top-down and from left-to-right. *CISCL Working Papers on Language and Cognition* 1, 38-75.

Chomsky, Noam. 1957. *Syntactic structures*. The Hague: Mouton.

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge: MIT Press.

Chomsky, Noam. 1981. *Lectures on government and binding*. Dordrecht: Foris.

Chomsky, Noam. 1982. *Some concepts and consequences of the theory of government and binding*. Cambridge: MIT Press.

Chomsky, Noam. 1986. *Barriers*. Cambridge: MIT Press.

Chomsky, Noam. 1992. A minimalist program for linguistic theory. *MIT Occasional Papers in Linguistics* 1.

Chomsky, Noam. 1995. *The minimalist program*. Cambridge: MIT Press.

Chomsky, Noam. 2000a. Minimalist inquiries: the framework. In Roger Martin, David Michaels and Juan Uriagereka, eds., *Step by step: minimalist essays in honor of Howard Lasnik*, 89-155. Cambridge: MIT Press.

Chomsky, Noam. 2000b. *New horizons in the study of language and mind*. Cambridge: Cambridge University Press.

Chomsky, Noam. 2001. Derivation by phase. In Michael Kenstowicz, ed., *Ken Hale: a life in language*, 1-52. Cambridge: MIT Press.

Chomsky, Noam. 2013. Problems of projection. *Lingua* 130, 33-49.

Chomsky, Noam. 2019a. Some puzzling foundational issues: the Reading program. *Catalan Journal of Linguistics special issue*, 263-285.

Chomsky, Noam. 2019b. The UCLA lectures. Ms., University of Arizona.

Chomsky, Noam, Ángel J. Gallego and Dennis Ott. 2019. Generative grammar and the faculty of language: insights, questions, and challenges. *Catalan Journal of Linguistics special issue*, 229-261.

Citko, Barbara. 2000. *Parallel merge and the syntax of free relatives*. PhD dissertation, Stony Brook University.

Collins, Chris. 2002. Eliminating labels. In Samuel D. Epstein and T. Daniel Seely, eds., *Derivation and explanation in the minimalist program*, 42-64. Malden: Blackwell.

De Belder, Marijke and Jeroen van Craenenbroeck. 2015. How to merge a root. *Linguistic Inquiry* 46, 625-655.

Epstein, Samuel D. 1999. Un-principled syntax: the derivation of syntactic relations. In Samuel D. Epstein and Norbert Hornstein, eds., *Working minimalism*, 317-345. Cambridge: MIT Press.

Epstein, Samuel D., Hisatsugu Kitahara and T. Daniel Seely. 2014. Labeling by Minimal Search: implications for successive-cyclic A-movement and the conception of the postulate 'phase'. *Linguistic Inquiry* 45, 222-239.

Epstein, Samuel D. and T. Daniel Seely. 2002. Rule application as cycles in a level-free syntax. In Samuel D. Epstein and T. Daniel Seely, eds., *Derivation and explanation in the minimalist program*, 65-89. Malden: Blackwell.

Fox, Danny. 1999. Reconstruction, variable binding, and the interpretation of chains. *Linguistic Inquiry* 30, 157-196.

Guimarães, Maximiliano. 2000. In defense of vacuous projections in Bare Phrase Structure. *University of Maryland Working Papers in Linguistics* 9, 90-115.

Hauser, Marc D., Noam Chomsky and W. Tecumseh Fitch. 2002. The faculty of language: what is it, who has it, and how did it evolve? *Science* 298, 1569-1579.

Jaspers, Dany. 1998. Categories and recursion. *Interface* 12, 81-112.

Kayne, Richard S. 1984. *Connectedness and binary branching*. Dordrecht: Foris.

Kayne, Richard S. 2011. Antisymmetry and the Lexicon. In Anna Maria di Sciullo and Cedric Boeckx, eds., *The biolinguistic enterprise: new perspectives on the evolution and nature of the human language faculty*, 329-353. Oxford: Oxford University Press.

Kayne, Richard S. 2019. What is suppletion? On **goed* and on *went* in Modern English. *Transactions of the Philological Society* 117, 434-454.

Kitahara, Hisatsugu. 1994. *Target α: a unified theory of movement and structure-building*.

PhD dissertation, Harvard University.

Kuratowski, Kasimierz. 1921. Sur la notion de l'ordre dans la théorie des ensembles. *Fundamenta mathematicae* 2, 161-171.

Langendoen, D. Terry. 2003. Merge. In Andrew Carnie, Heidi Harley, and MaryAnn Willie, eds., *Formal approaches to function in grammar: in honor of Eloise Jelinek*, 307-318. Amsterdam: John Benjamins.

Lebeaux, David. 1994. Where does the binding theory apply? Ms., University of Maryland.

Nunes, Jairo. 2004. *Linearization of chains and sideward movement*. Cambridge: MIT Press.

Phillips, Colin. 2003. Linear order and constituency. *Linguistic Inquiry* 34, 37-90.

Van Riemsdijk, Henk. 1998. Trees and scions—science and trees. *Chomsky 70th birthday celebration*. Ms., University of Tilburg.

Sauerland, Uli. 1998. *The meaning of chains*. PhD dissertation, MIT.

Sportiche, Dominique. 2019. Somber prospects for late merger. *Linguistic Inquiry* 50, 416-424.

Takahashi, Soichi and Sarah Hulsey. 2009. Wholesale late merger: beyond the A/A'-distinction. *Linguistic Inquiry* 40, 387-426.

Uriagereka, Juan. 1999. Multiple Spell-out. In Samuel D. Epstein and Norbert Hornstein, eds. *Working minimalism*, 251-282. Cambridge: MIT Press.

Zwart, Jan-Wouter. 2009. Prospects for top-down derivation. *Catalan Journal of Linguistics* 8, 161-187.

Zwart, Jan-Wouter. 2011. Order and structure: asymmetric merge. In Cedric Boeckx, ed., *The Oxford Handbook of Linguistic Minimalism*, 96-118. Oxford: Oxford University Press.

Zwart, Jan-Wouter. 2015. Top-down derivation, recursion, and the model of grammar. In Andreas Trotzke and Josef Bayer, eds., *Syntactic complexity across interfaces*, 25-42. Berlin: De Gruyter Mouton.

Zwart, Jan-Wouter. 2017. Eliminating external merge. Paper presented at the Incontro di Grammatica Generativa 43, October 5, IUSS Pavia.