

The Search for Minimal Search: A Graph-Theoretic Approach

Diego Gabriel Krivochen¹ 

[1] *Faculty of Linguistics, Philology and Phonetics, University of Oxford, Oxford, United Kingdom.*

Biolinguistics, 2023, Vol. 17, Article e9793, <https://doi.org/10.5964/bioling.9793>

Received: 2022-06-28 • **Accepted:** 2023-01-09 • **Published (VoR):** 2023-03-01

Handling Editor: Kleanthes K. Grohmann, University of Cyprus, Nicosia, Cyprus

Corresponding Author: Diego Gabriel Krivochen, Faculty of Linguistics, Philology and Phonetics, University of Oxford, Clarendon Institute, Walton Street, Oxford, United Kingdom, OX1 2HG. E-mail: diego.krivochen@lingphil.ox.ac.uk

Abstract

This paper examines Minimal Search, an operation that is at the core of current Minimalist inquiry. We argue that, given Minimalist assumptions about structure building consisting of unordered set-formation, there are serious difficulties in defining Minimal Search as a search algorithm. Furthermore, some problematic configurations for Minimal Search (namely, {XP, YP} and {X, Y}) are argued to be an artefact of these set-theoretic commitments. However, if unordered sets are given up as the format of structural descriptions in favour of directed graphs such that Merge(X, Y) creates an arc from X to Y, Minimal Search can be straightforwardly characterised as a sequential deterministic search algorithm: the total order required to define MS as a sequential search algorithm is provided by structure building.

Keywords

set theory, graph theory, Minimal Search, phrase structure

1 Introduction

Minimal Search (MS henceforth) was introduced as a mechanism underlying syntactic dependencies in Chomsky (2004, p. 109) (although mentions to ‘search’ in the context of Agree and ‘reducing search space’ as a goal of economy conditions can be found earlier, e.g., Chomsky, 2000, pp. 99, 122) and developed into a central part of Minimalist theorising in the context of the Problems of Projection program, as a ‘third factor’ property (Chomsky, 2013, p. 43; 2020, p. 48) which ‘falls under MC [Minimal Computation]’ (Chomsky, 2015, p. 6). Similar characterisations of MS can be found e.g. in Epstein et al.



This is an open access article distributed under the terms of the [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/), CC BY 4.0, which permits unrestricted use, distribution, and reproduction, provided the original work is properly cited.

(2015, 2020), Chomsky (2020, p. 36, 2021, p. 18), Goto (2019), Komachi et al. (2019), van Gelderen (2021, p. 4), Larson (2015), Hayashi (2021), among others.

MS has become a key component in the formulation of various syntactic operations. To name but a few, Chomsky (2013, p. 43, 2015, p. 6), Bauke and Blümel (2017, p. 4), Ke (2022), and Epstein et al. (2015, 2020), capitalise on MS for labelling. Collins (2017, p. 58) appeals to MS in a definition of linearisation (which highlights its fundamentally sequential nature, an issue we will return to). Chomsky (2013, 2015, 2020, p. 36) resorts to it in a discussion about subject movement as a precondition for labelling (seeking to eliminate the EPP feature) and accessibility in successive-cyclic movement. Branen and Erlewine (2021), Ke (2019, 2022), Preminger (2019), and Milway (2021) analyse MS in the context of Agree. MS underlies formal discussions about accounts of empirical phenomena (displacement, agreement). However, characterisations of MS in the literature are often informal (with some exceptions: e.g., Chow, 2022; Ke, 2019, 2022; Milway, 2021; Preminger, 2019), as in Chomsky (2020, p. 36):

[MS is] *a third factor principle, that says look for the closest copy and don't look any further*

Or the semi-formal formulation in Chomsky (2021, p. 18):

Minimal Search (MS): Σ searches as far as the first element it reaches and no further. In searching WS [WorkSpace], MS selects a member X of WS, but no term of X. In the optimal case of selection of a single SO [Syntactic Object] Z, Σ selects the first relevant term Y of Z, and stops.

In this context, it is essential to address the following questions: how do the properties of MS relate to those of structure building? Is there a way to define MS in a way that is both formally explicit and empirically fruitful? We argue that there are difficulties that arise when we attempt to define MS under the assumption that Merge produces unordered sets, difficulties that are mirrored in other areas of the grammar such as argument structure. Given that a search algorithm seems to be required independently in syntactic theory (e.g., in the treatment of Agree and long-distance dependencies), this constitutes an argument for exploring a format for syntactic representations not based on unordered sets. We will provide an alternative formalism for structural descriptions which departs from the set-theoretic foundations of current Minimalist syntax and which not only allows us to define MS, but also has independent empirical advantages.

2 Two Perspectives on Structure Building

In Minimalist theorising, Merge / MERGE (in what follows we will use 'Merge' for convenience, unless a technical point hinges in the distinction between 'Merge' and 'MERGE') are instances of an allegedly irreducible operation of unordered set formation

over elements in a workspace (Chomsky, 2019 et seq.; Collins, 2017; Epstein et al., 2015): this is intended to capture the idea that (i) syntax is based on unbounded recursion, delivering structure dependence (Chomsky, 2021, p. 13), and (ii) *linear* order (the binary, transitive, total relation *precedes* over lexical terminals) is not part of the ‘syntactic component’ of the grammar (but instead of the ‘externalization systems’; Chomsky, 2019, p. 272). We will return to the properties of syntactic objects (SOs) generated by Merge in Section 5, but the issue of order is relevant now. To give an example, Chomsky (2019, p. 257) defines the *Inclusiveness Condition* such that ‘*the operations should not add anything: they should not add order*’ (cf. Chomsky, 1995a, pp. 228, 381). Importantly, however, he does not specify if he refers only to linear precedence or if *any* order imposed over the output of Merge is ruled out. Let us see why this matters. In set-theoretic syntax, the structural description for a sentence like *The man falls* is (1) (taken from Collins, 2017, p. 65):

- (1) {{the, man}, { \emptyset , T}, {fall, {the, man}}}}

We contend that there are reasons to prefer an *asymmetric* definition of Merge. This asymmetry entails imposing an order over the output of the operation. As has been noted in the literature, the commitment to unordered sets creates problems for the representation of empirically motivated syntactic relations such as subcategorisation and argument structure (Langendoen, 2003, §2.3), and also labelling and chains dependencies (Gärtner, 2002, 2022). As an initial step in our inquiry, we propose that the former is a cornerstone of natural language syntax and -as such- a good basis to ground the asymmetry of Merge¹. Note, in this context, that unless selection is encoded in terms of containment, there is nothing in (1) that tells us that *{the, man}* is an argument of *{fall}*, all we know is that there is a set that contains *{the, man}* and excludes *{fall}*. What is more, if instead of *{the, man}* we had *{John}*², then *{fall, John}* would provide no asymmetric relation between a predicate and its argument. If we (reasonably) require of an empirically successful theory of syntax to capture argument structure, we must be able to define an order over the combination of the expressions *fall* and *John* or *fall* and *the man*. Chomsky and others have emphasised that linear order is independent

1) A reviewer suggests that ‘*a more generalized conception of asymmetric Merge should be possible, perhaps referring to the process of structure building itself*’. Examples of such proposals include Zwart (2011), who provides a definition of asymmetric Merge grounded on LCA-based linearisation, or implementations of feature-driven Merge (e.g., Minimalist Grammars, to which we will come back to below). We have preferred to ground the asymmetry of Merge in what is ultimately a lexical property (subcategorisation), as it will also allow us to make our structural descriptions maximally informative: not only conveying predicate-argument relations, but also grammatical functions can be read off the order between arcs in our proposed structural descriptions.

2) We will go back to the problem posed by the set-theoretic status of *John* in Section 5. We are not making a point about the internal structure of proper names or lack thereof: the same point can be made with a set such as {read, books}, and the same questions about the set-theoretic status of the N complement arise.

of structure building (e.g., Chomsky, 2021, p. 9: ‘linear order is simply not available to *I-language*’), something we agree with. However, it is possible to define an order over a set other than linear precedence. Consider for example the case of predicate-argument relations: if p is a predicate and a is its argument, the relation $p(a)$ imposes an order over the set of expressions $\{a, p\}$ that is not precedence.

One way to address the inadequacy of label-less unordered sets to represent argument structure is to allow Merge to output ordered sets. Langendoen (2003) defines an operation *List Merge* which captures this idea (see also Chomsky, 2001, 2004):

- (2) $L(\alpha, \beta) = \langle \alpha, \beta \rangle$ (if α is atomic)
 $L(\langle \alpha, \dots \rangle, \beta) = \langle \alpha, \dots, \beta \rangle$ (if α is a phrase)

This order is important in order to define, among other things, hierarchy and semantic interpretation: given a set of lexical items $LI = \{\dots, John, run, \dots\}$, an operation that delivers the unordered set $\{run, John\} = \{John, run\}$ does not provide the same information as one that delivers the ordered set $\langle run, John \rangle$: only the latter can be put in direct correspondence with a propositional representation along the lines of $run(John)$, where it is clear that *John* is an argument of *run* regardless of how those terminals are linearised (Heim & Kratzer, 1998, p. 44), without further theoretical machinery. Langendoen provides explicit derivations where this recursive definition of ordered sets under *List Merge* is used (assuming the Wiener-Kuratowski definition of ordered sets; Krivine, 1971, pp. 2–3. See Gärtner, 2002, p. 74, fn. 146; Dipert, 1982 for critical discussion in syntax and mathematical logic respectively). Despite his criticisms to Merge imposing no order, Langendoen remains committed to sets as the format of structural descriptions (with trees being used only as diagrams of derivations).

We want to take this objection to unordered sets a step further: we require of an adequate format for syntactic structures that they not only encode asymmetric relations (such as selection, which could also be encoded in feature-driven Merge), but also do it in such a way that a search sequence can be defined unambiguously over structural descriptions. For this, we need structure building to impose an order over expressions. The idea that syntactic representations are ordered (in ways other than precedence) has a long tradition in generative grammar, beyond set-theoretic commitments: to give but an example, McCawley’s (1968, 1982b) approach, which interprets phrase structure rules as admissibility conditions for local trees, defines two distinct two-place relations over nodes in trees: *precedes* and *dominates*³³ (see also Higginbotham, 1983; Ojeda, 1988). In

3) As observed by a reviewer, ‘virtually every theory that Chomsky put forward until 1993 also assumed linear order was part of the syntax’. However, these theories were mostly focused on mappings between strings or between levels of representation: structural descriptions in Chomskyan generative grammar were never graphs. McCawley (1982a, p. 165ff.; 1982b, p. 92ff.) makes a strong case for linearly unordered graph-theoretic structures (see also Sag et al. 1985, p.127), and distinguishes between strings and trees consistently.

these two relations we have the seed of our main departure from contemporary Minimalist theorising: the former pertains to linear order in a string; the latter, to hierarchy in structural descriptions which are graphs, not sets.

In this paper we argue that a shift from set theory to graph theory as the model for structural descriptions in natural language syntax is a desirable way to deliver structures where MS can be defined. Within Minimalism, McKinney-Bock & Vergnaud (2014) argue for graph-theoretic structure building, such that Merge generates directed graphs where the order imposed over nodes is based on the relations *Selection* and (feature) *Checking*. Outside Minimalism, some versions of Dependency Grammars (Kahane & Lareau, 2016) as well as Categorical Grammar (Huck, 1984), Arc Pair Grammar (Johnson & Postal, 1980), and Metagraph Grammar (Postal, 2010, p. 26) develop explicit graph-theoretic analyses where linear order is divorced from hierarchical relations. We can leverage these insights within a derivational framework. Let *run* and *John* be basic expressions⁴ of the language, points in the workspace (Krivochen, 2022). Then, Merge(*run*, *John*) defines a syntactic relation between these expressions. In our proposal, instead of creating the set {*John*, *run*}, we define a graph like that in Figure 1, a diagram of a directed edge (an *arc*) from a predicate to its argument⁵:

Figure 1

Directed Edge

run —————→ *John*

Alternatively, we can notate that edge as $e\langle run, John \rangle$: both formats characterise the same formal object. The fundamental aspect of this proposal is that Merge no longer delivers unordered set-theoretic outputs: instead of Merge(X, Y) = { X, Y }, we define Merge(X, Y) = $e\langle X, Y \rangle$. In this way, Merge imposes an order over the expressions *run* (a lexical 1-place predicate) and *John* (its argument) that plays two important roles: (i) it defines a unique sequence for purposes of MS, and (ii) encodes argument structure.

The goal of this paper is twofold: to define MS as a search algorithm and to characterise a format for syntactic structure where MS can apply and which can be independently

4) Schmerling (2018) defines *basic expression* as follows:

The basic expressions of [a language] L are simply the expressions that are not derived by any operation – they can be thought of as comprising the language’s lexicon. If L has a non-empty set of operations (as any natural language does), the outputs of those operations are derived expressions. (Schmerling, 2018, p. 16. Highlighting ours)

5) McKinney-Bock and Vergnaud (2014, p. 220) use arrows to indicate that the tail of the arc (here, *John*) projects, but we depart from that convention.

justified. In this light, we will argue that (i) structure building must encode selection and argument structure, which can be accomplished with directed graphs, and (ii) this leads to a natural characterisation of MS as a sequential search algorithm (in addition to avoiding or solving problems that arise independently in set-theoretic syntax).

The paper is structured as follows. Section 3 introduces search algorithms. Section 4 defines graphs and compares the properties of sets and graphs as the output of generative operations. Section 5 discusses the implications of Merge-as-unordered-set-formation for MS in the ‘easy cases’ (i.e., $\{X, YP\}$). Section 6 discusses aspects of the labelling algorithm related to MS. Section 7 focuses on the ‘hard cases’ for MS: $\{XP, YP\}$ and $\{X, Y\}$.

3 What Is a Search Algorithm?

In computer science, a search algorithm is a sequence of well-defined, implementable instructions that retrieves some information stored in a data structure; in other words, a sequence of steps to locate a memory address and retrieve the information contained in that address (Knuth, 1998; Sedgewick & Wayne, 2014). Informally, a search algorithm probes into a sequence of ‘addresses’ or ‘keys’, where it looks for a target value. If the target value is found after a finite number of comparisons between the target value and values in the input data, the search is successful. This process can be expressed in terms of string searches, such that given a string s we can look for the first occurrence of substring u of arbitrary length (shorter than s).

We can distinguish between sequential, parallel, and random search. In a sequential search, values in a data structure are read one at a time starting from a root: this corresponds to the characterisation of MS in Chomsky (2020, 2021) and the vast majority of the Minimalist literature on Agree, Label, and even linearisation; a formal definition of sequential search for Agree is provided in Preminger (2019). In a parallel search, the search also starts from a root node, but given multiple searching processors more than one value is accessed at any given time (Ke, 2022, pp. 9–10). Finally, in a random search the starting point is not pre-defined. In the simplest case of random search, the method homogenises the probability distribution of all points within a space and accesses them in a non-fixed sequence, while keeping track of the best approximation to the target it has found (Romeijn, 2008). Here we will deal only with sequential search methods for two reasons: first, most of the characterisations of MS we can find in the literature correspond to a sequential search. Second, syntactic structures have an inherent order (based on set (co-)membership) imposed by the sequentiality of Merge (and leveraged, e.g., for the derivational definition of c -command in Epstein, 2000, p. 150⁶), and if MS is

6) A comment is in order (pun intended): given $\{a, \{b, c\}\}$ we can tell that a was introduced in the derivation later than b and c , and that b and c are co-members of a set that excludes a . Given Epstein’s definition of c -command, we also know that a c -commands b and c (and b and c c -command each other, by virtue of having been paired by Merge;

a third-factor operation (Chomsky, 2020, pp. 36, 48; van Gelderen, 2021, p. 4), it should build only on the properties of Merge and its output. We will also see that one of the main arguments for parallel search, that it delivers an ambiguity in {XP, YP} objects, is not a desirable consequence.

