The search for Minimal Search: a graph-theoretic view

Diego Gabriel Krivochen

University of Oxford

diegokrivochen@hotmail.com

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 ({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) defines an asymmetric connection from X to Y, where Y is an argument of X, 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: 109) 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: 43) which '*falls under MC* [Minimal Computation]' (Chomsky, 2015: 6). Similar characterisations of MS can be found e.g. in Epstein et al. (2015, 2020), Chomsky (2020a: 36, 2020b), Goto (2019), Komachi et al. (2019), Van Gelderen (2021), Larson (2015), Hayashi (2021), a.o. MS has become a key component of operations like Labelling (Bauke & Blühmel, 2017: 4; Chomsky, 2013, 2015, 2020a, b, 2021).

If *labelling* is a properly defined algorithm, an unambiguous sequence of steps that carries out a computational process which maps an input to an output of specific kinds (in this case, labelling would be a function from syntactic objects SO into labels), then MS being such a central component should also be properly defined. MS's relevance is not limited to labelling: it is invoked for a variety of operations. Collins (2017: 58) appeals to MS in a definition of linearisation (which highlights its fundamentally sequential nature, an issue we will return to); Chomsky (2013, 2015, 2020a, b) resorts to it in the discussion about subject movement (from Spec-*v*P to Spec-TP, seeking to eliminate the EPP feature; see also Ke, 2021) and accessibility in successive-cyclic movement (2020a: 36); Branan & Erlewine (2021), Ke (2019), Preminger (2019) and Milway (2021) analyse MS in the context of Agree, to name a few. 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., Ke, 2019, 2021; Milway, 2021; Preminger, 2019), as in Chomsky (2020a: 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 (2020b: 17):

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

In this context, it is essential to address the following question: 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. 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.

2. Two perspectives on structure building

Minimalist theory assumes that Merge / MERGE are instances of an allegedly irreducible operation of unordered set formation over elements in a workspace (Chomsky, 2019, 2020a, b): this is intended to capture the idea that *linear* order (the binary, transitive, total relation *precedes*) is not part of the 'syntactic component' of the grammar (but instead of the 'externalization systems'; Chomsky, 2019: 272). To give an example, Chomsky (2019: 257) defines *Inclusiveness* in such a way that '*the operations should not add anything: they should not add order*'. Importantly, he does not specify what kind of *order* he has in mind, if only linear precedence or if *any* order imposed over the output of Merge is ruled out. This commitment to unordered sets creates problems for the representation of empirically motivated syntactic relations, e.g., subcategorisation and argument structure (Langendoen, 2003: §2.3), labelling, and chains dependencies (Gärtner, 2002; 2021). In set-theoretic syntax, the structural description for a sentence like *The man falls* is (1) (taken from Collins, 2017: 65):

1) {{the, man}, {{φ, T}, {fall, {the, man}}}}

Unless selection is encoded in terms of containment, there is nothing in (1) that tells us that {the, man} is an argument of {fall}. What is more, if instead of {the, man} we had {John}[1], then {fall, John} would provide no asymmetric relation between a predicate and its argument. If we 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*. Contemporary

[1] We will go back to the problem posed by the set-theoretic status of *John* in **Section 5**.

Minimalism has emphasised that linear order is independent from structure building (Chomsky, 2021 goes as far as saying that '*linear order is simply not available to I-language*'). 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. To address the inadequacy of label-less unordered sets to represent argument structure, Langendoen (2003) defines an operation *list merge* which captures this idea (see also Chomsky, 2001, 2004):

2) $L(\alpha, \beta) = <\alpha, \beta>$ (if α is atomic)

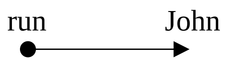
   $L(<\alpha, \ldots>, \beta) = <\alpha, \ldots, \beta>$ (if α is a phrase)

This order is important in order to define, among other things, immediate constituency and semantic interpretation: given a set of lexical items {…, John, run, …}, an operation that produces an unordered set {run, John} = {John, run} does not provide the same information as one that produces an ordered set <run, John>: only the latter can be put in direct correspondence with a propositional representation along the lines of *run*(*John*) with no further assumptions, where it is clear that *John* is an argument of *run* regardless of how those terminals are linearised (see Heim & Kratzer, 1998: 44). Langendoen provides explicit derivations where this recursive definition of ordered sets under *list Merge* (assuming the Wiener-Kuratowski definition of ordered sets; Krivine, 1971: 2-3. See Gärtner, 2002: 74, fn. 146; Dipert, 1982 for critical discussion) is used. Langendoen remains committed to sets as the format of structural descriptions (with trees being diagrams of derivations), however. We want to take this a step further. 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) approach, which interprets phrase structure rules as admissibility conditions for local trees, defines two distinct two-place relations over trees: *precedes* and *node dominates*[2]. The former pertains to linear order in a string; the latter, to hierarchy in structural