A simple sequential search algorithm can be exemplified as follows (Knuth, 1998, p. 396ff.): suppose we have a set of values $Val = \{V_1, \dots V_n\}$, each of which is assigned a *key* (a unique address that allows us to retrieve the value) $Key = \{K_1, \dots K_n\}$. Then, a search algorithm may be called upon to find a specific K_x (in our case, the target may be a feature that matches a probe's specification). The algorithm starts at K_i , checks whether $K_i = K_x$, and if it is, the procedure terminates. If not, it proceeds to K_{i+1} and checks again; the procedure is repeated until reaching K_n . The algorithm knows *what* to look for in advance (namely, K_x), and includes a procedure of comparison such that each key in the sequence is compared with the target value. Furthermore, the sequence is ordered, such that the search can proceed from K_i ($i = 1$) all the way to the end of the input sequence (K_n) and not leave anything unchecked. Variations of this algorithm are possible if the keys are themselves ordered (so instead of $\{K_i, K_j, \dots K_n\}$ we have $K_i < K_j < \dots K_n$), if the algorithm is sensitive to the number of times a particular key has been accessed if more than once, etc. A parallel search algorithm requires multiple processors with a shared control unit, each of which conducts a sequential search and may or may not report search results to other processors. Parallel search may be carried out by concurrent or exclusive processors, depending on whether more than one can read/write simultaneously from/on a given memory location. Most search algorithms are optimised such that backtracking is avoided if possible.

Search algorithms have been devised not only for table-ordered datasets (e.g., a phonebook), but also for datasets structured in tree form, where each node in the tree is assigned a uniquely identifying address: this is sometimes known as a Gorn addressing scheme (Gorn, 1967a, 1967b). The addressing system will play a central role in our analysis, as the basic expressions that correspond to nodes in our graphs will be indexed by a set of addresses, following Gorn's insight. Despite Merge being explicitly defined as an operation that delivers unordered sets, recent characterisations of MS have sometimes implicitly assumed tree search algorithms (e.g., Branen & Erlewine, 2021; Chow, 2022; Ke, 2019, 2022; Preminger, 2019, p. 24), as have some approaches to structure linearisation as a tree traversal (e.g., Kural, 2005, and Kremers, 2009; from a different viewpoint, Medeiros, 2021 also makes use of traversals to model word order variation in a non-transformational model). The problematic relation between trees and sets is

there is no order between sisters). We do not know, however, whether c is an argument of b , or vice-versa, nor do we know what relation a establishes with b and/or c . For purposes of MS, b and c are not ordered with respect to each other.

sometimes acknowledged; other times, as in Ott (2012, p. 9); Epstein et al. (2020, p. 2), trees and sets are taken as notational variants of each other.

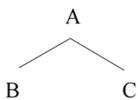
Sequential tree search algorithms, as assumed widely in the Minimalist literature, are broadly divided in two kinds: *breadth-first* and *depth-first*. Both have access to every node in a connected tree, but differ in terms of the order in which the search proceeds.⁷

A *breadth-first* search goes ‘in waves’ from the root, searching each generation from left to right⁸ before proceeding to the next generation. A node is marked as the start node, then all nodes adjacent to it are accessed one by one (and their values are added to a queue) until exhausting the set of nodes adjacent to the starting node. Then, the process continues at the next generation.

A *depth-first* search also starts from the root, but instead of going through all the nodes in a generation before proceeding to the next, it explores the leftmost branch exhaustively before backtracking to the last branching node visited and proceeding with the immediately adjacent branch (Cormen et al. 2001, p. 531ff.; Even & Even, 2012, pp. 11, 46–49). We can illustrate the order in which nodes are visited in a simple rooted tree such as that in Figure 2 for both search algorithms (see Figure 3):

7) Recall that in a *parallel* search all daughters of the root are explored simultaneously to find heads bearing specific features (Ke, 2022, pp. 8–9). New routines are initiated for each set to be looked into. This allows the algorithm to find two targets at the same time, which Ke argues (for reasons unclear to us) is a desirable outcome.

8) The order in which nodes are checked depends on the type of traversal algorithm chosen, since there are three possible ways to implement it. Take a branching node A immediately dominating nodes B on the left and C on the right as an example:



A *preorder* traversal defines the sequence: A B C (root, left, right)

An *inorder* traversal defines the sequence: B A C (left, root, right)

A *postorder* traversal defines the sequence B C A (left, right, root)

The choice of *preorder* traversals in this paper (whenever trees are involved) obeys considerations of how the graph has been constructed and what MS is supposed to accomplish: for purposes of labelling, for instance, if we follow the generative literature, B and C are already labelled. The node that should be looking for a target is A, not B or C; thus, it makes sense to start from A. Preminger’s (2019) approach to MS for Agree also defines a preorder: specifiers are checked first.

Figure 2

Binary Branching Tree

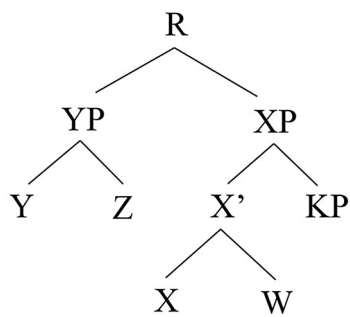
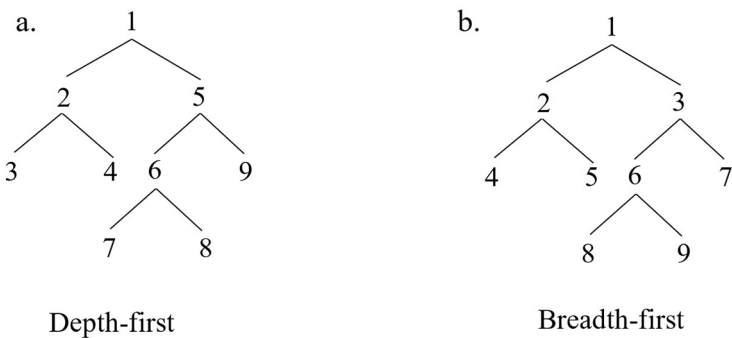


Figure 3

Illustration of Depth-First and Breadth-First Searches



Suppose that the search starts from the root, and the target value is Y (a terminal node). A *depth-first* traversal would define the search sequence Σ , indicating only the nodes in the order they are visited:

(3) $\Sigma = \langle R, YP, Y \rangle$

The algorithm, at every node visited, compares the element in its input with its target value (a head or a valued feature of a specific kind). If the scanned symbol matches the target value, the algorithm halts; otherwise, it keeps going. This mechanism underlies both depth-first and breadth-first algorithms, the only thing that changes is the order in

which nodes are visited. For a breadth-first algorithm (assuming a *preorder*), the sequence would be:

$$(4) \quad \Sigma = \langle R, XP, YP, Y, Z, X', KP, X, W \rangle$$

In this example both algorithms find Y before any other head; however, a depth-first search finds Y after visiting two nodes, whereas a breadth-first search finds Y after visiting three.

As highlighted above, the Minimalist framework in which MS plays a major role is strongly committed to the idea that structural descriptions are *sets* (see also Section 5): trees in most current versions of Minimalism are diagrams, not formal objects. Since there is existing literature on MS as tree traversal, and we have proposed a definition of asymmetric Merge whose output is a directed graph, we must now formally define graphs and compare them to Minimalist sets.

4 Graphs and Sets

A graph is a pair $G = (V, E)$, where V is a set of vertices (or *nodes*) and E is a set of edges; $v \in V$ is a vertex, and $e \in E$ is an edge. If edges in a graph are ‘one-way roads’ connecting a *head* and a *tail*, they are referred to as *directed edges* or *arcs*, and the graph is a *directed graph* (or *digraph*), such as Figure 1 above. As above, we will use $e \langle a, b \rangle$ to denote an arc with head a and tail b .

The *indegree* of a vertex v_x is the number of edges of which v_x is a tail; the *outdegree* of v_x is the number of edges of which v_x is a head. For instance, in Figure 1, repeated here

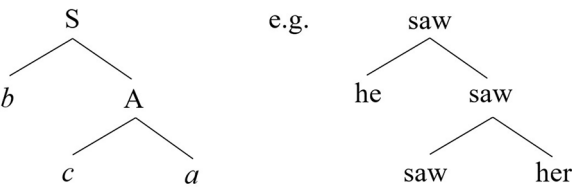
run \longrightarrow John

the *indegree* of *run* is 0, and its *outdegree* is 1; the *indegree* of *John* is 1, and its *outdegree*, 0. Let v_1 and v_2 be two (possibly distinct) vertices in G : a v_1 - v_2 *walk* in G is a finite ordered alternating sequence of vertices and edges that begins in v_1 and ends in v_2 .

Trees are specific kinds of graphs: a *tree* T is a graph that has no loops (there is no walk in T that begins and ends in the same vertex) and is connected (for every v_x, v_y there is a finite walk from v_x to v_y or vice-versa) (van Steen, 2010, p. 113). It is usual in the graph-theoretic literature to reserve the term *tree* for undirected graphs that satisfy the aforementioned conditions; directed trees are sometimes called *arborescences*. The trees used as diagrams of sentence structure in generative grammar are, in addition, *simple* (no vertex can appear more than once in a walk, nor is there more than one edge between any two vertices⁹), *labelled*, and *rooted* (there is a node that is not dominated

by any other node) (see [Huck, 1984: Appendix](#); [McCawley, 1968](#) for discussion). We can illustrate these notions in [Figure 4](#), a standard tree diagram¹⁰, alongside a Bare Phrase Structure (BPS; [Chomsky, 1995b](#)) representation:

Figure 4
Tree Diagram and BPS Representation



In [Figure 4](#), $V = \{a, b, c, A, S\}$. Node A has degree 3, S—the designated root—has degree 2, and all terminal symbols have degree 1. Two aspects are noteworthy: first, because we are still dealing with trees, edges are undirected. Second, if edges correspond to applications of Merge, we do not have $\text{Merge}(c, a)$, but rather $\text{Merge}(A, c)$ and $\text{Merge}(A, a)$. In the BPS tree, there is an additional complication introduced by the appearance of *saw* several times in the tree, delivering a category with multiple segments: we will come back to BPS representations below. Our proposal will depart from trees as the graph theoretic objects we propose to be delivered by Merge; for now, however, we will use a format familiar to the reader to illustrate the differences between sets and graphs.

A set theoretic representation of [Figure 4](#), following recent Minimalist literature, would be (5):

(5) $\{S, \{b, \{A, \{a, c\}\}\}\}$

9) This is because Minimalism rejects multidominance ([Chomsky, 2019](#)), sometimes on the grounds that multidominance trees are graphs, not sets (e.g., [Chomsky et al., 2019](#); [Collins & Groat, 2018](#)). The underlying claim is that graph-theoretic objects are in some unspecified sense more ‘complex’ than sets and not defined by simplest MERGE ([Chomsky et al., 2019, p. 246](#)). See [Gärtner \(2022, p. 133\)](#) for critical discussion about the Copy Theory of Movement pertaining to this point, where it is shown that the supposed ‘complexity’ of graph-theoretic representations may be necessary to circumvent the problems generated by the interpretation of copies under strict set-theoretic assumptions.

10) The tree in [Figure 4](#) is consistent with [Collins & Stabler’s \(2016, p. 48\)](#) characterisation of intermediate nodes as ‘sets [...] with arcs pointing to their members’, a somewhat mixed metaphor. If Merge generates sets, we do not see where the ‘arcs’ come from. We will argue for a more minimal syntactic representation, which eliminates these intermediate nodes.

In (5), A contains $\{a, c\}$ and is a subset of S , which contains $\{b\}$ and the set $\{A, \{a, c\}\}$; in graph-theoretic terms A is a sub-graph of S iff $V(A) \subset V(S)$ and $E(A) \subset E(S)$. From this perspective, both mathematical frameworks allow us to capture the same relations. Another example: in graph-theoretic terms, we define a *root* of a tree as a node that is not dominated by any other node (Wilson, 1996, p. 56; Zwicky & Isard, 1963) or in terms of the definition of path:

A digraph $G(V, E)$ is said to have a root τ if $\tau \in V$ and every vertex $v \in V$ is reachable from τ ; that is, there is a directed path that starts in τ and ends in v [for all v]. (Even & Even, 2012, p. 37)

The notion of *root* has a set-theoretic analogue:

K is the root iff:

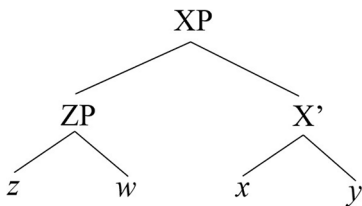
for any Z , Z a term of K , every object that Z is a term of is a term of K . (Epstein et al., 2013, p. 262)

Set-theoretically, the *root* of an SO is the set that is not a proper subset of any other set (in (8), there is no X such that $S \subset X$). In both set and graph theoretic representations, the property of being a ‘root’ is predicated of the same symbol: S .

Minimalism has sometimes equated set-theoretic representations with graph-theoretic ones; for example, Chomsky (1995a, p. 247) offers a direct translation between a tree diagram and its set-theoretic interpretation:

Figure 5

Tree Diagram Taken From Chomsky (1995a, p. 247)



Here [in Figure 5] $ZP = \{z, \{z, w\}\}$, $X' = \{x, \{x, y\}\}$, $XP = \{x, \{ZP, X'\}\}$; more accurately, the tree with ZP as root corresponds to $\{z, \{z, w\}\}$, and so on, the labels of the roots having no status, unlike standard phrase markers.

This alleged correspondence is not unproblematic. In the set-theoretic representation, the status of ZP , X' , and XP is unclear: they seem to be used as proxies for subsets, as informal ways to ‘refer to’ sets (Chomsky says they have ‘no status’, but then it is not

easy to see why they are used at all). As *per* BPS, they are syntactically inert and not part of syntactic representations (Collins, 2017, p. 49; Seely, 2006, p. 189; but see Epstein, 2000, p. 148, for whom $\{a, \{a, b\}\}$, with a the label, is a set with two members: a and $\{a, b\}$). This is a very important difference with a graph-theoretic interpretation of a diagram like Chomsky's: in a graph every node counts, since it is part of its formal definition.

Chomsky's fragment suggests that the choice between sets and graphs as the format of linguistic descriptions is merely notational. We want to emphasise that this is not the case: there are relations and operations that we can define in objects like Figure 4 (graphs) but not in objects like (5) (unordered sets). If these graph-exclusive relations are relevant for the definition of MS, then a set-theoretic syntax runs into trouble given the centrality of MS in the definition of other syntactic relations and operations. For example, in Figure 4 (a graph-theoretic X-bar tree) we can define a *walk* W between nodes b and c :

$$(6) \quad W_{b,c} = \langle e \langle b, S \rangle, e \langle S, A \rangle, e \langle A, c \rangle \rangle$$

In (6) we have an ordered set of edges: we go from b to S , from S to A , and from A to c . Because trees are undirected, we can also define a walk *from* c *to* b , going through the same edges 'in reverse'. It is not possible, however, to define such a walk for a set-theoretic representation like (5) unless a total order is defined over elements of the set, which requires an additional operation on top of Merge¹¹. Set-theory allows us to work in terms of co-membership and containment (which in turn has been used to define the binary relation *term of*; Chomsky, 1995a, p. 247, 2020, p. 22). Taking trees seriously as graph-theoretic objects is a first step, but our approach to structure building will involve more radical departures from classical diagrams. Note, for example, that the trees in Figure 4 maintain intermediate nodes (i.e., intermediate and maximal phrasal nodes. In the BPS version, we have a multi-segment category): we aim to eliminate these, too.

We have emphasised that problems arise when trees are used as diagrams to represent set-theoretic Merge. Suppose that we Merge X and Y (for X and Y arbitrary SOs in the workspace), yielding the set $\{X, Y\}$. This is diagrammed in Minimalism using a binary-branching tree:

11) As a reviewer points out, Collins and Stabler (2016, p. 51) define *path* in terms of recursive immediate containment:

A path is a sequence of syntactic objects $\langle SO_1, SO_2, \dots, SO_n \rangle$ where for every adjacent pair SO_i, SO_{i+1} of objects in the path, $SO_{i+1} \in SO_i$ (i.e., SO_{i+1} is immediately contained in SO_i).

In Figure 4, we can define a path $\langle S, b, A \rangle$ following immediate containment. This means, however, that if two SO are not in a relation of (transitive) containment, a path cannot be defined: such is the case of a and c in Figure 4. Additional structure needs to be invoked to create a containment relation (e.g., via Self-Merge). The graph-theoretic approach requires no such complication: A walk is readily available from c to a and vice-versa.

Figure 6

Unlabelled Binary Branching Tree

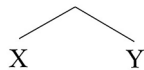


Figure 6 is usually regarded as a graphical representation of the set $\{X, Y\}$. However, from a graph-theoretic perspective we need to consider the fact that Figure 6 is a rooted tree with *three* nodes, not two (as noted in Chomsky, 2020, p. 39) and *two edges* (each edge by definition connecting two nodes). Since the edges converge, it means that there is a node created by the merger (which we will call \bullet , remaining agnostic about the indexed category it is to be assigned to) such that $e_1 \langle \bullet, X \rangle$ and $e_2 \langle \bullet, Y \rangle$. As observed above, this object can be obtained by starting with three symbols and merging \bullet and X and \bullet and Y; how to obtain it simply by merging X and Y, particularly under set-theoretic ‘Simplest Merge’, is unclear. The formal object defined as the set $\{X, Y\}$ and the graph in Figure 6 are thus, crucially, distinct and not mutually translatable.

A more accurate representation of $\text{Merge}(X, Y) = \{X, Y\}$ than Figure 6 would be Figure 7 (McKinney-Bock & Vergnaud, 2014, p. 219ff.):

Figure 7

Undirected Edge



Here, no new nodes are introduced¹², we have just the input of Merge (points X and Y in the workspace) and a non-directed edge connecting the two terms involved in the operation. This is, informally, all we can do with the information contained in the output of Merge as unordered set formation.

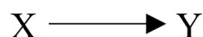
An alternative is to define that Merge is asymmetric, driven by the satisfaction of selectional requirements (e.g., Chomsky, 1995a, p. 246; Stabler, 2011). In this context, if

12) Graphs like that diagrammed in Figure 7 and directed variants of it satisfy Chomsky’s (2021, p. 19) *Minimal Yield* condition more naturally than traditional Minimalist trees, since no new nodes are created when X and Y are merged. The arc is the syntactic relation between X and Y. Furthermore, there is no need to ‘remove’ X and Y from the workspace, since X and Y are unchanged: syntax relates X and Y by means of an arc, which defines a trajectory in the workspace. Minimal Yield becomes a consequence of the format of structure building under graph-theoretic assumptions.

Merge of Y to X satisfies a selectional requirement on X, then $\text{Merge}(X, Y) = e\langle X, Y \rangle$, and we need an arc *from* X *to* Y:

Figure 8

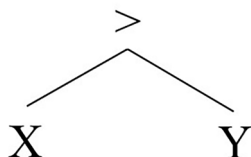
Directed Edge



In Minimalist Grammars, for example, X would have a selector feature $=F$, matched with a categorial feature F in Y. Implemented as in Figure 8, structure building delivers not an unordered set, but a digraph. This notation allows McKinney-Bock and Vergnaud to dispense with the additional node that is used in Minimalist Grammars (e.g. Stabler, 2011, illustrated in Figure 9) since the asymmetry of selection is encoded in the arc itself:

Figure 9

Labelled Binary Branching Tree



Note that in defining Merge graph-theoretically we have not lost any of the classical properties of classical Merge-based syntax: binarity, recursion, and discrete infinity (what Chomsky, 2021, pp. 6–7 calls the *Basic Property of Language*). Nor have we introduced information about linear precedence. Shifting from ‘*form set*’ to ‘*form digraph*’ does not entail giving up on properties that are at the core of Minimalist desiderata for syntactic structure. Figure 8 may now be interpreted as a dependency-like structure capable of representing predicate-argument relations; again, this is less of a departure from Minimalist assumptions than it may seem (Osborne et al., 2011 provide extensive discussion about the relations between BPS and dependency grammars): Structure building delivers an order than feeds MS as well as semantic interpretation. The argument is both theoretical and empirical. For example, let $X = \text{read}$ and $Y = \text{books}$: Merge of *books* to *read* satisfies a selectional restriction in *read*. Encoding this syntactic relation as an arc allows us to capture that ‘*argument structure, is invariably given by external MERGE*’ (Chomsky,

2020, p. 44): under set-theoretic assumptions, as Langendoen observes, the asymmetric nature of argument structure remains unexplained and unrepresented. In our approach, subcategorisation, implemented through digraphs such that (as in (6)) a structural description is an ordered set of arcs, allows the syntax to guide MS in cases that would be set-theoretically ambiguous (Section 7). This is a desirable result. At the same time, we can encode in structure building the asymmetry of selection (and thus provide a link to compositional interpretation) without resorting to designated nonterminal nodes (labels), the elimination of which has long been an aim of Minimalist inquiry (Chomsky, 1995b; Collins, 2002, 2017; Seely, 2006).

Trees and sets are neither equivalent nor notational variants of each other, and the choice between one and the other as the basis for syntactic theory has far-reaching consequences in terms of the relations and operations that can be defined in each. This is important, as one of our objectives is to evaluate the feasibility of MS as a search algorithm defined over sets formed by Merge/MERGE and over graphs. If MS is as fundamental an operation as current Minimalist theorising makes it to be (at the core of labelling, long-distance dependencies, anaphoric binding, and Agree), then we may use MS as part of an argument to decide what the best format for structural descriptions is. And if some of the areas where MS becomes crucial are also shared with other theories of syntactic structure, we take this to be an argument in favour of paying more attention to those domains.

The next section presents the properties of the outputs of the generative operation Merge in the context of recent Minimalist works. We must examine these properties and determine to what extent a search algorithm like the one assumed in the works we have cited here can apply to SO generated by Merge.

5 Merge: What It Is and What It Can Do

It is necessary at this point to characterise the generative operation to evaluate the feasibility of defining a search algorithm for its output. Collins (2017, pp. 50–53) gives a list of the properties of Merge: summarising,

- Merge is iterable
- Merge is binary (the input of Merge is always a pair of objects)
- Merge is commutative ($\text{Merge}(X, Y) = \text{Merge}(Y, X)$)¹³
- (The output of) Merge is unspecified for linear order (Chomsky, 2013, p. 40)

¹³ This property of Merge makes MS algorithms such as Chow's (2022) difficult to implement. Chow's proposed version of MS depends crucially on the system being able to define left and right daughters of a node, such that the left daughter is checked first, followed by the right daughter. Then, the search goes back to the left daughter, and applies recursively. Clearly, in Chomsky-Collins sets there is no such thing as a 'left' or 'right' daughter. Chow's system assumes graph-theoretic trees -as far as we can understand- albeit implicitly.

- (The output of) Merge is unlabeled
- Merge is not triggered (by a head, a feature, etc.)
- Merge is never counter-cyclic
- Merge is all there is structure building-wise: there is no Move or Copy
- Merge *cannot produce* {XP, YP} or {X, Y} objects (where X and Y are heads) (Kayne, 2018, p. 11)
- Merge allows to dispense with traces, indices, and copies
- Merge allows to dispense with the notion of Chain

These properties constitute the background against which the definition and role of MS can be evaluated. The properties of Collins-style Merge that matter for our purposes also hold for the approach in Fukui and Narita (2014), Epstein et al. (2015, 2020), Chomsky (2013, p. 42 et seq.), among others. Chomsky's recent works (Chomsky, 2019, 2020, 2021) propose a somewhat reworked version of the generative operation, called MERGE. MERGE involves unordered set formation plus removal from a workspace (Chomsky, 2021, p. 19 needs to stipulate a condition Minimal Yield as a 'corollary' of the Strong Minimalist Thesis, which does not follow from the set-theoretic foundations of structure building, to 'eliminate [...] recourse to the illegitimate operation Remove'):

- (7) Workspace (WS) contains X and Y: $[_{WS} X, Y]$
 $MERGE(X, Y) = [_{WS} \{X, Y\} X, Y]$ ¹⁴
 Remove X, Y from WS = $[_{WS} \{X, Y\}]$

The basic properties of Merge are still there. The removal of X and Y from the workspace intends to restrict the probing space and the number of elements available for further computations.

This version of Minimalism differs quite substantially from previous stages of the theory in terms of the properties of the outputs of the generative operation. In the first incarnation of Minimalism, the generative operation Merge was defined as follows:

Applied to two objects α and β , Merge forms the new object K, eliminating α and β . (Chomsky, 1995a, p. 243)

Furthermore, because SO are interpreted at the C-I and A-P interfaces differently depending on whether they are verbal, nominal, etc. (Chomsky, 1995a, pp. 245–247), K was defined as a set $\{\gamma, \{\alpha, \beta\}\}$, with γ the label of $\{\alpha, \beta\}$. Chomsky continues:

K must therefore at least (and we assume at most) be of the form $\{\gamma, \{\alpha, \beta\}\}$, where γ identifies the type to which K belongs, indicating its relevant properties. (Chomsky, 1995a, p. 243)

¹⁴ Chomsky (2020, p. 34, 2021, p. 20) formulates the input of the operation as $MERGE(X, Y, WS)$, as if WS was a syntactic object.

Since then, Chomsky and others have separated labelling from structure building (e.g. Boeckx, 2015; Epstein et al., 2015; Hornstein, 2009, and related work), such that under the simplest definition of Merge, the operation only delivers unordered sets, and labelling is taken care of by a different mechanism. Overall, there seems to be nothing in Chomsky's recent papers and talks that constitutes a break with Collins' positions: MERGE(X, Y) = $\{X, Y\}$. Thus, we will refer to the outputs of generative operations, these unordered, unlabelled objects, as 'Chomsky-Collins sets'.

5.1 Searching and Labelling in Chomsky-Collins Sets

Now we have a characterisation of the objects that, in current Minimalist theorising, MS is supposed to apply to: Chomsky-Collins sets. How does it work and how do we know if the search is successful? This latter point has been addressed explicitly: MS looks for a *head* bearing relevant features. If in the domain of the search a suitable head is found, other operations may apply (labelling, Agree, Internal Merge, Form Copy, etc.). We will first focus on labelling, since it has received the most attention in the Minimalist literature: even though MS also underlies Agree (Branan & Erlewine, 2021; Ke, 2019, 2022; Milway, 2021; Preminger, 2019) and successive cyclic movement (Chomsky, 2020, p. 36), they have not received as much attention as labelling by MS. The conclusions, however, extend to all applications of MS.

For purposes of labelling,

LA [Labelling Algorithm] is trivial for $\{H, XP\}$ structures, H a head. In this case, LA selects H and the usual operations apply (Chomsky, 2015, p. 7; also 2013, p. 43)

For things to work as Chomsky and others have suggested, it must be possible for LA, given an SO, to determine whether there is a head: this means that *head* must be definable in set-theoretic terms. Given MERGE(X, Y), we need to know if X or Y is a head. Suppose that we have

(8) $\{read, \{the, book\}\}$

as the output of MERGE($read, \{the, book\}$). We must determine whether we want to consider *read* as a singleton (i.e., is *read* actually $\{read\}$?¹⁵) This is important because if lexical terminals are singletons, then determining whether something is a head or not requires some additional operation apart from MERGE: for instance, we may evaluate the cardinality of a set built by MERGE. A head would be a set of cardinality 1, a

15) As observed by a reviewer, *read* may have internal structure: $\{v, \sqrt{read}\}$ (v a categoriser). In this case, in Chomsky's quotation below we need to consider the root and the categoriser as two distinct elements in the workspace (Chomsky, 2021, p. 36). See also fn. 30.

non-head would be of cardinality greater than 1. Alternatively, the labelling algorithm must distinguish sets of features (lexical items; Collins & Stabler, 2016, p. 44) from sets of sets of features (i.e., phrases): this latter criterion is explicitly adopted in Ke (2022). It is important to determine, then, the relation between SO, workspaces, and sets. Chomsky (2020, p. 37) says:

We want to say that $[X]$, the workspace which is a set containing X is distinct from X .

$[X] \neq X$

We don't want to identify a singleton set with its member. If we did, the workspace itself would be accessible to MERGE. However, in the case of the elements produced by MERGE, we want to say the opposite.

$\{X\} = X$

We want to identify singleton sets with their members

This approach is somewhat confusing¹⁶. Let us explore some of its consequences. If MERGE involves an operation of removal or replacement such that a workspace containing a and b , to which MERGE applies, contains the set $\{a, b\}$ with a and b removed, then determining what is a head and what is not precisely requires the aforementioned kind of cardinality-sensitive procedure: this process is independent of (and of course distinct from) MS. Thus, labelling of unordered sets cannot be reduced to MS. We can make this more concrete. Suppose the workspace WS contains, as before, the singleton $\{read\}$ and the set $\{some, books\}$:

(9) WS: $[\{read\}, \{some, books\}]$

Then, MERGE replaces this pair of objects with a set containing them:

(10) $MERGE(\{read\}, \{some, books\}) = WS': [\{\{read\}, \{some, books\}\}, \{read\}, \{some, books\}]$

Under Chomskyan assumptions, the sets $\{read\}$ and $\{some, books\}$ need to be removed from WS' . In this context, determining whether $\{read\}$ is a head involves establishing its cardinality, which in turn requires defining an injection or bijection to the other set in the workspace, $\{some, books\}$ (as Kunen, 2013, p. 62 observes, talking about the cardinality of a set such as $\{read\}$ in isolation has no meaning). Collins (2017, p. 56) provides a definition compatible with our characterisation¹⁷:

16) And may lead to inconsistencies. For example, $\{\emptyset\}$ is a subset of every set, but \emptyset is not an element of every set.

$A \in \{A, B\}$ is final in SO iff there is no C contained in (or equal to) SO such that $A \in C$, and C contains $\{A, B\}$. Otherwise, A is non-final in SO.

Which is exactly what we sketched above: following Chomsky's (2020) approach (that is: if we want to identify a lexical item with a singleton), in defining 'head' it is necessary to determine if a set contains subsets (other than itself).

If, however, we do *not* want to identify a lexical item with a singleton (against Chomsky, 2020), then when we have $\{read, \{some, books\}\}$ the system may simply find the object that is *not* a set: given Merge(X, Y), establish which of those is not a set. If lexical items are not sets, the system may simply look for something that is not a set, and thus it is a head. How MS, as a search algorithm, would accomplish that is unclear. This does not mean the difference between heads and phrases cannot be captured (see, e.g., Collins & Stabler, 2016, pp. 45–46), but that such differentiation is independent of MS. Suppose for a moment that the only difference between XP and X was that the former is a (proxy for a) set, but the latter is not (as the logical alternative to the scenario sketched in the previous paragraph). Then, if H is a head, (i) Merge(H, XP) merges a non-set and a set; (ii) Merge(XP, YP) merges two sets; and (iii) Merge(H, H) merges two non-sets (the outputs of these operations are always sets, however). This means that Merge should be formulated in such a way that it is specified that its output is always a set, but its input need not be: it can be a pair of non-sets, a pair of sets, or a set and a non-set. Interestingly, from this perspective, the configurations that have been dubbed problematic for MS (e.g., in Chomsky, 2013, 2015; Epstein et al., 2015; van Gelderen, 2021; Kitahara, 2020, among others) are the ones that involve either two sets ($\{XP, YP\}$) or two non-sets ($\{X, Y\}$); the proposed 'solutions' for the symmetry problem are—again—independent of MS.