[2] 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.

descriptions that are graphs, not sets. This point is particularly relevant for this paper, as we will argue that a shift from set theory to graph theory as the model for structural descriptions in natural language syntax is desirable. Furthermore, there are proposals along these lines already. Within Minimalism, McKinney-Bock & Vergnaud (2014) argue for a graph-theoretic approach to phrase markers such that Merge generates directed graphs where the order imposed over the set of nodes is based on Selection or Checking. Outside Minimalism, some versions of Dependency Grammars (Kahane & Lareau, 2016) as well as Metagraph Grammar (Postal, 2010: 26) develop explicit graph-theoretic analyses where linear order is divorced from hierarchical relations. Thus, we may define a graph like (1), where there is a directed edge (an *arc*) from a predicate to its argument which encodes selection (but not necessarily precedence in a string):

3)



We notate a directed edge from *run* to *John* as *e*<run, John>. Note that this is not equivalent to the unordered set {run, John}, which gives no information about selection or predicate-argument relations[3].

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 justified. In this light, we will argue that (i) structure building must encode selection and argument structure, which can be accomplished with digraphs, and (ii) this leads to a natural characterisation of MS as a

McCawley (1982: 165) makes a strong case for linearly unordered graph structures, and distinguishes between strings and trees consistently.

[3] As pointed out by a reviewer, it is possible to encode things like order of composition in type-theoretic annotations or the definition of functional application. In that case, however, structure building remains divorced from argument structure (furthermore, none of the references on structure building and MS cited in this paper assume type-theoretically annotated syntactic objects). Krivochen (2022: 70) provides a semantic interpretation rule for digraphs along such lines within a non-derivational system:

> *A semantic interpretation SI function for L is a function from* <*N*[odes], *A*[ddresses]> *to S*[emantic values]. […] *SI(Exp) = S.*
>
> […] *In the walk through an elementary graph,* $e{<}v_1, v_2{>}$ *(an arc between nodes* v_1 *and* v_2*, where* v_1 *is a functor, and* v_2 *an argument) is the semantic value of* v_1 *applied to the semantic value of* v_2*.*

sequential search algorithm (in addition to avoiding or solving independent problems with set-theoretic syntax). In sum, we contend that Merge yields directed graphs, not unordered sets.

The paper is structured as follows. **Section 3** introduces search algorithms. **Section 4** introduces 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': {XP, YP} and {X, Y} in the context of the definition of MS as a search algorithm.

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 (Sedgewick & Wayne, 2017; Knuth, 1998). 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 (2020a, b) 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 (see Ke, 2021). 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 (Romejin, 2009). 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 imposed by the sequentiality of Merge, and if MS is indeed a central syntactic operation, it should build 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: 396, ff.): suppose we have a set of values $\{V_1, \ldots V_n\}$, each of which is assigned a *key* (an address that allows us to retrieve the value) $\{K_1, \ldots K_n\}$. Then, a search algorithm may be called upon to find a specific K_x. 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 find (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 in an increasing order (such that instead of $\{K_i, \ldots K_n\}$ we have $K_i < K_j < \ldots K_n$), if the algorithm is sensitive to the number of times a particular key has been accessed, if more than one 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 (Gorn, 1967: 124 specifies that only 'object characters' are indexed by the addresses; linguistically these correspond to categorematic expressions, see Schmerling, 2018). Recent characterisations of MS have assumed tree search algorithms (e.g., Ke, 2019 for Agree and Labelling; Branan & Erlewine, 2021, Preminger, 2019: 24 for Agree), as have some approaches to linearisation as tree transversals (Kural, 2005; Kremers, 2009; Medeiros, 2021 also makes use of transversals to model word order variations in a non-transformational model), even though the problematic relation between trees and set-theoretic representations is sometimes acknowledged (other times, as in Ott, 2012, trees and sets are taken as notational variants).

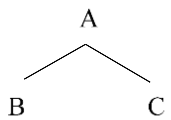
Sequential tree search algorithms, as assumed widely in the Minimalist literature, are broadly divided in two kinds: *breadth-first* and *depth-first*. Both have access to every node in a connected tree, but differ in terms of the order in which the search proceeds.[4]

A *breadth-first* search goes 'in waves' from the root, searching each generation from left to right[5] 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 (Even & Even, 2012: 11, 46-49; Cormen et al. 2001: 531, ff.). We can illustrate the order in which nodes are visited in a simple tree for both search algorithms given a rooted, directed tree:

[4] In a *parallel* search, all daughters of the root are explored simultaneously to find heads bearing specific features (Ke, 2021: 8). 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.

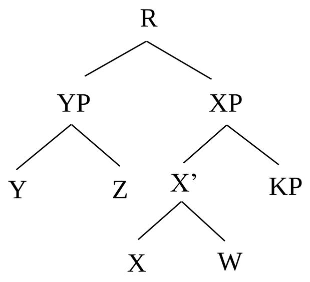
[5] The order in which nodes are checked depends on the type of transversal algorithm chosen, since there are three possible ways to implement it. Take a branching node as an example:



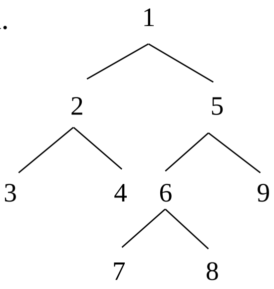
A *preorder* transversal defines the sequence: A B C (root, left, right)
An *inorder* transversal defines the sequence: B A C (left, root, right)
A *postorder* transversal defines the sequence B C A (left, right, root)

The choice of *preorder* transversals in this paper 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. The same transversal would apply to a system where all predicates dominate their arguments, as in Dependency Grammars. See also Kural (2005), Kremers (2009) for discussion and application of transversal algorithms for linearisation purposes.

4) a.

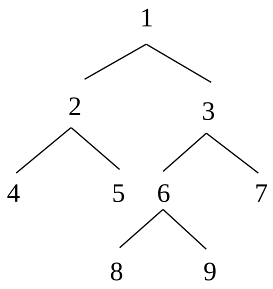


5) a.



Depth-first

b.



Breadth-first

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

6) Σ = <R, YP, **Y**>

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:

7) Σ = <R, XP, YP, **Y**, Z, X’, KP, X, W>

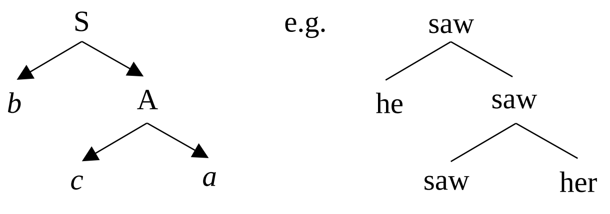
In this example both algorithms find Y before any other head; however, a depth-first search finds Y after visiting two nodes, where as a breadth-first search finds Y after visiting three.

As we will see in the next section, the Minimalist framework in which MS is needed is strongly committed to the idea that structural descriptions are *sets*: trees in most current versions of Minimalism are diagrams, not formal objects (e.g. Ott, 2012: 9; Epstein et al., 2020: 2). Since there is work on MS as tree transversal, and we have suggested that the output of Merge is a directed graph and not an unordered set, we must now formally define graphs and compare Minimalist sets with graphs.

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 (1) above. We will use $e\langle a, b\rangle$ to denote an arc from *a* to *b*. The *indegree* of a vertex v_x is the number of edges that go *from* another vertex to v_x; the *outdegree* of v_x is the number of edges that go *from* v_x to some other vertex: in (3), the *indegree* of *run* is 0, and its *outdegree* is 1. 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: 113).

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 vertices[6]), *directed* (such that edges have directionality; this defines the binary asymmetric relation *dominates* for every two adjacent vertices), and *rooted* (there is a node that is not dominated by any other node). We can illustrate these notions in (8):

8)


In (8), V = {*a*, *b*, *c*, A, S}. Nodes A and S have outdegree 2, A has indegree 1, S has indegree 0. Nodes *a*, *c*, and *b* have indegree 1 and outdegree 0. S is the *root* of the graph. A node with outdegree 0

[6] This is because contemporary Minimalism rejects Multidominance (Chomsky, 2019), sometimes on the grounds that multidominance trees are graphs, not sets (e.g., Collins & Groat, 2018). See Gärtner (2021) for critical discussion about the Copy Theory of Movement pertaining to this point.

can only be the last node in a walk through a tree, whereas a node with indegree 0 can only be the first node in a walk. In X-bar-style trees, heads always have indegree 1, and non-heads always have outdegree 2: binarity has been a staple of Merge since its inception (Chomsky, 1995: 189-190), and continues to be an axiom of structure building under MERGE (Chomsky, 2020a: 22; 2020b: 18).

A set theoretic representation of (8), following recent Minimalist literature, would be (9):

9) {S, {*b*, {A, {*a*, *c*}}}}

In (9), 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) ⊂ V(S) and E(A) ⊂ E(S). From this perspective, both approaches allow us to capture the same relations. In graph-theoretic terms, we define the *root* of a directed tree as a designated node that is not dominated by any other node (Wilson, 1996: 56) or in terms of the definition of paths in G:

> *A digraph G(V, E) is said to have a root τ if τ ∈ V and every vertex v ∈ 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: 37)

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

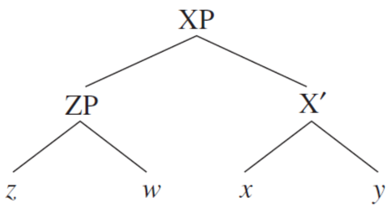
> *K is the root iff:*
>
> *for any Z, Z a term of K, every object that Z is a term of is a term of K.* (Epstein et al., 2012: 262)

In set-theoretic terms, 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 ⊂ X).