A further issue that impacts on the implementation of MS in labelling is that the identification of heads seems to be completely independent from subcategorisation. This is surprising if External Merge delivers argument structure, as claimed by Chomsky. The precise nature of the relation between structure building and predicate-argument relations in set-theoretic syntax is unclear. Thus, in determining a label for the output of Merge($\{read\}, \{books\}$) so that the interfaces may determine the relevant semantic and phonological properties of the derived object (Chomsky, 1995a, p. 243), it is not sufficient to determine cardinality, since both sets are unary (or they are both non-sets). It is also not sufficient to stipulate that MS can distinguish sets of syntactic terms from sets of features (i.e.: sets of sets of features and sets of features): in this case we would have two sets of features. Two options are possible if we wanted to maintain the set-theoretic foundations of Minimalism:

17) For purposes of Agree by MS, this characterisation is challenged by Preminger (2019, p. 24), who defines that a search criterion must *not* incorporate relational information (such as 'x is/is not dominated by a projection of the same head').

- (i) transform a relation between two singletons into a relation between a singleton and a set with a greater cardinality by adding more structure, as in Self-Merge proposals (thus multiplying elements in representations)
- (ii) remove one of the objects after additional structure has been introduced, as in labelling-driven movement proposals (multiplying steps in derivations)

Both involve complications and departures from Minimalist desiderata (Chomsky 1995a, p. 151).

An alternative is readily available: to define the output of Merge(*read*, *books*), where *read* and *books* are points in the workspace, as a digraph *e*<*read*, *books*> (see Figure 10):

Figure 10
Directed Edge Between Basic Expressions



Note first that the digraph is not a new object independent of the nodes *read* and *books*, in contrast to Chomsky’s (2019) claim that the output of MERGE is a workspace that contains both *read* and *books* as well as the set {*read*, *books*}. Removal can therefore be dispensed with. In our view, structure building delivers a set of digraphs, whose nodes are basic expressions indexed by a set of uniquely identifying addresses (along the lines of Gorn, 1967a, 1967b, and leveraged for linguistic purposes in Sarkar & Joshi, 1997; Karttunen & Kay, 1985 and much related work). For any expression E, [E] is the uniquely identifying address of E (Krivochen, forthcoming, Chapter 2). Furthermore, the content of an address is a semantic value: for any expression E, [E] is the semantic value of E (Heim & Kratzer, 1998). The reader may think of this analogously to how a computer file can be accessed from different shortcut icons, which may be located in different folders. Insofar as all shortcuts for a file are instructions to access the same memory location regardless of their context, the shortcuts correspond to our ‘addresses’. Continuing the analogy, the content of the file is its ‘semantic value’. Strictly speaking, then, we should have Figure 11:

Figure 11
Directed Edge Between Addresses



Having an address indexing system allows us to refer to expressions unambiguously, regardless of context (i.e., what they are immediately dominated by or immediately dominate): wherever in a structure that we find the address $\llbracket \text{books} \rrbracket$, the same content will be accessed. If addresses are uniquely identifying, the computation can remain ‘dumb’ (Chomsky, 2021, p. 16), and *Stability* (Chomsky, 2019, p. 275, 2021, p. 16) follows: in a derivation, the same address may be called and the same semantic value accessed more than once (as we will see in Section 7), but there is no need to stipulate that each call is a distinct ‘occurrence’ of a single symbol (Chomsky, 2021) or a ‘discontinuous object’ (Chomsky et al., 2019, p. 237).

In our view, Merge delivers asymmetric relations between expressions: the order thereby imposed is not linear precedence, but determined by subcategorisation. We want to capture the fact that *books* is an argument of *read*. To implement such a system we could, for example, assume the following featural compositions for *read* and *books* (based on Stabler, 2011):

- (11) *read*:: =D =D +case V
books:: D -case

=F is a selector feature, which requires Merge with a category F, +F is a licenser (or ‘probe’) feature, and -F is a licensee (or ‘goal’) feature. In this case, *books* satisfies the selectional requirement =D. The arc *e*-*read*, *books*> plays two major roles: (i) it partially satisfies the selectional requirements of *read* (there being a =D feature still unsatisfied) and (ii) it delivers an ordered object where search may apply unambiguously without the need for additional structure¹⁸. The core idea is that we want of a successful syntactic theory to represent argument structure and modification, both empirically motivated relations.

The final difficulty that we will consider, and which underlies not only MS for purposes of labelling but also Internal Merge, is precisely that search algorithms are defined for structured data: there must be an order imposed over the data that search algorithms go through. If a search algorithm can be defined such that paths can be compared and shortest paths can be chosen (Chomsky, 2020, p. 36, 2021, p. 18; Epstein et al., 2020; Fitzpatrick, 2002, p. 450; Hayashi, 2021, p. 22ff.; Kitahara, 2020)¹⁹, it is because

18) There is a potential redundancy between selector-categorical features and arcs, since they both represent selection. Given the choice, arcs should stay, since they also provide a way to define search sequences for the satisfaction of nonlocal features (e.g., [wh], see Section 7).

19) Incidentally, many of these proposals involve computing at least two paths (sets of terms) and comparing their cardinality, choosing the shortest path (Fitzpatrick, 2002, p. 450 makes this explicit). But once more than one path is computed, there is no independent reason to choose the one with the smallest cardinality over any other; certainly not one of computational efficiency since at least two paths were already computed to begin with (see Jacobson, 1998 for a version of this argument related to transderivational conditions favouring ‘shortest derivations’ in early Minimalism).

the data that constitutes the probing space for that algorithm is ordered (and paths can be stored, which brings about issues of memory that we will not analyse here). This order is precisely what we want for structural descriptions of natural language sentences.

From a set-theoretic perspective, *co-membership* and *containment* are the only relations created by MERGE (Internal or External). We can, then, sketch the following derivation (taken from Chomsky, 2020, p. 48):

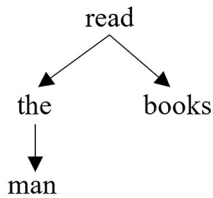
- (12) a. {b}
 b. {a, b}
 c. {b₁, {a, b₂}}

Here, b₁ is Internally Merged to the set {a, b}. According to Chomsky (2020), MS renders b₂ inaccessible to further operations because b₁ is found first. This entails, however, that the system knows (i) that it is looking for an object of category b, and (ii) that b₁ and b₂ are copies of each other (or ‘occurrences’ of b; see Chomsky et al., 2019, p. 237; Collins & Stabler, 2016, p. 51), neither of which follow from either Merge or MS. Here, an order based on containment seems sufficient: there is a set that contains b₂ and excludes b₁, but there is no set that contains b₁ and excludes b₂ (so (12c) must be a multiset; see Gärtner, 2022 for discussion). As mentioned above, Chomsky (2019, p. 275, 2021, p. 16) must introduce a new principle, *Stability*, which determines that we are dealing with ‘occurrences’ of a single object b. However, considering only containment will not work more generally: if we order SO in terms of proper containment, phrasal sister nodes (sets in a relation of co-containment) cannot be ordered, since neither is a subset of the other (this is known as a *symmetry point*; see Kitahara, 2020; Uriagereka, 2002):

- (13) a. {reads, books}
 b. {{the, man}, {reads, books}}

Graph-theoretically, the dependency between the lexical predicate and its arguments can be minimally represented as arcs, as can dependencies between members of each SO (leaving aside for the time being *v*, *T*, and *C*)²⁰:

²⁰ We assume in (14) and Figure 12 a DP-like structure, where the noun is selected by a D head. An alternative, explored in Krivochen (forthcoming), is to assume (i) that only categorematic expressions are assigned addresses and thus only these correspond to nodes in structural descriptions, and (ii) that determiners are syncategorematic (what amounts to an NP analysis). In that case, the ordered set of arcs would be $G' = \langle e\langle \text{read}, \text{man} \rangle, e\langle \text{read}, \text{books} \rangle \rangle$. In contrast to McKinney-Bock and Vergnaud (2014), we do not incorporate intermediate nodes in the graph-theoretic representations. These authors also annotate edges with the kind of syntactic relation that holds between nodes (e.g., Checking or Selection); we are focused on Selection since operations over features may occur at a long distance (as explored in Branagan & Erlewine, 2021; Preminger, 2019). Thus, Checking would hold for walks, not single edges (however, Gärtner, 2002, p. 85 argues that there is no long-distance checking; in this case, both Checking and Selection would be strictly local operations).

Figure 12*Directed Graph Analysis of “the man reads books”*

- (14) a. $e<\text{read}, \text{books}>$
 b. $e<\text{the}, \text{man}>$
 c. $e<\text{read}, \text{the}>$

Neither (13b) nor (14) (diagrammed in Figure 12) feature intermediate nodes (e.g., VP), but only (14) encodes selection. If External Merge delivers argument structure, digraphs are better suited than unordered sets to encode syntactic dependencies. As in (6), we also need to impose an order over edges (but now the edges themselves are directed): a structural description is then *an ordered set of arcs* (Krivochen, forthcoming, Chapter 5). For (14), we have (15):

- (15) $G = \langle e<\text{read}, \text{the}>, e<\text{the}, \text{man}>, e<\text{read}, \text{books}> \rangle$

The order between edges can be defined in several ways: one possibility is to have the order represent order of composition. In that case, (15) defines a top-down parse (Sobin, 2020; Zwart, 2009, 2015, and much related work)²¹. Other options are available, within the limits of the formalism and depending on restrictions imposed by the theory²². A digraph

21) Alternatively, as explored in Krivochen (forthcoming, Chapter 5), the order may be dictated by the Grammatical Function hierarchy (Bell, 1983; Dalrymple et al., 2019; Keenan & Comrie, 1977; Postal, 2010, among many others); a similar approach is adopted in HPSG, in that elements in the ARG-ST list of a predicate are ordered following the GF hierarchy (Müller, 2015). In this case (admittedly, a departure from Minimalist assumptions) the order of dominance relations corresponds to the order of grammatical relations. Suppose that x is a node corresponding to a predicative basic expression which takes two arguments, y and z . Minimally, if Merge satisfies selectional properties of predicates, we need two arcs: $e<x, y>$ and $e<x, z>$. Now, if we define a graph as an *ordered* set of arcs, we can define a correspondence between orders such that if an arc $e<x, y>$ is ordered before $e<x, z>$ in the description of a graph, the grammatical relation between x and y is also ordered before that between x and z in the GF hierarchy, starting from the highest ranked GF. In other words: the two orders are isomorphic. In Figure 12, for example (and using Relational Grammar terms) the constituent with head *the* is the arc 1 of *read*, since 1 is the highest grammatical function in the hierarchy, and the constituent with head *books*, ordered after the constituent with head *the*, is the arc 2 of the same predicate (the next grammatical function in the hierarchy).

22) See Shieber (1988), Pollard (1997) for discussion about the distinction between formalism and theory relevant to this point.

is, then, much more informative than an unordered set for purposes of MS as well as semantic composition.

Set-theoretic Merge only defines two relations: containment and co-containment. If (co-)containment alone does not provide enough information to define a sequential search, can we appeal to some other mechanism? In the original, asymmetric version of Merge, the order in the output is given by labelling, such that $\{X, Y\}$ is either $\{X, \{X, Y\}\}$ or $\{Y, \{X, Y\}\}$ ($\langle X, Y \rangle$ or $\langle Y, X \rangle$ under the Wiener-Kuratowski definition). However, if order is given by labelling, then MS cannot be a pre-condition for LA (since order is itself a pre-condition for MS), nor can MS be the labelling algorithm itself (since MS underpins other operations, such as Agree). If Merge/MERGE creates Chomsky-Collins sets, and a search algorithm is to be defined over those sets, they must be ordered (and MS itself cannot impose that order).

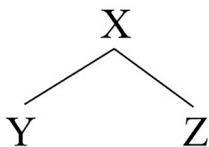
Assuming that labelling is driven by MS results in a problem: before labelling, the result of Merge is an unordered array and MS should apply either randomly (for sequential search) or in parallel. We will come back to labelling on Section 6; before doing that, we need to revise some basic assumptions about the (a)symmetry of Merge.

5.2 Symmetric and Asymmetric Structure Building

Suppose we have a tree as in Figure 13,

Figure 13

Labelled Binary Branching Tree



where Y and Z may be internally complex. We will first explore the case where Y is a head and Z a non-head. Y is an expression assigned to an indexed category in the grammar, and so is Z. What is the indexed category assigned to an expression of the form $[Y Z]$ (which will determine the kind of syntactic rules that can affect that object, the rules of semantic interpretation that will apply, its distributional properties, etc.)? For example, if $Y = v$ and $Z = VP$, then $X = vP$, because

minimal search finds v as the label of SO since v is unambiguously identifiable (Epstein et al., 2015, p. 203)

An object embedded in VP cannot provide a label because

in any {H, XP}, the head H is always found with “less search” than any feature bearing (hence relevant information-bearing) element within XP. (Op. Cit.)

The idea is intuitive. It also imposes requirements over specific configurations: for example, an implementation of this idea requires us to assume that *carefully* in *read a paper carefully* cannot be introduced in the derivation as a head, but rather (i) as a full phrase, or (ii) by means of some additional operation (e.g., Pair-Merge), or (iii) at a derivational point after labelling has taken place (e.g. Stepanov, 2001; Zyman, 2022). This is so because if the structure *{read {a paper}}* is a VP, and we introduce an adverb as in *Merge({carefully}, {read, {a, paper}})*, the labelling algorithm would incorrectly label the resulting SO as an AdvP. To avoid these situations, Chomsky (2020, pp. 48–49) proposes that structure building comes in two flavours: symmetric and asymmetric:

In symmetrical MERGE, if you happen to have a head and an XP, then the head will provide the label – in earlier versions, what projects. But that’s a case of MS [...].

Asymmetric structure is exemplified with adjectival modification (such that, under any version of immediate constituency analysis, *young man* is an NP, not an AP), but the argument is equally valid for D-N structures, V-NP, etc. This distinction seems to refer to Set-Merge vs. Pair-Merge, but if so, the new terminology is unexplained. Part of the problem is that, since selection plays no role in structure building and the category of SOs is claimed to be relevant only at the interfaces, the argument vs. adjunct distinction must be encoded in some other way (e.g., derivational timing under Late Adjunction)²³. It is also unclear what is *symmetrical* about {H, XP}: in *{read, {some, books}}*, *read* subcategorises for an NP object, not the other way around; this imposes an (asymmetric, total) order over the set. How this is ‘a case of MS’ is not explained. Recall that the original definition of Merge argues for the *asymmetry* of the operation:

23) In Multiple Spell-Out (MSO), a complex specifier or an adjunct is derived in parallel, and once introduced in the main structure its internal dynamics are inaccessible. In the ‘radical’ version of MSO (Uriagereka, 2012, p. 91) material derived in a parallel workspace cannot provide a label since they are not part of the SO at the time of labelling (note that we do not need to multiply occurrences of adjuncts if a proper notion of derivational timing is assumed, compare with the proposal in Nakashima, 2022). Uriagereka summarises ‘radical’ MSO as follows:

A more radical way to spell out a current [a SO that can be expressed in a finite-state fashion without information loss; Op. cit.: 86] is assuming it does not even merge with the rest of the structure. This is radical because it treats separate currents as separate main currents—in that case living quite separate derivational fates.

Under present assumptions, selection in combination with derivational timing delivers the results we want: first, adverbs do not satisfy the selectional requirements of the verb *read*; combining *carefully* with *read books* results in an expression of the same category of *read books* (whereas combining *books* with *read* changes the category from V to VP; or from TV to IV in categorial terms), see Dowty (2003). Second, the adjunct is introduced ‘late’ in the derivation (certainly after the satisfaction of selectional requirements, possibly counter-cyclically; Frank, 2013, p. 231ff.).