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

> *Suppose that we have the structure represented informally as (9), with x, y, z, w terminals*

(9)



Here 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 translation 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 clear why they are used at all). As per *bare phrase structure* (BPS, Chomsky, 1994), they are not part of the syntactic representation (also Collins, 2017: 49). 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 the formal definition of a graph.

Chomsky's fragment suggests that the choice between sets and graphs as the format of linguistic descriptions is merely notational. However, this is not the case: there are relations and operations that we can define in objects like (3) and (8) (graphs) but not in objects like (9) (unordered sets). If these relations are relevant for the definition of MS, then a set-theoretic stance runs into trouble given the centrality of MS in the definition of other, independently defined, syntactic processes. For example, in (8) we can define a *walk* W between nodes *b* and *c* as an ordered set of nodes and edges:

10) $W_{b,\, c} = <e<b, S>, e<S, A>, e<A, c>>$

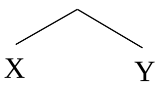
In (10) we have an ordered set of arcs: each arc, in turn, is an order over nodes. (10) tells us that the edge from *b* to S is walked on first, then we go from S to A, and finally from A to *c* (an *inorder* transversal). It is not possible, however, to define such a walk for a set-theoretic representation like (9) unless a total order is defined, which requires an additional operation on top of Merge[7]. Set-theory

[7] As a reviewer points out, Collins & Stabler (2016: 51) define *path* in terms of recursive immediate containment:

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, 1995: 247; 2020a: 22; see also Epstein, 1998 for a derivational definition of c-command built on these notions).

Directed graphs give us different kinds of information: we can, for example define *walks* from node n_1 to node n_2, and compare the length of different walks; if Merge produces digraphs, an order is delivered by the structure building operation without additional stipulations. The possibility of defining walks is fundamental for a proper definition of MS, and it seems to underlie the account of several empirical phenomena (e.g., anaphoric binding, as in Kayne, 1984; Sternefeld, 1998; Gärtner, 2014). If we allow Merge to output digraphs, not only can we define encode argument structure, but also define unambiguous sequences for MS. These graph-theoretic structural descriptions are very different from the illustrations of set-theoretic derivations commonly used in Minimalism. We may ask, then, whether there is enough information in a set-theoretic representation that allows us to map it into a digraph where search algorithms can apply, unambiguously (see Branan & Erlewine, 2021 for discussion).

Problems arise when tree diagrams are used to graphically represent set-theoretic Merge (Postal, 2010: 7, ff.). Suppose that we Merge X and Y (for X and Y arbitrary syntactic objects in the workspace), yielding the set {X, Y}. This is diagrammed using a binary-branching tree:

11)


From a set-theoretic perspective, (11) may be seen as a graphical representation of the set {X, Y} (and it is assumed to be so in the Minimalist literature). However, from a graph-theoretic perspective we need to consider the fact that (11) has *three* nodes, not two (as noted in Chomsky, 2020a: 39). The

---

A path is a sequence of syntactic objects ⟨SO_1, SO_2, ..., SO_n⟩ 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 (9), we can define a path ⟨S, *b*, A⟩ 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 (9). Additional structure needs to be invoked to create a containment relation. The graph-theoretic approach requires no such complication.

tree in (10) is rooted, because there are *two edges*, each edge connecting two nodes. Since the edges converge, it means that there is a node (which we will call ●, remaining agnostic about the indexed category it is to be assigned to; Collins & Stabler, 2016: 48 take ● to be a set which in (11) would be 'pointing to' its elements, somewhat mixing sets and graphs) such that e_1 <●, X> and e_2 <●, Y>. The formal object defined as the set {X, Y} and the graph in (11) are thus, crucially, distinct and not mutually translatable. A more accurate representation of Merge(X, Y) = {X, Y} would be (12) (McKinney-Bock & Vergnaud, 2014: 219, ff.):

12) X ——— Y

Here, no new nodes are introduced[8], we have just the input of Merge and a non-directed edge connecting the two terms involved in the operation. This is all we can do with the information contained in the output of Merge as unordered set formation.

The alternative is to define that Merge is asymmetric, driven by the satisfaction of selectional requirements (Chomsky, 1995: 246; also Minimalist Grammars, e.g. Stabler, 2011). In this context, if Merge of Y to X satisfies a selectional requirement on X[9], then Merge(X, Y) = <X, Y>, and we need an arc *from* X *to* Y, represented here as an arrow (McKinney-Bock & Vergnaud, 2014: 220 use arrows to indicate that Y projects, but we depart from that convention):

12') $X \longrightarrow Y$

In Minimalist Grammars, for example, X would have a selector feature =F, matched with a categorial feature F in Y. Implemented as in (12'), structure building delivers not an unordered set, but a digraph[10]. Note that in defining (12') we have not lost any of the classical properties of Merge-based

[8] Incidentally, (12) and directed variants of it satisfy Chomsky's (2020b) *minimal yield* better 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.