The operation $\text{Merge}(\alpha, \beta)$ is asymmetric, projecting either α or β , the head of the object that projects becoming the label of the complex formed. If α projects, we can refer to it as the target of the operation (Chomsky, 1995a, p. 246, 1995b, p. 63)

Labelling in BPS was encoded as part of Merge itself, with some later developments requiring Merge to be triggered by featural requirements (e.g., [Stabler, 2011](#); [Sobin, 2014](#); [Wurmbrand, 2014](#)): in these cases, Merge satisfies some selector feature on a Probe, which projects after Merge. As argued above, a graph-theoretic output meets these requirements with no further stipulations.

In any case, the situation first explored by Chomsky (namely, $\{H, XP\}$) seems straightforward: if an SO contains a head and a non-head, the labelling algorithm, which works by MS, finds that head and labels the SO. However, formalising that is not a trivial task. The first question is how the search would take place; in other words, how to define each step that makes up the algorithm. Above, when defining a search algorithm, we specified that it needs some way to compare inputs with the target of the search (the case of MS applied to Agree, where probes search for matched features that need valuation, is considered in [Ke, 2019](#), p. 44; [Preminger, 2019](#), p. 24, and [Branan & Erlewine, 2021](#)); we can adapt the simplest sequential search algorithm presented in [Knuth \(1998, p. 396\)](#) to the present context:

- (16) Given a sequence of syntactic objects SO_1, SO_2, \dots, SO_n , the algorithm searches for a head.

Step 1: Initialise. Set $i \leftarrow 1$

Step 2: Compare. If SO_i is a head, terminate.

Step 3: Advance. Increase i by 1, proceed to Step 2.

Step 4: End of input – terminate.

A problem with the application of this algorithm to Minimalist syntactic structures (in addition to the issue, noted above, that if Merge yields unordered sets it is not possible to arrange terms in a unique sequence) is that the system must somehow know how to determine, given an SO, whether it is a head or not: an object cannot be identified as a head by the search procedure (as suggested by a reviewer) because the algorithm needs to know what it is looking for in advance (e.g., [Ke's 2019](#), p. 44, 2022, p. 5 *Search Target* or [Preminger's 2019](#) and [Chow's 2022](#) featural requirements, specified as part of the input for MS). In set-theoretic syntax this requires the system to (a) evaluate the cardinality of a set (which involves defining a mapping between sets) or (b) distinguish between sets of features and sets of sets of features. This is hardly something that 'blind' Merge could do. Indeed, computational implementations of Minimalism do away with 'blind' Merge and unordered sets²⁴. In our digraph alternative, in contrast, order is a property of the

structural description to which MS applies, and independently motivated by argument structure. Defining a search sequence is unproblematic.

6 Labelling the Unlabelled

Let us preface this section by pointing out that the ‘problem of labelling’ arises only if (i) the derivation proceeds bottom-up, step-by-step *and* (ii) the output of the generative operation is unordered. If we think of grammars as Post rewriting systems (what Pullum, 2020 calls ‘expansion-based grammars’), for instance, the ‘labelling’ problem does not exist: in a structure like Figure 13, X would be introduced in the derivation one turn before Y and Z, and this introduction is precisely calling the indexed category to which X belongs. Within ‘composition-oriented grammars’, labelling is also not a problem in top-down Minimalist models of syntactic derivations (e.g., Sobin, 2020 and references therein; Zwart, 2009, 2015). Under MERGE, in contrast, it is sometimes necessary to label an object *after* that object has been created and more structure introduced, counter-cyclically. Furthermore, it has been proposed that SOs may remain label-less (Citko, 2008; Collins, 2002; Hornstein, 2009). The idea of label-less objects pushes Minimalism even farther away from immediate-constituent analyses and the class of expansion-based formal grammars that Chomsky studied in the ‘50s, and the empirical advantages of pursuing such an approach are still unclear, in terms of providing descriptively adequate treatments of empirical phenomena that were unaccounted for in previous models.

Consider now what a label-less object commits us to, for purposes of MS. In set-theoretic terms, all objects are either sets or members of sets (notation like $\{_{\alpha} X, Y\}$ has no formal status in set theory). Thus, if we have $\text{Merge}(X, Y) = \{Z, \{X, Y\}\}$, with Z the ‘label’ of $\{X, Y\}$ (Chomsky, 1995b, p. 62), we have two options. One is to consider Z a member of the set $\{Z, \{X, Y\}\}$ (e.g., Epstein, 2000, p. 148; Hornstein & Pietroski, 2009). The other is to use Z as an informal proxy for the set $\{X, Y\}$ (e.g., Seely, 2006, p. 189). If a label is just a proxy, it is unclear why labelling should matter at all. If a label is an element

24) Minimalist Grammars (e.g., Fong & Ginsburg, 2019; Stabler, 2011) assume a feature-rich system with a designated set of labels and projection indicated at every point in the tree. There is, to the best of our knowledge, no implementation of free Merge and MS-based labelling in Minimalist Grammars. Ginsburg (2016), inspired heavily in Chomsky (2013, 2015) finds himself forced to give up ‘free Merge’:

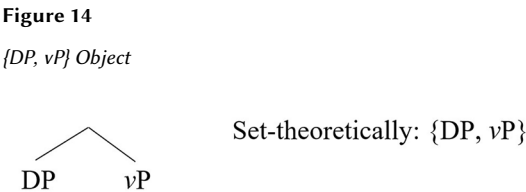
This type of free merge would create a huge computational burden because our model would have to compute an enormous number of unsuccessful derivations in order to arrive at a successful derivation. While free merge may have its merits [...] it is not clear to us how to implement free merge in a computationally efficient manner. (Op. cit., fn. 14).

Even when a root is ‘unlabelled’, it may enter further computations because the computational system is fed a single lexical item at a time; this presupposes the kind of Select operations that Chomsky et al. (2019, p. 245) explicitly consider unnecessary.

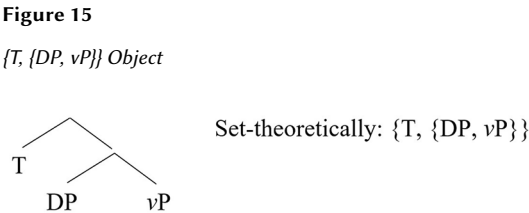
of a set, the counter-cyclicity issue identified below becomes more relevant, particularly given conditions on the Markovian properties of derivations (Chomsky, 2021, p. 20).

In graph-theoretic terms, in contrast, what we would call a ‘label’ is the root of a graph, at each derivational step. In other words: the label of a SO (a set of nodes and edges) is the node that is the root of the local graph which defines that SO. Thus, if an operation takes that SO as part of its input, the structural description of that operation will refer to the root of the SO, not to every node properly contained in it. In set-theoretic terms, there must be a way to refer to sets in a more abstract way; a ‘variable over sets’ as it were, such that we can formulate structure mapping rules without mentioning specific sets.

In what pertains to MS, it is unclear how approaches with *unlabelled* objects make this work²⁵. We can give an example (adapted from Epstein et al., 2015, p. 203; Ott, 2015; van Gelderen, 2021, p. 14ff.): suppose we have Figure 14 created by Merge (trees are diagrammatic only):



Neither object is a head (both vP and DP are presumably proxies for sets), so LA cannot apply. The object in Figure 14 merges with T, yielding Figure 15:



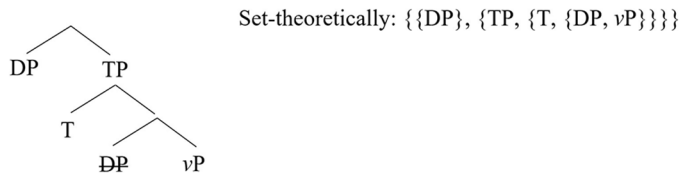
25) In some Minimalist works (e.g., Chomsky, 2013; Ott, 2015), *label* is taken to be equivalent to *head*, which is a sense of *label* very different from the use in formal language theory (e.g., Hopcroft & Ullman, 1969, p. 19) and early generative grammar. In these works, an ‘unlabelled’ object is an object where no head can be ‘detected by Minimal Search’ (Ott, 2015, p. 157).

At this point, we need to refer to the set extensionally (since there is no root address). Here we can clearly distinguish between a graph-theoretic and a set-theoretic approach: set-theoretically, there is no root accessible to the Narrow Syntax in Figure 15 since there is no label (Epstein et al., 2013, p. 254); graph-theoretically, there is a root, because there are two edges that converge in a node (even if there is no indexed category assigned to that node). What is the next step? Epstein et al. (2015), building on Chomsky (2013), say²⁶

*Suppose $SO = \{XP, YP\}$ [in our case, $\{DP, vP\}$], neither a head. Here **minimal search is ambiguous**, locating the heads X, Y of XP, YP , respectively. There are, then, two ways in which SO can be labelled: (A) modify SO so that there is only one visible head, or (B) X and Y are identical in a relevant respect, providing the same label, which can be taken as the label of the SO . (Epstein et al. 2015, p. 202. Our highlighting)*

As in Moro (2000), a symmetry point is broken via movement. In Figure 15 the DP must move in order to break this configuration, thus leaving the vP to be projected as a label counter-cyclically (henceforth, we leave aside the problem of whether Figure 16 should be formalised as a multiset):

Figure 16
Structure After DP Raising



This process involves backtracking: the object $\{DP, vP\}$ must remain somehow active and accessible, requiring a label, and the system must be able to modify a SO by adding a new element to a previously formed set: $\{T, \{DP, vP\}\}$ becomes $\{T, \{vP, \{DP, vP\}\}\}$ with DP ‘ignored’ for purposes of MS due to being a copy. The ‘Markovian’ character of derivations emphasised in Chomsky (2021) seems to be violated. Importantly, this kind

26) Collins (2017, p. 53) presents a very similar picture of Chomsky’s LA:

- If $SO = \{XP, YP\}$ and neither is a head, then*
- a. if XP is a lower copy, $Label(SO) = Label(YP)$.*
- b. if $Label(XP)$ and $Label(YP)$ share a feature F by Agree, $Label(SO) = \langle F, F \rangle$.*

See also van Gelderen (2021, p. 13ff.), Ott (2015, p. 173ff.), Chomsky et al. (2019, p. 248).

of operation required by the system in Chomsky (2013, 2015), as noted in Ginsburg (2016), is inherently counter-cyclic, with Ginsburg's proposal (which restricts probing to a root node) being '*an improvement over the manner in which feature checking occurs in [Chomsky, 2013, 2015]*' (Op. Cit.). The fact that the system allows for (or even *requires*) unlabelled objects (which Chomsky et al. 2019, p. 248 equate to *exocentric* structures, for unclear reasons²⁷), and counter-cyclic operations to label them is a source of formal difficulties that did not exist in previous stages of Minimalist theorising and which does not seem to be required to provide empirical analyses that would be otherwise impossible.

In the literature, the fact that DP movement in structures like Figure 15 (and similar cases, such as structures of the form {copula {XP, YP}}) is required for labelling reasons is usually considered an advantage, insofar as it allows for the elimination of the EPP as a motivation for subject raising. However, it is unclear whether (i) this mechanism of symmetry breaking via movement to yield a label-able object is consistent with the characterisation of labelling as following from MS and (ii) the supposed simplification of the generative procedure entailed by unordered set formation outweighs the problems it generates. In what follows we will focus on whether {XP, YP} and {X, Y} situations are indeed as problematic for MS as theoretical Minimalist work suggests²⁸. If they are not, then the application of further movement operations to 'dissolve' the symmetry point would be unwarranted, and independent justification must be found.

7 Graphs and MS: The 'Ambiguity' of {XP, YP} and {X, Y}

Above we introduced the concept of search algorithms for trees, in two variants: breadth-first and depth-first. We also reviewed the difficulties of defining a search for unordered sets. However, what if Minimalist trees were taken as more than just graphical aids, notational variants of sets? The alternative, as emphasised throughout this paper, is to consider them not sets, but graphs (Arikawa, 2019; Gärtner, 2002, 2014; Kayne, 1984; McCawley, 1968, 1982b, for a variety of perspectives). How does a search apply in this case? There are two options. We may assume that the search is *not* triggered by a head, as in the case of labelling (presumably required to satisfy interface legibility conditions).

²⁷ Exocentricity seems to us to be clearly distinct from the lack of a label. A phrase structure tree defined by the phrase structure rule $S \rightarrow NP VP$ is exocentric, but labelled.

²⁸ We leave aside, for reasons of space, a discussion of the procedure that labels the root of the structure in Figure 16 as $\langle \varphi, \varphi \rangle$ (Chomsky, 2013, p. 43). If $\langle \varphi, \varphi \rangle$ is intended as an ordered set (as suggested by the notation), a problem that it is not straightforward to formally characterise a labelling operation that can yield either a unary set (e.g., {VP}) or an ordered pair other than by stipulation. There is also the question of what the contribution of $\langle \varphi, \varphi \rangle$ is to semantic interpretation and how constituency can be read off that label (which is a pair of features, or a proxy for one).

In this case, the ‘search’ should start from the root of the local structure being probed: Labelling involves finding the least embedded head within the SO that constitutes the probing space for MS²⁹. If, on the other hand, we assume that the search is triggered by a head (as seems to be the case for Agree; see Branan & Erlewine, 2021; Ke, 2019; Preminger, 2019) not much changes for the formalisation of search: The starting point will no longer be the root of the local structure (which by definition is not a head). Following the Minimalist literature on Agree, we can characterise MS as a procedure that defines the unique sequence from a probe P to a goal G. Importantly, in this context, a search algorithm seems to be useful in the characterisation of phenomena that arise independently of set-theory based syntax: filler-gap dependencies, anaphoric reference, agreement, etc.

Following Minimalist proposals, once a head is found the search terminates. This can be generalised: when the algorithm finds an object in the input that matches its target, the search halts (Chomsky, 2021, p. 18; Preminger, 2019, p. 24). This brings up the additional problem (noted above) of having the system know what it is looking for *a priori*. But suppose that part of the input for MS is a specification of the target, as much of the literature assumes. The problem of heads still arises: as Preminger observes, under BPS assumptions there cannot be a featural difference between a ‘maximal projection’ and a ‘head’, since non-terminal levels of projection ‘(...) are now viewed as additional instances of the very same syntactic object that constitutes the head’ (Preminger, 2019, p. 23). Still, we have not provided a way for the algorithm to determine what a ‘head’ is: we did point out some problems that arise in a strict set-theoretical approach to syntax and pointed out that featural specifications will not do, but, can we provide a characterisation of the notion of ‘head’ in graph-theoretic terms that requires no additional stipulations?

7.1 No XP-YP Ambiguity in Graph-Theoretic Trees

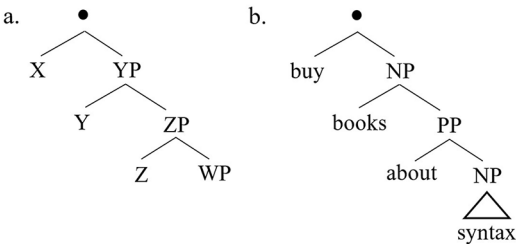
In Section 4 we introduced the notions of *indegree* and *outdegree* of a vertex. In this context, interpreting Minimalist trees graph-theoretically, we can provide the following definitions:

- The *root* of a graph is a node with indegree 0 and outdegree non-zero
- A *head* is a node with indegree non-zero and outdegree 0
- Intermediate nodes are nodes with indegree non-zero and outdegree non-zero

29) More than one cyclic model of syntactic computation can implement MS in these terms, delivering different results. A Multiple Spell-Out model (Uriagereka, 2002, 2012) restricts the probing space to the *command unit* that contains the probe (a SO generated via monotonic Merge). A Lexicalised Tree Adjoining Grammar restricts it to a single elementary tree, which are units of argument structure (within which all thematic roles selected by the anchor of the elementary tree are assigned; see Frank, 2002, p. 55, 2013). A phase-based system requires additional assumptions, since even within a monotonically derived syntactic object there may be a phase head establishing a boundary.