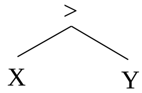
[9] Specifically, X and Y are either basic or derived categorematic expressions of the language which are assigned a uniquely identifying address in the workspace (see Gorn, 1967 for an early implementation of an indexing system and Sarkar & Joshi, 1997 for an application of addressed expressions to lexicalised TAGs; also Krivochen & Padovan, 2021).

[10] This notation allows McKinney-Bock & Vergnaud to dispense with the additional node that is used in Minimalist Grammars -as in (i)-, since the asymmetry of selection is encoded in the arc:

syntax: binarity, recursion, and discrete infinity. 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. We have, however, given up unordered sets: (12') may now be interpreted as a dependency-like structure capable of representing predicate-argument relations (Osborne et al., 2011 for 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, we can let X = *read* and Y = *books*: Merge of *books* to *read* satisfies a selectional restriction in *read*. Encoding this syntactic relation as an arc allows us to capture the claim that '*argument structure, is invariably given by external MERGE*' (Chomsky, 2020a: 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 (10)) 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**). We take this to be a desirable result (but see Milway, 2021; Ke, 2021 for the opposite conclusion). 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 been an aim of Minimalist inquiry (Chomsky, 1994; Collins, 2002, 2017).

Trees and sets are not equivalent nor are they 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 since part of our goal is evaluate the feasibility of MS as a search algorithm defined over sets formed by Merge/MERGE and over graphs. If MS is as core 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

i)



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 models 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 then to characterise the generative operation to evaluate the feasibility of defining a search algorithm for its output. Collins (2017: 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 (Merge(X, Y) = Merge(Y, X))
- (The output of) Merge is unspecified for linear order (Chomsky, 2013: 40; 2020b)
- (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: 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 & Narita (2014); Epstein et al. (2015, 2020); Chomsky (2013, 2015, 2020a, b), among others. Chomsky's recent works (2019, 2020a, b) propose a version of the generative operation called MERGE, which involves unordered set formation plus removal from a workspace:

13) Workspace (WS) contains X and Y: $[_{WS}$ X, Y]

MERGE(X, Y) = $[_{WS}$ {X, Y} X, Y][11]

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 latest 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, 1995: 243)

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

> *K must therefore at least (and we assume at most) be of the form {γ, {α , β}}, where γ identifies the type to which K belongs, indicating its relevant properties.* (Chomsky, 1995: 243)

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'.

[11] Chomsky (2020a: 34, 2020b: 19) formulates the input of the operation as MERGE(X, Y, WS), as if WS was a syntactic object.

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 do we know that the search is successful? This 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, co-indexing, etc.). We will focus on labelling, since it has received the most attention in the Minimalist literature; even though MS also underlies Agree (Ke, 2019, 2021; Branan & Erlewine, 2021; Milway, 2021; Preminger, 2019) and successive cyclic movement (Chomsky, 2020a: 36), they have not received as much attention as labelling by MS. The formal 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: 7; also 2013: 43)

Exactly how 'trivial' is it? 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 we have

14) {*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}?[12]) 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, evaluating the cardinality of a set built by MERGE (which, as

[12] As observed by a reviewer, *read* may have internal structure: {*v*, √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.

Kunen, 2013: 62 observes, always entails defining injections/bijections between two sets). A head would be a set of cardinality 1, a non-head would be of cardinality greater than 1. Or, the labelling algorithm must distinguish sets of features (lexical items; Collins & Stabler, 2016: 44) from sets of sets of features (i.e., phrases): this latter criterion is explicitly adopted in Ke (2021). It is important to determine, then, the relation between SO, workspaces, and sets. Chomsky (2020a: 37) says:

> *We want to say that [X], the workspace which is a set containing X is distinct from X.*
>
> *[X] ≠ 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 confusing[13]. Let us explore some of its consequences. If MERGE involves an operation of removal or replacement such that a workspace containing *a* and *b*, notated [*a*, *b*] contains the set {*a*, *b*} with *a* and *b* removed from the workspace as the output of MERGE, then determining what is a head and what is not precisely requires the aforementioned kind of cardinality-sensitive procedure: this process is independent from (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 contains, as before, the singleton {*read*} and the set {*some, books*}:

15) Workspace: [{*read*}, {*some, books*}]

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

16) MERGE({*read*}, {*some, books*}) = [{{*read*}, {*some, books*}}]

[13] And may lead to inconsistencies. For example, {∅} is a subset of every set, but ∅ is not an element of every set.