These definitions are weakly equivalent to saying that a leaf in a tree is a head, and require nothing other than the tools already available to us to characterise the format of syntactic structures as graphs and the concept of tree traversals (Kremers, 2009; Kural, 2005). MS, in this context, can be characterised as a search algorithm that, applying in a tree T , defines a sequence Σ of alternating nodes and edges which specifies a unique walk from either the root (for purposes of labelling) or a designated internal node (for purposes of Agree, the designated node being a Probe) to a node with indegree 1 and outdegree 0 (and a suitable featural composition, in the case of Agree). Let us see an example. Given a classical labelled phrase structure tree (interpreted graph-theoretically) such as Figure 17

Figure 17
Phrase Structure Trees

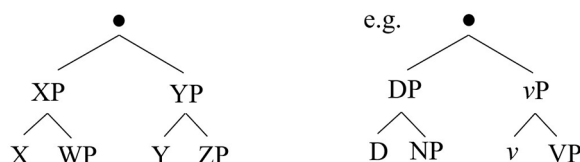


the search sequence from the root until the first node with outdegree 0 would be $\Sigma = \langle \bullet, X \rangle$ (i.e., $\Sigma = \langle \bullet, \text{buy} \rangle$), both under depth-first and breadth-first algorithms, assuming a preorder. By Chomsky’s LA, this identifies \bullet as XP (VP in (b)).

The first question to address is whether objects of the type $\{XP, YP\}$ are indeed ambiguous for MS. To this end, we need to examine how the search algorithms defined before would work. Suppose that XP and YP both contain a head and a phrase:

- (17) $XP = \{X, WP\}$ (e.g., $\{D, NP\}$)
 $YP = \{Y, ZP\}$ (e.g., $\{v, VP\}$)

And we Merge XP and YP to create $\{XP, YP\}$, represented in tree form in Figure 18:

Figure 18*Tree With a Symmetry Point*

Suppose that the label of the new node created by this merger is to be determined by MS. The theory in Chomsky (2013, 2015) and related works requires that (i) the new node created by merging DP and v P be left unlabelled, (ii) with the introduction of a subsequent head—namely, T—DP is internally Merged at the root, leaving behind a copy, and (iii) the object in Figure 18 is counter-cyclically labelled v P because the DP copy would be ‘invisible’ for the labelling algorithm. It seems clear that this reasoning does not hold under the present view of how MS is defined, as a sequential search algorithm: both the head D of DP and v of v P are ‘accessible’ (in the sense that they can be found by the algorithm), but one of them will necessarily be found first. If the search sequence is defined from left to right, the head that will be found is D. If defined from right to left, it will be v (see also Chow, 2022). But moving DP to get a configuration where only v can be found is a stipulation that is independent from the formal definition of MS, since there is nothing wrong with the configuration for MS purposes; furthermore, the additional auxiliary hypothesis that copies are ‘invisible’ for labelling is needed.

Consider now what happens if the object in Figure 18 is interpreted as a graph-theoretic tree. A traversal through the tree would contain the sequence $\langle \bullet, \text{XP}, \text{WP} \rangle$ (assuming a preorder), but not $\langle \text{WP}, \text{XP}, \bullet \rangle$. In this situation, a depth-first search over the structure just described would follow the sequence Σ below until finding the first node with outdegree 0 (a ‘head’), where search halts. The depth-first sequence would be (18):

$$(18) \quad \Sigma = \langle \bullet, \text{XP}, \text{X} \rangle$$

And a sequential breadth-first search would follow Σ' :

$$(19) \quad \Sigma' = \langle \bullet, \text{XP}, \text{YP}, \text{X} \rangle$$

In neither case is there an ambiguity: when the algorithm finds an object that matches the target (here, a head X), the search stops. An advantage of the present approach is that it is not necessary to compare the distance between the root and X vs. the root and Y (cf. e.g. Hayashi, 2021; Kitahara, 2020); if MS cannot do anything other than probe structure

sequentially and compare each node visited with a target value, no ambiguity arises. We consider this to be a desirable scenario.

In Figure 18 above both XP and YP are *accessible* for the search algorithm. However, either XP or YP will be probed into first in a sequential search (for example, see Kural, 2005, pp. 373–376 for an explicit implementation of linearisation-as-traversal that also finds no issue with ‘symmetry points’; contra Collins, 2017, pp. 58–59). Crucially, no hierarchical information is ignored under sequential depth-first search: hierarchy is at the core of the graph-theoretic approach by virtue of dependencies being asymmetric.

Finally, if a situation where two non-terminals are in a relation of sisterhood was ambiguous for MS, and there was no external controller the search procedure would just halt with no output, regardless of the trigger of the search. But it doesn’t, in formal characterisations of sequential search algorithms. If MS doesn’t just halt, then we must assume that there is either a bias in the algorithm that directs the branch that is to be looked at first or there is some external factor that dictates which branch is to be probed first. We may define that the search follows a *preorder*, *inorder*, or *postorder* traversal, and make sure that the left or right branches are always, consistently, searched first (e.g. Chow, 2022). The ‘ambiguity’ of {XP, YP} situations is an artefact of set-theoretic commitments, not a necessary formal property of structural descriptions or of MS.

So far as we can see, essentially the same argument holds for {X, Y} situations. However, some additional considerations must be made. An {X, Y} situation emerges, according to Chomsky (2013, 2015), in two cases: (i) when a root and a categoriser Merge and (ii) in instances of head movement. Both involve stipulations that do not impact on the definition of MS (e.g., category-less roots, head movement being post-syntactic or not, etc.). Presumably, there could be other cases, depending on how much silent structure is allowed. For example, the bolded sequences in (20a–b) could also involve two terminal expressions, depending on the set-theoretic status of *her* and *books*³⁰:

- (20) a. I **love her**
 b. John **reads books**

Under BPS assumptions, *love* and *her* should be two heads. If this is correct, then mono-transitive verbs with pronominal or bare N objects would generate {X, Y} situations, unless additional structure for the object is proposed in the form of functional nodes. There is no difficulty for MS, for reasons explained above. Labelling-wise, a system where it is

30) It may be worth noting that in the Minimalist literature cited in this paper the notation for a head is always H, not {H}. That is: a configuration that contains a head and a complex object is notated {H, XP} (where XP is possibly a proxy for a set), and not {{H}, XP}. This suggests that indeed the sequence *love her* receives an analysis {love, her} and not {love, {her}} or {{love}, {her}}. Self-Merge (see Guimarães, 2000. Lasnik & Uriagereka, 2022, p. 104ff. restrict Self-Merge to nouns), which could deliver an initial asymmetry and add structure by artificially creating a set, is also banned in Epstein (2000): assuming Self-Merge alongside the derivational definition of c-command entails having c-command as a reflexive relation.

possible to say that the monotransitive verb selects an object, and the pronominal object partly satisfies the valency of the verb seems preferable to a system where the syntax is independent of such considerations. In this case, selection should play a role in labelling, if constituency is to be represented: *love her* behaves like an expression that can concatenate with an NP to yield a finite clause (however that is encoded). Graph-theoretically, instead of (21a) (which gives no information about argument structure or hierarchy) we have the arc in (21b):

- (21) a. {love, her}
b. $e\langle\text{love, her}\rangle$

Only the arc (21b) allows us to define an ordered sequence without further stipulations or intermediate nodes, while at the same time representing predicate-argument relations. The treatment of $\{X, Y\}$ situations in the framework of Chomsky (2013, 2015) seems to require more theory-specific auxiliary hypotheses than $\{XP, YP\}$, and since they do not pertain to the definition of MS, we will not analyse them in this paper.

7.2 Leveraging the Addressing System: Filler-Gap Dependencies

In addition to Agree and labelling, MS has also been invoked in the analyses of long-distance relations. For example, the interpretation of a filler-gap dependency may be construed as a search³¹. These structures raise additional difficulties (noted in the Minimalist literature) pertaining to the distinction between copies and repetitions: simply treating binary branching Minimalist trees as graph-theoretic trees is not enough to solve these. We will see how having the set of nodes in our graphs indexed by a set of addresses, under $\text{Merge}(X, Y) = e\langle X, Y\rangle$, makes a difference.

Consider the following intermediate structural description of a *wh*-interrogative:

- (22) $[C \text{ } [[\text{Mary}] \text{ } [T \text{ } [[\text{Mary}] \text{ } [v \text{ } [\text{read } [\text{what}]]]]]]]]?$

If *C* searches for a SO to Internally Merge at the root, it is necessary to specify that in addition to being an NP, the target of the search must bear a(n unchecked) *wh*-feature; otherwise, *Mary* would be a suitable target³². Similarly, for reconstruction purposes, the

31) This is not just the case in transformational analyses. For example, LFG's treatment of long-distance dependencies, using functional uncertainty, defines a dependency path between filler and gap as a regular expression (Dalrymple et al., 2019, p. 207ff.). This allows the grammar to determine the grammatical function of a constituent when it cannot be determined by reference to local subcategorisation frames. The solution of functional uncertainty requires a search, and functional uncertainty defines the path.

32) If an adequately restrictive meta-theory of features is devised (as of yet a missing part of the Minimalist theory; see Panagiotidis, 2021 for discussion), such a possibility is not unreasonable. Indeed, Preminger (2019) formalises an approach to sequential MS where only featural (not relational) information matters when determining whether the

what that is Internally Merged in Spec-CP needs to find a copy of itself which has not checked its uninterpretable *wh* feature in a chain $CH = \langle \textit{what}_{wh}, \textit{what}_{wh} \rangle$.

What to do after a target has been found is a different matter from the search itself: In Minimalist terms, the target is in some sense ‘copied’ and re-Merged at the root: we obtain a SO with two ‘tokens’ of *what* which need to be somehow linked. Chomsky (2021, p. 17) formulates an operation Form Copy which ‘*assigns the relation Copy to certain identical inscriptions*’, and elaborates on the distinction between ‘copies’ and ‘repetitions’ (see also Collins & Groat, 2018). The conditions under which *identity* is determined are unclear, particularly if syntactic operations are triggered by featural requirements (see also Epstein & Seely, 2002; Gärtner, 2002, pp. 87ff., 96 for related discussion).

How can a graph-theoretic definition of Merge help us solve these problems? We showed above that in binary-branching graph-theoretic trees, search ambiguities do not arise, but even then complications related to copies and repetitions remain. Simply replacing tree diagrams with graph-theoretic trees does not suffice. Recall that our proposal departs from trees as the format of structural descriptions: in our analysis, the syntactic workspace is a lattice of uniquely indexed basic expressions (Krivochen, 2022) and structure building delivers digraphs in which predicates always dominate their arguments. This allows us to identify the category of the root of the structure corresponding to *John run* without adding new nodes: Merge of *John* and *run* delivers a digraph where *run* immediately dominates *John*: $G = e \langle \textit{run}, \textit{John} \rangle$. Omitting some functional structure for now, Merge of T with $e \langle \textit{run}, \textit{John} \rangle$ delivers $G' = \langle e \langle T, \textit{run} \rangle, e \langle \textit{run}, \textit{John} \rangle \rangle$, with root T. So far, so good, but we have not yet needed to resort to copies.

The analysis of filler-gap dependencies allows us to emphasise the usefulness of the unique indexing system in simplifying structural descriptions. For example, following standard Minimalist analyses, in

(23) What did Mary read *what*?

both *what* are distinct nodes, assigned the relation Copy given some condition of ‘identity’ and thus forming a chain. Under present assumptions, however, each node in a graph corresponds to an expression indexed by an address, which points to the semantic value of that expression. In (23), then, we are dealing with two calls for the same semantic value: the same address is called twice in a graph. In practical terms, this means that in checking the *wh*-feature in C, a new edge may be created between the root C and the only suitable goal: *what* becomes multidominated. This, building on an observation by Gärtner (2002, p. 96), is the simplest way to enforce identity in chains: no copy or trace is created to begin with. Under the Copy theory, where the *what* remain distinct

SO being evaluated is a suitable target (cf. Featural Relativised Minimality as in Villata et al., 2016; however, here relational information is relevant).

nodes, a search sequence can be defined for purposes of reconstruction (assuming a graph-theoretical reading of Minimalist trees), but the copy/repetition problem appears. By assuming an addressing system, chains and copies are effectively eliminated, and *Stability* is satisfied.

The derivation of (23), under present assumptions, involves the following steps (for concreteness, we follow McKinney-Bock and Vergnaud (2014) in assuming that two relations may correspond to an arc: *selection* or *checking*. We will focus on the *wh*-dependency, glossing over the relation between T and the external argument as simply an instance of checking. Also for concreteness, as in Stabler, 2011, +F is a probe feature, and -F is a goal feature):

- (24) a. Merge(read, what) = $e\langle \text{read}, \text{what}_{-wh} \rangle$ ----- **Selection**
 b. Merge(v , read) = $\langle e\langle v, \text{read} \rangle, e\langle \text{read}, \text{what}_{-wh} \rangle \rangle$
 c. Merge(v , Mary) = $\langle e\langle v, \text{Mary} \rangle, e\langle v, \text{read} \rangle, e\langle \text{read}, \text{what}_{-wh} \rangle \rangle$ ³³
 d. Merge(T, v) = $\langle e\langle T, v \rangle, \langle e\langle v, \text{Mary} \rangle, e\langle v, \text{read} \rangle, e\langle \text{read}, \text{what}_{-wh} \rangle \rangle \rangle$
 e. Merge(Mary, T) = $\langle e\langle T, \text{Mary} \rangle, e\langle T, v \rangle, e\langle v, \text{Mary} \rangle, e\langle v, \text{read} \rangle, e\langle \text{read}, \text{what}_{-wh} \rangle \rangle$ ----- **Checking**
 f. Merge(C_{wh} , T) = $\langle e\langle C_{wh}, T \rangle, e\langle T, \text{Mary} \rangle, e\langle T, v \rangle, e\langle v, \text{Mary} \rangle, e\langle v, \text{read} \rangle, e\langle \text{read}, \text{what}_{-wh} \rangle \rangle$ ³⁴

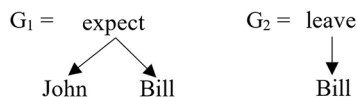
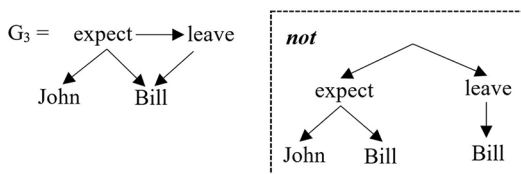
At this point, we can define a search sequence triggered by C_{+wh} , where C as a probe (or its +wh feature) searches for a suitable goal to check its *wh*-feature:

- (25) $\Sigma = \langle C_{+wh}, T, v, \text{read}, \text{what}_{-wh} \rangle$

If expressions are assigned uniquely identifying addresses, there is no need (and indeed no way) to ‘copy’ *what* in a derivation, if the same semantic value is called in all cases. Rather, a new edge can be created from C to *what*, delivering a cyclic digraph:

33) At this point, having saturated the valency of the predicate, we can put the order between arcs in correspondence, e.g., with the grammatical function hierarchy (see fn. 21), and read off grammatical functions from the ordered set of arcs: *Mary* as a 1 (Subject) of *read* and *what* as its 2 (Object). Note that the establishment of a relation between C and *what* by means of a new arc does not disrupt the dependency between the verb and its object, as *what* never leaves its position as a tail of the arc headed by *read*.

34) Note that *Mary* is also multidominated (by *read*, of which it is an argument, and by T, with which it checks features): excluding multidominance under graph theoretic assumptions would require additional stipulations. Given a uniquely identifying addressing system for nodes, no ‘copies vs. repetitions’ problem arises (cf. Chomsky et al., 2019; Collins & Groat, 2018).

Figure 19*Elementary Graphs***Figure 20***Derived Graphs*

Contrary to set-theoretic Minimalism, we do not start with one ‘occurrence’ of *Bill* and create copies of them as the derivation proceeds: the indexed basic expression *Bill* is a point in the workspace, and cannot be ‘copied’, although its semantic value can be accessed more than once. If each subgraph is a syntactic unit where subcategorisation and thematic properties of predicates are satisfied (again, building on lexicalised Tree Adjoining Grammars), that expression is a node in every graph that contains a predicate that subcategorises for it (in our case, *expect* and *leave*).

If each expression is uniquely indexed (by the set of addresses), nodes corresponding to expressions with distinct indices (i.e., distinct addresses and thus distinct semantic values) will *not* be collapsed into one node, by the definition of structure sharing: they are distinct points in the workspace. What Chomsky calls ‘repetitions’ corresponds to a situation in which two or more expressions happen to share PF exponents, and nothing else. No further assumptions are required to assign the correct, disjoint reference reading to *Bill expects Bill to leave* (where both *Bill* are distinct nodes with distinct addresses and distinct semantic values, say, $\llbracket \text{Bill Clinton} \rrbracket$ and $\llbracket \text{Bill Bailey} \rrbracket$). A similar argument can be made for *John, John met yesterday* (Chomsky, 2021, pp. 27–28): If there has been no unification, the topicalised *John* and the subject *John* must be distinct nodes in the graph (for example, in this case, with semantic values $\llbracket \text{John Bonham} \rrbracket$ and $\llbracket \text{John McLaughlin} \rrbracket$).

8 Conclusions

The aim of this paper was to explore MS as a search algorithm and its interaction with structure building operations. We contended that the set-theoretic commitments of Minimalism conspire against a definition of sequential search algorithms over the output of Merge³⁵. We thus propose to replace

$$(28) \quad \text{Merge}(X, Y) = \{X, Y\}$$

with

$$(29) \quad \text{Merge}(X, Y) = e\langle X, Y \rangle$$

where X takes Y as an argument (for X and Y uniquely indexed expressions in the workspace). In this view, structural descriptions are ordered sets of arcs. The historically peripheral role that graph theory has played in the development of generative grammar with respect to operations over strings and sets (despite early work such as Bach, 1964; McCawley, 1968; Zwicky & Isard, 1963) could be reversed as a result of pursuing lines of inquiry related to MS. In a graph-theoretic context, MS can be defined as either a breadth-first or a depth-first search. If we do that, the ‘problems’ posed by $\{XP, YP\}$ and $\{X, Y\}$ to MS must be critically revisited: if a well-defined MS renders these situations unambiguous, then it is not MS that should be reformulated (the opposite conclusion is reached in Ke, 2022; Milway, 2021). Our aim is not to devise a MS procedure that delivers the ambiguities of set-theoretic syntax, but rather to eliminate those ambiguities and the configurations where they arise, if possible and empirically sound³⁶. If graph-theoretic Merge is adopted, the phenomena that are supposedly characterised in terms of set-theoretic ambiguities and their resolution need to be defined in different terms. Incidentally, the same argument form is used in current Minimalism against so-called ‘extensions of Merge’ such as Multidominance or Sideways Movement (Chomsky, 2019).

Interestingly, BPS trees can be mapped to irreducible graphs without intermediate symbols by means of an operation made available by the formalism: edge contraction. In graph theory, *edge contraction* removes an edge from a graph, and collapses its head and tail into a single node. Then, given the tree in Figure 21, we can apply edge contraction

35) Branan and Erlewine (2021) do not specify to what extent their proposal is compatible with Chomsky-Collins sets. They assume to be dealing with graphs, although ‘*the underlying representation could itself be set-theoretic, with processes such as labeling and some process determining “order” [...] applying when necessary to give us the necessary information to treat the representation as a graph*’ (Erlewine, p.c.). Given Chomsky-Collins sets, an MS-independent mechanism needs to be invoked to generate an object where a search algorithm may apply.

36) The MS procedure argued for here has the same properties as structure building: asymmetry, recursion, and sequentiality. So far as we can see, Ke’s arguments from parallel search in vision do not obviously apply to a syntactic algorithm in stepwise structure building.

and obtain the graph in Figure 22, introducing the additional requirement that an edge may be contracted only under address identity of the nodes it joins (contracted edges are dashed):

Figure 21

BPS Tree

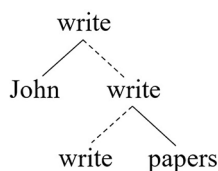
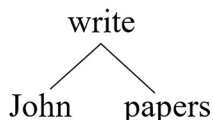


Figure 22

Graph Obtained From BPS Tree via Edge Contraction



The mapping from Figure 21 to Figure 22 identifies edges whose head and tail are identically indexed (remember that the set of nodes is indexed by the set of addresses), removes these edges, and collapses the nodes delivering a single *write*. To do this, however, BPS structures need to be interpreted as graph-theoretic to begin with. Also, the conversion is unidirectional: BPS can be mapped to irreducible graphs but not the other way around, since vertex splitting (Eades & de Mendonça, 1995), applied recursively, could in principle produce an unlimited number of intermediate nodes (unless restricted by conditions that would ultimately result in a set of restrictions over phrase structure along the lines of X-bar theory, and/or a condition such as LFG's *off-line parsability*, Dalrymple et al., 2019, p. 245ff.).

Finally, we want to briefly mention some issues that in our opinion set the agenda for future research. A crucial one pertains to *sequential* vs. *parallel* search. The choice depends to a great extent on what the format for structural descriptions is, and what they are designed to capture. Preminger (2019), Branen and Erlewine (2021) and us prefer sequential search, all for different reasons. Ke (2019, 2022), in contrast, argues for a non-modular parallel search on the grounds that it best implements Chomsky's LA (by capturing the ambiguities of set-theoretic representations in {XP, YP} objects) and an argument from search in vision (Ke, 2022, p. 6ff.; fn. 17) which depends on MS being a 'third factor' operation. Ke stores searched sets as vectors, with sets searched in parallel

being stored in the same vector, so no ‘counter’ is needed. Goto (2019) also seems to implement parallel search but, unlike Ke, he does not explicitly discuss the matter. A detailed comparison of the empirical consequences of both (applied to trees as well as to our more minimal graphs) strikes us as a necessary next step.

Another salient question is whether there is a way of determining empirically if MS is best modelled as a *depth-first* or a *breadth-first* algorithm. We agree with Branan and Erlewine (2021, fn. 11) that

it is productive to consider options for the search procedure in the most general case, without reference to such information [relational information about nodes in a structure], and to then consider the shape and size of the search space separately

Again, the literature has explored both options. Ke (2019, p. 44ff., 2022) argues in favour of a parallel breadth-first interpretation of MS on several grounds, including that (i) depth-first, allegedly, cannot capture some aspects of hierarchical structure by ignoring c-command relations in the definition of a traversal, and (ii) the results of Merge are unordered sets, and breadth-first is more consistent with symmetric relations between set members (which we argue are an artefact). These objections to depth-first depend on a particular interpretation of what the objects manipulated by the syntax are: Chomsky-Collins sets or digraphs. If structure building yields local digraphs, Ke’s objections do not hold (since there are no unordered sets of nodes at any point). However, this does not mean that breadth-first search has no place in a graph-theoretic grammar. Consider, for instance, the approach to non-configurationality in Hale (1983), Austin and Bresnan (1996), and related works. If non-configurationality implies a radical reorganisation of phrase structure, such that syntactic representations in languages like Warlpiri are ‘flat’ (as proposed in modular theories, where the representation of argumental relations over-rides a morphological component, whereas the syntactic component can only generate *n*-ary branching phrase markers), a depth-first search may not be optimal: all targets may be at the same depth. A parallel breadth-first search may be preferable. In a phrase structure system that is more rigidly organised (‘configurational’), depth-first seems to be more convenient (Kremers, 2009). In this sense, Branan and Erlewine (2021, fn. 11) observe that Preminger’s (2019) definition ‘does not invite a clear classification as *depth-first* or *breadth-first*’: it is not impossible to conceive of ‘mixed’ systems, applied to tree structures. Choosing between depth-first and breadth-first MS may be a source of cross-linguistic variation.

Finally, we want to emphasise that the graph-theoretic view of structure building proposed here has important advantages over the set-theoretic one. Considering Chomsky’s (2019, 2020, 2021) conditions over optimal structure building, graph-theoretic Merge satisfies:

(i) **Minimal Yield:** by virtue of adding no new elements to the workspace. In this view, Merge creates *motherhood without sisterhood* (cf. Epstein, 2000, p. 147ff.). Furthermore, in contrast to what happens under set-theoretic assumptions, no ‘removal’ operation needs to be assumed as concomitant to Merge, since the workspace never contains $e\langle X, Y \rangle$ plus X and Y .

(ii) **Determinacy:** because graphs are directed, Determinacy is satisfied in the definition of search sequences (since for any Probe-Goal pair there is a unique walk from Probe to Goal). Sequential MS can always apply unambiguously, since an (independently motivated) order is defined over the set of syntactic objects in a derivation.

(iii) **Stability:** the fact that expressions are indexed by a set of uniquely identifying addresses satisfies Stability. A single address may be called multiple times in a derivation, either locally (as in the analysis of reflexives in Krivochen, forthcoming, Chapter 6, or filler-gap dependencies that do not span more than a single lexicalised structure, e.g. (24)) or across local units, as a consequence of graph union (as in (27) and Figure 20).

(iv) **‘Markovian’ derivations:** Finally, while there is no ‘Markovian’ Form Copy (Chomsky, 2021, p. 28), walks are deterministic, and the node to be visited next, at any point, is determined exclusively by the current node and the order imposed over the set of arcs (binary branching in the diagram of a graph is misleading: it obscures the fact that the set of arcs is ordered): MS implemented in digraphs requires no separate memory stack.

Funding: The author has no funding to report.

Acknowledgments: I would like to thank Kleanthes Grohmann for his patience and editorial work, and two anonymous *Biolinguistics* reviewers for insightful and constructive comments which greatly helped me improve both content and presentation. I have also benefitted from discussion with many colleagues, including (in alphabetical order) Mostafa Barzegar, Kenyon Branan, Michael Yoshitaka Erlewine, Hans-Martin Gärtner, Elly van Gelderen, David Medeiros, Andrea Padovan, and Doug Saddy. The usual disclaimers apply.

Competing Interests: The author has declared that no competing interests exist.

References

- Arikawa, K. (2019). Graph theory teaches us something about grammaticality. *Prague Bulletin of Mathematical Linguistics*, 112, 55–82. <https://doi.org/10.2478/pralin-2019-0002>
- Austin, P., & Bresnan, J. (1996). Non-configurationality in Australian aboriginal languages. *Natural Language and Linguistic Theory*, 14, 215–268. <https://doi.org/10.1007/BF00133684>
- Bach, E. (1964). *An introduction to Transformational Grammars*. Holt, Rinehart & Winston.
- Bauke, L., & Blümel, A. (2017). Introduction. In L. Bauke & A. Blümel (Eds.), *Labels and roots* (pp. 1–16). De Gruyter. <https://doi.org/10.1515/9781501502118-001>

- Bell, S. (1983). Advancements and ascensions in Cebuano. In D. Perlmutter (Ed.), *Studies in Relational Grammar 1* (pp. 143–218). University of Chicago Press.
- Boeckx, C. (2015). *Elementary syntactic structures*. Cambridge University Press.
- Branan, K., & Erlewine, M. Y. (2021). Locality and (minimal) search. To appear in K. Grohmann & E. Leivada (Eds.), *Cambridge handbook of Minimalism*. <https://ling.auf.net/lingbuzz/005791>
- Chomsky, N. (1995a). *The minimalist program*. MIT Press.
- Chomsky, N. (1995b). Bare phrase structure. In H. Campos & P. Kempchinsky (Eds.), *Evolution and revolution in linguistic theory* (pp. 51–109). Georgetown University Press.
- Chomsky, N. (2000). Minimalist inquiries: The framework. In R. Martin, D. Michaels & J. Uriagereka (Eds.), *Step by step – Essays in Minimalist syntax in honor of Howard Lasnik* (pp. 89–155). MIT Press.
- Chomsky, N. (2001). Derivation by phase. In M. Kenstowicz (Ed.), *Ken Hale: A life in language* (pp. 1–52). MIT Press.
- Chomsky, N. (2004). Beyond explanatory adequacy. In A. Belletti (Ed.), *Structures and beyond* (pp. 104–131). Oxford University Press.
- Chomsky, N. (2013). Problems of projection. *Lingua*, 130, 33–49. <https://doi.org/10.1016/j.lingua.2012.12.003>
- Chomsky, N. (2015). Problems of projection: extensions. In E. Di Domenico, C. Hamann & S. Matteini (Eds.), *Structures, strategies and beyond: Studies in honour of Adriana Belletti* (pp. 1–16). John Benjamins. <https://doi.org/10.1075/la.223.01cho>
- Chomsky, N. (2019). Some puzzling foundational issues: The Reading program. *Catalan Journal of Linguistics*, 2019, 263–285. <https://doi.org/10.5565/rev/catjl.287>
- Chomsky, N. (2020). *The UCLA lectures* [Unpublished manuscript]. <https://ling.auf.net/lingbuzz/005485>
- Chomsky, N. (2021). Minimalism: Where we are now, and where we can hope to go. *Gengo Kenkyu*, 160, 1–41. https://doi.org/10.11435/gengo.160.0_1
- Chomsky, N., Gallego, Á. G., & Ott, D. (2019). Generative grammar and the faculty of language: Insights, questions, and challenges. *Catalan Journal of Linguistics*, 2019, 229–261. <https://doi.org/10.5565/rev/catjl.288>
- Chow, K. J. (2022). A novel algorithm for minimal search. *Snippets*, 42, 3–5. <https://doi.org/10.7358/snip-2021-042-chow>
- Citko, B. (2008). Missing labels. *Lingua*, 118(7), 907–944. <https://doi.org/10.1016/j.lingua.2008.01.001>
- Collins, C. (2002). Eliminating labels. In S. Epstein & T. D. Seely (Eds.), *Derivation and explanation in the Minimalist Program* (pp. 42–64). Blackwell.
- Collins, C. (2017). Merge(X, Y) = {X, Y}. In L. Bauke & A. Blümel (Eds.), *Labels and roots* (pp. 47–68). De Gruyter. <https://doi.org/10.1515/9781501502118-003>
- Collins, C., & Groat, E. (2018). *Distinguishing copies and repetitions* [Unpublished manuscript]. <http://ling.auf.net/lingbuzz/003809>
- Collins, C., & Stabler, E. (2016). A formalization of Minimalist syntax. *Syntax*, 19(1), 43–78. <https://doi.org/10.1111/synt.12117>