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 observes, |{read}| in isolation has no meaning). Collins (2017: 56) provides a picture that is compatible with our characterisation[14]:

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

Which is exactly what we sketched above: following Chomsky's (2020a) 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, 2020a), then when we have {*read*, {*some*, *books*}} the system may simply find the object that is *not* a set: given Merge(A, B), 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: 45-46), but that it is independent from MS. Suppose for a moment that the only difference between XP and X was that the former is a set, but the latter is not (as the logical alternative to the scenario sketched in the previous paragraph). Then, (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. If Merge is limited to set + non-set situations ({H, XP}), it would exclude any version of Pair-Merge (or equivalent operations to yield sister-adjunction and Chomsky-adjunction). Interestingly, from this perspective, the configurations that have been dubbed problematic for MS (e.g., in Chomsky, 2013, 2015; Epstein

[14] For purposes of Agree by MS, this characterisation is challenged by Preminger (2019: 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').

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 independent from 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, 1995: 243), it is not sufficient to determine cardinality, since both sets are unary (|{books}| ≈ |{read}| in Kunen, 2013). 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:

(i) transform a relation between two singletons into a relation between a singleton and a set with a greater cardinality by adding more structure (thus multiplying elements in representations)

(ii) remove one of the objects after additional structure has been introduced (multiplying steps in derivations)

Both involve complications and departures from Minimalist desiderata (Chomsky 1995: 151), derived from set-theoretic commitments.

An alternative is readily available: the output of Merge(*read, books*) is a digraph *e*<*read*, *books*>:

17) read ⟶ books

The order, as above, is not precedence, but determined by subcategorisation: *books* is an argument of *read*. We could, for example, assume the following featural compositions for *read* and *books* (based on Stabler, 2011):

18) read :: =D =D +case V

books :: D -case

=F is a selector feature, which requires Merge with a category F, +case is a licensor (or 'probe') feature, and -case 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 (for labelling or Case checking) without the need for additional structure[15]. Importantly, (18) is a notational choice (alternatively, we could specify the type of each expression, defining also the order of composition; however, type-theoretic annotations are not themselves substitutes for structure building, nor do they define a search sequence). 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, 2020a: 36; 2020b: 17; Kitahara, 2020), it is because the data that constitutes the probing space for that algorithm is ordered. This is precisely what we want for structural descriptions for natural language sentences. From a set-theoretic perspective, *co-membership* and *containment* are the only relations

[15] There is a potential redundancy between selector-categorial 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).

created by MERGE. If every SO (head or phrase) is a set, then we can sketch the following derivation (Chomsky, 2020a: 48):

19) a. {b}
    b. {a, b}
    c. {b_1, {a, b_2}}

Here, b_1 is Internally Merged to the set {a, b}. According to Chomsky (2020a), MS renders b_2 inaccessible to further operations because b_1 is found first. This entails, however, that the system knows (i) that it is looking for an object of category b, and (ii) that b_1 and b_2 are copies of each other (or 'occurrences' of *b*; see Collins & Stabler, 2016: 51), neither of which follow from either Merge or MS. Here, an order based on containment seems sufficient: there is a set that contains b_2 and excludes b_1, but there is no set that contains b_1 and excludes b_2 (so (19c) must be a multiset; see Gärtner, 2021 for discussion). 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* in the antisymmetric literature; see Uriagereka, 2002; Kitahara, 2020):

20) a. {reads, {books}}
    b. {{the, man}, {reads, {books}}}

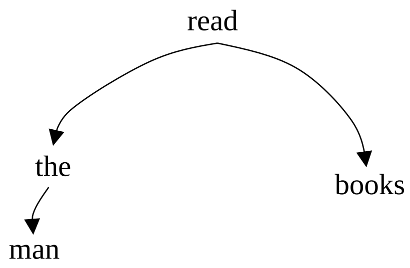
Graph-theoretically, the dependency between the lexical predicate and its arguments can be minimally represented as arcs, as can dependencies between members of each SO[16]:

[16] We assume in (21) a DP-like structure, where the noun is selected by a D head. In contrast to McKinney-Bock & Vergnaud (2014), we do not incorporate intermediate nodes in the graph-theoretic representations. They 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 Branan & Erlewine, 2021; Preminger, 2019). Thus, Checking would hold for walks, not single edges (however, Gärtner, 2002: 85 argues that there is no long-distance checking: in this case, both Checking and Selection would be strictly local operations). We can annotate edges with other information, such as grammatical functions, as in Relational Grammar, or encode that information in ordered relations between edges (Krivochen, 2022).