- Cormen, T. C. L., Rivest, R., & Stein, C. (2001). *Introduction to algorithms*. MIT Press.
- Dalrymple, M. L. M., Lowe, J., & Mycock, L. (2019). *The Oxford reference guide to Lexical Functional Grammar*. Oxford University Press.
- Dipert, R. (1982). Set-theoretical representations of ordered pairs and their adequacy for the logic of relations. *Canadian Journal of Philosophy*, 12(2), 353–374.
<https://doi.org/10.1080/00455091.1982.10715803>
- Dowty, D. (2003). The dual analysis of adjuncts/complements in Categorical Grammar. In E. Lang, C. Maienborn & C. Fabricius-Hansen (Eds.), *Modifying adjuncts* (pp. 33–66). De Gruyter.
<https://doi.org/10.1515/9783110894646.33>
- Eades, P., & de Mendonça, C. F. X. (1995). Vertex splitting and tension-free layout. In F. J. Brandenburg (Ed.), *Graph drawing* (pp. 202–211). Springer. <https://doi.org/10.1007/BFb0021804>
- Epstein, S. (2000). Un-principled syntax and the derivation of syntactic relations. In S. Epstein (Ed.), *Essays in syntactic theory* (pp. 141–162). Routledge.
- Epstein, S., Kitahara, H., & Seely, T. D. (2013). Structure building that can't be! In M. Uribe-Etxebarria & V. Valmala (Eds.), *Ways of structure building* (pp. 253–270). Oxford University Press.
- Epstein, S., Kitahara, H., & Seely, T. D. (2015). Labeling by Minimal Search: Implications for successive-cyclic A-movement and the conception of the postulate “phase”. In S. Epstein, H. Kitahara & T. D. Seely (Eds.), *Explorations in maximizing syntactic minimization* (pp. 201–221). Routledge.
- Epstein, S., Kitahara, H., & Seely, T. D. (2020). Unifying labeling under Minimal Search in “single-” and “multiple-specifier” configurations. *Coyote Papers: Working Papers in Linguistics*, 22, 1–11.
<http://hdl.handle.net/10150/641481>
- Epstein, S., & Seely, T. D. (2002). Rule application as cycles in a level-free syntax. In S. Epstein & T. D. Seely (Eds.), *Derivation and explanation in the Minimalist Program* (pp. 65–89). Wiley-Blackwell. <https://doi.org/10.1002/9780470755662.ch4>
- Even, S., & Even, G. (2012). *Graph algorithms*. Cambridge University Press.
- Fitzpatrick, J. (2002). On Minimalist approaches to the locality of movement. *Linguistic Inquiry*, 33(3), 443–463. <https://doi.org/10.1162/002438902760168563>
- Fong, S., & Ginsburg, J. (2019). Towards a Minimalist machine. In R. Berwick & E. Stabler (Eds.), *Minimalist parsing* (pp. 16–38). Oxford University Press.
<https://doi.org/10.1093/oso/9780198795087.003.0002>
- Frank, R. (2002). *Phrase structure composition and syntactic dependencies*. MIT Press.
- Frank, R. (2013). Tree adjoining grammar. In M. den Dikken (Ed.), *The Cambridge handbook of Generative Syntax* (pp. 226–261). Cambridge University Press.
- Fukui, N., & Narita, H. (2014). Merge, labelling, and projection. In A. Carnie, Y. Sato & D. Siddiqi (Eds.), *The Routledge handbook of syntax* (pp. 3–23). Routledge.
- Gärtner, H.-M. (2002). *Generalized transformations and beyond*. Akademie Verlag.
- Gärtner, H.-M. (2014). Strange loops: Phrase-linking grammar meets Kaynean pronominalization. *Linguistische Berichte*, 24, 1–13. <https://core.ac.uk/download/pdf/42927114.pdf>

- Gärtner, H.-M. (2022). Copies from “standard set theory”? A note on the foundations of Minimalist syntax in reaction to Chomsky, Gallego and Ott (2019). *Journal of Logic Language and Information*, 31, 129–135. <https://doi.org/10.1007/s10849-021-09342-x>
- Ginsburg, J. (2016). Modelling of problems of projection: A non-countercyclic approach. *Glossa (London)*, 1(1), Article 7. <https://doi.org/10.5334/gjgl.22>
- Gorn, S. (1967a). Handling the growth by definition of mechanical languages. *Proceedings of the April 18-20, 1967, spring joint computer conference* (pp. 213–224). Association for Computing Machinery. <https://doi.org/10.1145/1465482.1465513>
- Gorn, S. (1967b). Explicit definitions and linguistic dominoes. In J. Hart & S. Takasu (Eds.), *Systems and computer science* (pp. 77–115). University of Toronto Press. <https://doi.org/10.3138/9781487592769-008>
- Goto, N. (2019). *When and how does search take place?* [Unpublished manuscript]. Toyo University. <https://ling.auf.net/lingbuzz/005000>
- Guimarães, M. (2000). In defense of vacuous projections in Bare Phrase Structure. *University of Maryland Working Papers in Linguistics*, 9, 90–115. <https://ling.auf.net/lingbuzz/000463>
- Hale, K. (1983). Warlpiri and the grammar of non-configurational languages. *Natural Language and Linguistic Theory*, 1, 5–47. <https://doi.org/10.1007/BF00210374>
- Hayashi, N. (2021). *Labels at the interfaces: On the notions and the consequences of Merge and Contain* [Doctoral dissertation]. Kyushu University.
- Heim, I., & Kratzer, A. (1998). *Semantics in generative grammar*. Blackwell.
- Higginbotham, J. (1983). A note on phrase markers. *Revue Québécoise de Linguistique*, 13(1), 147–166. <https://doi.org/10.7202/602508ar>
- Hopcroft, J., & Ullman, J. (1969). *Formal languages and their relation to automata*. Addison-Wesley.
- Hornstein, N. (2009). *A theory of syntax: Minimal operations and Universal Grammar*. CUP.
- Hornstein, N., & Pietroski, P. (2009). Basic operations: Minimal syntax-semantics. *Catalan Journal of Linguistics*, 8, 113–139. <https://raco.cat/index.php/CatalanJournal/article/view/168907>.
- Huck, G. (1984). *Discontinuity and word order in Categorical Grammar* [Doctoral dissertation]. University of Chicago.
- Jacobson, P. (1998). Where, if anywhere, is transderivationality located? In P. Culicover & L. McNally (Eds.), *Syntax and semantics 29: The limits of syntax* (pp. 303–336). Academic Press. <https://brill.com/previewpdf/book/edcoll/9789004373167/BP000012.xml>
- Johnson, D., & Postal, P. (1980). *Arc pair grammar*. Princeton University Press.
- Kahane, S., & Lareau, F. (2016). Word ordering as a graph-rewriting process. In A. Foret, G. Morrill, R. Muskens, R. Osswald, & S. Pogodalla (Eds.), *Formal grammar* (pp. 216–239). Springer. <https://shs.hal.science/halshs-01740481>
- Karttunen, L., & Kay, M. (1985). Structure sharing with binary trees. *Proceedings of the 23rd annual meeting of the association for computational linguistics* (pp. 133–136). University of Chicago. <https://doi.org/10.3115/981210.981226>
- Kayne, R. (1984). *Connectedness and binary branching*. Foris.

- Kayne, R. (2018). *The place of linear order in the language faculty* [Talk handout].
<https://as.nyu.edu/content/dam/nyu-as/linguistics/documents/Kayne%200118%20Venice%20The%20Place%20of%20Linear%20Order%20in%20the%20Language%20Faculty.pdf>
- Ke, A. (2019). *The syntax, semantics, and processing of agreement and binding grammatical illusions* [Doctoral dissertation] University of Michigan.
- Ke, A. (2022). Can Agree and Labeling be reduced to Minimal Search? *Linguistic Inquiry*. Advance online publication. https://doi.org/10.1162/ling_a_00481
- Keenan, E., & Comrie, B. (1977). Noun phrase accessibility and universal grammar. *Linguistic Inquiry*, 8(1), 63–99. <https://www.jstor.org/stable/4177973>
- Kitahara, H. (2020). ‘Multiple specifier’ configurations revisited. *Reports of the Keio Institute for Cultural and Linguistic Studies*, 51, 207–216.
https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/download.php/AN00069467-00000051-0207.pdf?file_id=147222
- Knuth, D. (1998). *The art of computer programming. Vol. 3: Sorting and searching*. Addison-Wesley.
- Komachi, M. H. K., Uchibori, A., & Takita, K. (2019). Generative procedure revisited. *Reports of the Keio Institute of Cultural and Linguistic Studies*, 50, 269–283.
https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/download.php/AN00069467-00000050-0269.pdf?file_id=142996
- Kremers, J. (2009). Recursive linearization. *Linguistic Review*, 26(1), 135–166.
<https://doi.org/10.1515/tlir.2009.004>
- Krivine, J.-L. (1971). *Introduction to axiomatic set theory*. Reidel.
- Krivochen, D. G. (2022). *Towards a theory of syntactic workspaces: Neighbourhoods and distances in a lexicalised grammar* [Unpublished manuscript]. <https://ling.auf.net/lingbuzz/005689>
- Krivochen, D. G. (forthcoming). *Syntax on the edge: A graph-theoretic analysis of sentence structure*. Brill.
- Kunen, K. (2013). *Set theory*. College Publications.
- Kural, M. (2005). Tree traversal and word order. *Linguistic Inquiry*, 36(3), 367–387.
<https://doi.org/10.1162/0024389054396890>
- Langendoen, D. T. (2003). Merge. In A. Carnie, M. Willie, & H. Harley (Eds.), *Formal approaches to function in grammar: In honor of Eloise Jelinek* (pp. 307–318). John Benjamins.
<https://doi.org/10.1075/la.62.22lan>
- Larson, B. (2015). Minimal search as a restriction on Merge. *Lingua*, 156, 57–69.
<https://doi.org/10.1016/j.lingua.2014.12.009>
- Lasnik, H., & Uriagereka, J. (2022). *Structure*. MIT Press.
- McCawley, J. (1968). Concerning the base component of a transformational grammar. *Foundations of Language*, 4(3), 243–269. <https://www.jstor.org/stable/25000330>
- McCawley, J. (1982a). *Thirty million theories of grammar*. University of Chicago Press.
- McCawley, J. (1982b). Parentheticals and discontinuous constituent structure. *Linguistic Inquiry*, 13(1), 91–106. <https://www.jstor.org/stable/4178261>

- McKinney-Bock, K., & Vergnaud, J.-R. (2014). Grafts and beyond: Graph-theoretic syntax. In K. McKinney-Bock & M. L. Zubizarreta (Eds.), *Primitive elements of grammatical theory* (pp. 207–236). Routledge.
- Medeiros, D. P. (2021). *Tree traversal grammar* [Unpublished manuscript]. University of Arizona.
- Milway, D. (2021). *Agree as derivational operation: Its definition and discontents*.
<https://ling.auf.net/lingbuzz/005842>
- Moro, A. (2000). *Dynamic antisymmetry*. MIT Press.
- Müller, S. (2015). HPSG: A synopsis. In T. Kiss & A. Alexiadou (Eds.), *Syntax – theory and analysis: An international handbook / Handbücher zur Sprach- und Kommunikationswissenschaft* (pp. 937–973). De Gruyter. <https://doi.org/10.1515/9783110363708-004>
- Nakashima, T. (2022). *How to generate adjuncts by MERGE* [Unpublished manuscript].
<https://ling.auf.net/lingbuzz/006649>
- Ojeda, A. (1988). A linear precedence account of cross-serial dependencies. *Linguistics and Philosophy*, 11, 457–492. <https://doi.org/10.1007/BF00668683>
- Osborne, T., Putnam, M. T., & Groß, T. (2011). Bare phrase structure, label-less trees, and specifier-less syntax: Is Minimalism becoming a dependency grammar? *Linguistic Review*, 28(3), 315–364. <https://doi.org/10.1515/tlir.2011.009>
- Ott, D. (2012). *Local instability*. De Gruyter.
- Ott, D. (2015). Symmetric Merge and local instability: Evidence from split topics. *Syntax*, 18(2), 157–200. <https://doi.org/10.1111/synt.12027>
- Panagiotidis, P. (2021). *Towards a (minimalist) theory of features: Preliminary notes* [Unpublished manuscript]. <https://ling.auf.net/lingbuzz/005615>
- Pollard, C. (1997). The nature of constraint-based grammar. *Linguistic Research*, 15, 1–18.
<http://isli.khu.ac.kr/journal/content/data/15/1.pdf>
- Postal, P. M. (2010). *Edge-based clausal syntax*. MIT Press.
- Preminger, O. (2019). What the PCC tells us about “abstract” agreement, head movement, and locality. *Glossa*, 4(1), 1–42. <https://doi.org/10.5334/gjgl.315>
- Pullum, G. (2020). Theorizing about the syntax of human language: A radical alternative to generative formalisms. *Cadernos de Linguística*, 1(1), 1–33.
<https://doi.org/10.25189/2675-4916.2020.v1.n1.id279>
- Romeijn, E. (2008). Random search methods. In C. Floudas & P. Pardalos (Eds.), *Encyclopaedia of optimization* (pp. 3245–3251). Springer. https://doi.org/10.1007/978-0-387-74759-0_556
- Sag, I., Gazdar, G., Wasow, T., & Weisler, S. (1985). Coordination and how to distinguish categories. *Natural Language and Linguistic Theory*, 3, 117–171. <https://doi.org/10.1007/BF00133839>
- Sarkar, A., & Joshi, A. (1997). *Handling coordination in a tree adjoining grammar* [Technical report]. University of Pennsylvania. <https://www2.cs.sfu.ca/~anoop/papers/pdf/tag-coordination.pdf>
- Schmerling, S. F. (2018). *Sound and grammar: Towards a neo-Sapirian theory of language*. Brill.
- Sedgewick, R., & Wayne, K. (2014). *Algorithms*. Addison-Wesley.

- Seely, T. D. (2006). Merge, derivational c-command, and subcategorization in a label-free syntax. In C. Boeckx (Ed.), *Minimalist essays* (pp. 182–217). John Benjamins.
<https://doi.org/10.1075/la.91.13see>
- Shieber, S. (1988). Separating linguistic analyses from linguistic theories. In U. Reyle & C. Rohrer (Eds.), *Natural language parsing and linguistic theories* (pp. 33–68). Reidel.
https://doi.org/10.1007/978-94-009-1337-0_2
- Sobin, N. (2014). Th/Ex agreement and case in expletive sentences. *Syntax*, 17(4), 385–416.
<https://doi.org/10.1111/synt.12021>
- Sobin, N. (2020). Directing syntactic traffic. *Syntax*, 23(3), 241–274.
<https://doi.org/10.1111/synt.12194>
- Stabler, E. (2011). Computational perspectives on minimalism. In C. Boeckx (Ed.), *Oxford handbook of minimalism* (pp. 617–641). Oxford University Press.
<https://doi.org/10.1093/oxfordhb/9780199549368.013.0027>
- Stepanov, A. (2001). Late adjunction and Minimalist phrase structure. *Syntax*, 4(2), 94–125.
<https://doi.org/10.1111/1467-9612.00038>
- Uriagereka, J. (2002). Multiple Spell-Out. In J. Uriagereka (Ed.), *Derivations: Exploring the dynamics of syntax* (pp. 45–65). Routledge.
- Uriagereka, J. (2012). *Spell-Out and the Minimalist Program*. Oxford University Press.
- van Gelderen, E. (2021). *Third factors in language variation and change*. Cambridge University Press.
- van Steen, M. (2010). *Graph theory and complex networks: An introduction*.
<https://www.distributed-systems.net/index.php/books/gtcn/gtcn/>
- Villata, S., Rizzi, L., & Franck, J. (2016). Intervention effects and Relativized Minimality: New experimental evidence from graded judgments. *Lingua*, 179, 76–96.
<https://doi.org/10.1016/j.lingua.2016.03.004>
- Wilson, R. (1996). *Introduction to graph theory*. Adison-Wesley.
- Wurmbrand, S. (2014). The Merge condition: A syntactic approach to selection. In P. Kosta, S. Franks, T. Radeva-Bork, & L. Schürcks (Eds.), *Minimalism and beyond: Radicalizing the interfaces* (pp. 130–167). John Benjamins. <https://doi.org/10.1075/lfab.11.06wur>
- Zwart, J.-W. (2009). Prospects for top-down derivations. *Catalan Journal of Linguistics*, 9, 161–187.
<https://raco.cat/index.php/CatalanJournal/article/view/168909>
- Zwart, J.-W. (2011). Order and structure: Asymmetric merge. In C. Boeckx (Ed.), *The Oxford handbook of linguistic Minimalism* (pp. 96–118). Oxford University Press.
<https://doi.org/10.1093/oxfordhb/9780199549368.013.0005>
- Zwart, J.-W. (2015). Top-down derivation, recursion, and the model of grammar. In A. Trotzke & J. Bayer (Eds.), *Syntactic complexity across interfaces* (pp. 25–42). De Gruyter.
<https://doi.org/10.1515/9781614517900.25>
- Zwicky, A., & Isard, S. (1963). *Some aspects of tree theory* [Working Paper W-6674, The MITRE Corporation]. <https://web.stanford.edu/~zwicky/some-aspects-of-tree-theory.pdf>
- Zyman, E. (2022). Phase-constrained obligatory late adjunction. *Syntax*, 25(1), 84–121.
<https://doi.org/10.1111/synt.12226>