21) a. *e*<read, books>

b. *e*<the, man>

c. *e*<read, the>



Neither (20) nor (21) feature intermediate nodes, but only (20) encodes selection. If External Merge delivers argument structure, digraphs seem better suited than unordered sets to encode syntactic dependencies . As in (10), we also need to impose an order over arcs: a structural description is an ordered set of arcs. For (21), we have (21')[17]:

21') G = <*e*<read, the>, *e*<read, books>>

The order between edges can be defined in several ways: one possibility is to have the order represent order of composition. In that case, (21') defines a top-down parse (Zwart, 2009, 2015 and much related work). Alternatively, as explored in Krivochen (2022), the order may be dictated by the Grammatical Function hierarchy (Keenan & Comrie, 1977; Bell, 1983; Postal, 2010; Dalrymple et al., 2019, among many others). In this case (admittedly, a departure from Minimalist assumptions about the epiphenomenal nature of grammatical functions in the grammar) the order of dominance relations corresponds to the order of grammatical relations, such that if a relation (x, y) is ordered before a relation (x, z) in terms of dominance, the grammatical relation (x, y) is also ordered before (x, z) in the hierarchy of grammatical functions (the two orders are isomorphic, which means that there is a function f with an inverse that relates both sets of orders: we can go from one to the other no questions asked). In (21'), then, the constituent with head *the* is the Subject of *read*, since Subject is the highest grammatical function in the hierarchy, and the constituent with head *books*, ordered *after* the constituent with head *the*, is the Object (the next grammatical function in the hierarchy). Other options

[17] For simplicity, we assume here that *the* is a basic expression of the language, and thus corresponds to a node in the graph (the resulting structure being equivalent to a DP analysis). An alternative, explored in other works, is to assume (i) that only categorematic expressions correspond to nodes in structural descriptions, and (ii) that determiners are syncategorematic (an NP analysis). In that case, the ordered set of arcs would be G' = <*e*<read, man>, *e*<read, books>>.

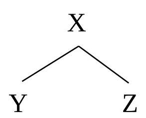
are available, within the limits of the formalism. A digraph is, then, much more informative than an unordered set for purposes of MS as well as semantic composition.

In contrast, as observed above, 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 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}} (<X, Y> or <Y, X> under the Wiener-Kuratowski definition of ordered sets, see above). 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. If MERGE creates Chomsky-Collins sets, and a search algorithm is to be defined over those sets, they must be ordered. If labelling is driven by MS we have 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 symmetry of Merge.

5.2 Symmetric and asymmetric structure building

Suppose we have a tree of the form

22)



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 a symbol assigned to a category, 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 = *v*P, because

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

This does not help in understanding how 'search' is implemented, but the idea is intuitive. It 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 (Late Merge; 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 AdvP. To this end, Chomsky (2020a: 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 (*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 (Chomsky, 2004; Langendoen, 2003), 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 syntactic objects 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)[18]. It

[18] In Multiple Spell-Out, 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: 91) material derived in a parallel workspace cannot provide a label since they are not part of the SO at the time of labelling:

> *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*

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 it is a case of MS is not explained. Recall that the original definition of Merge argues for the *asymmetry* of the operation:

> *The operation Merge(α , β) 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, 1995: 246)

Labelling in the context of BPS was encoded in Merge itself, with some versions requiring Merge to be triggered by featural requirements (Wurmbrand, 2014; Sobin, 2014; Panagiotidis, 2015; Stabler, 2011): 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.

In any case, the situation first explored by Chomsky ({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; Preminger, 2019, and Branan & Erlewine, 2021); we can adapt the simplest sequential search algorithm presented in Knuth (1998: 396) to the present context (see also Ke, 2019: 44. Preminger, 2019: 24 defines a sequential search algorithm for purposes of Agree that instead of heads searches for a suitable goal starting from a probe):

23) Given a sequence of syntactic objects SO_1, SO_2, …SO_n, the algorithm searches for a head.

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

Under present assumptions, selection in combination with derivational timing delivers the results we want: on the one hand, adverbs do not satisfy the selectional restrictions of the verb *read*; adding *carefully* to *read books* results in an expression of the same category of *read books* (whereas adding *books* to *read* changes the category from V to VP), see Dowty (2003). On the other, the adjunct is introduced late in the derivation.

Step 2: Compare. If SO_1 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 2021 *Search Target*, which is 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' or 'free' Merge could do. Indeed, computational implementations of Minimalism do away with blind Merge and unordered sets[19]. In our digraph alternative, in contrast, order is a property of the structural description to which MS applies, and independently motivated by argument structure.

6. Labelling the unlabelled

It is worth 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, for instance, the 'labelling' problem does not exist: in a structure like (21) X is introduced in the derivation one turn before Y and Z, and this introduction is precisely calling the indexed category to which X belongs. Labelling is also not a problem in top-down

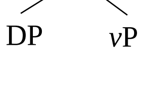
[19] Minimalist Grammars (Stabler, 2011; Fong & Ginsburg, 2019, a.o.) assume a feature-rich system with a designated set of labels and projection indicated at every point in the tree. There is, to the best of our knowledge, no implementation of free Merge and MS-based labelling in Minimalist Grammars. Ginsburg's (2016) proposal, inspired heavily in Chomsky (2013, 2015) finds itself 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: 245) consider unnecessary.

Minimalist models of syntactic derivations (e.g., Zwart, 2009, 2015; Sobin, 2020 and references therein). Under MERGE assumptions, in contrast, it is necessary to provide a label to an object *after* that object has been created, counter-cyclically. Furthermore, it has been proposed that SO may remain label-less (Collins, 2002; Citko, 2008; Hornstein, 2009). The idea of label-less objects pushes contemporary Minimalism even farther away from the well-defined grammars in canonical form that Chomsky himself 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 what a label-less object commits us to for purposes of MS. In set-theoretic terms, the 'label' of a set must be a member of the set since there is no other option: set theory allows for objects and operations, all objects are either sets or members of sets (notation like $\{_\alpha X, Y\}$ is not standard set theory). In graph-theoretic terms, in contrast, a 'label' is the root of a graph. In other words: the label of a SO is the node that is the root of the local tree which defines it. 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 (e.g., α in 'Move-α' should be a variable over sets, such that for any α, α a set, α can be 'moved').

In what pertains to MS, it is unclear how approaches with *unlabelled* objects make this work[20]. We can give an example (adapted from Ott, 2015; Van Gelderen, 2021: 14, ff.): suppose we have (24) created by Merge (trees are diagrammatic only):

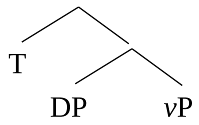
24)



Set-theoretically: {DP, *v*P}

[20] In some Minimalist works (e.g., Chomsky, 2008; 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: 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: 157).

Neither object is a head, so LA cannot apply. Note, however, that both *v*P and DP are labels, presumably proxies for sets (i.e., a way to refer to a set like {the, book}, but not a member of a set itself). The object in (24) merges with T, yielding (25):

25) 

Set-theoretically: {T, {DP, *v*P}}

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 (24) since there is no label (Epstein et al., 2013: 254); graph-theoretically, there is, 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), say[21]

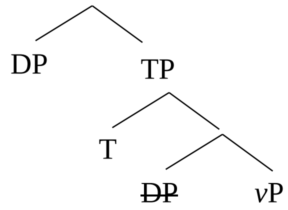
> *Suppose SO = {XP, YP}, neither a head. Here* ***minimal search is ambiguous****, locating the heads X, Y of XP, YP, respectively. There are, then, two ways in which SO can be labeled: (A) modify SO so that there is only one visible head, or (B) X and Y are identical in a relevant respect, providing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

As in Moro (2000), a point of symmetry is broken via movement. In (24) the DP must move, thus leaving the *v*P to be projected as a label counter-cyclically (henceforth, we leave aside the problem of whether (26) should be formalised as a multiset):

[21] Collins (2017: 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) = <F,F>.*

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

26) 

Set-theoretically: {{DP}, {TP, {T, {DP, *v*P}}}}

This process involves backtracking: the object {DP, *v*P} must remain somehow active and accessible, requiring a label, and the system must be able to modify a SO, adding a new element to a previously formed set: {T, {DP, *v*P}} becomes {T, {*v*P, {DP, *v*P}}}. Importantly, this kind 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: 248 equate to *exocentric* structures, for unclear reasons[22]), 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 accounts that would be otherwise impossible.

In the literature, the fact that DP movement in structures like (25) (and similar cases, such as copulative sentences of the form {copula {XP, YP}}, e.g. {{my, sister} {is {{my, sister}, {an, artist}}}}) is required for labelling reasons is usually considered an advantage, insofar as it allows for the elimination of the EPP. However, it is unclear whether 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 if the supposed simplification of the generative procedure entailed by unordered set formation outweigh 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 suggests[23]: if

[22] Exocentricity seems to us to be clearly distinct from the lack of a label. A phrase structure tree defined by the rule S → NP VP is exocentric, but labelled.
[23] We leave aside, for reasons of space, a discussion of the procedure that labels (26) as <phi, phi> (Chomsky, 2013: 43). If <phi, phi> is intended as an ordered set, it is not straightforward to formally characterise a labelling operation that can yield either a unary set (e.g., {VP}) or an ordered pair. There is also the question of what the contribution of <phi, phi> is to semantic interpretation and how constituency can be read off that label (which is a pair of features, or a proxy for one).

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 reviewed the difficulties of defining a search for unordered sets, but, 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 (McCawley, 1968, 1982; Kayne, 1984; Gärtner, 2002, 2014; Arikawa, 2019 for a variety of perspectives). There are two options. We may assume that labelling is *not* an operation triggered by a head (but, for instance, by interface legibility conditions). 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 MS. Here, we are asking about the distance between that root R and a head H: call it *d*(R, H). If, on the other hand, we assume that the search is triggered by a head (as seems to be the case for Agree; see Preminger, 2019; Ke, 2019; Branan & Erlewine, 2021) 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 starts from a probe P searches through a tree and finds a target T, and determines *d*(P, T): the length of the unique path that goes from P to T. Importantly, in this context, a proper definition of 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.

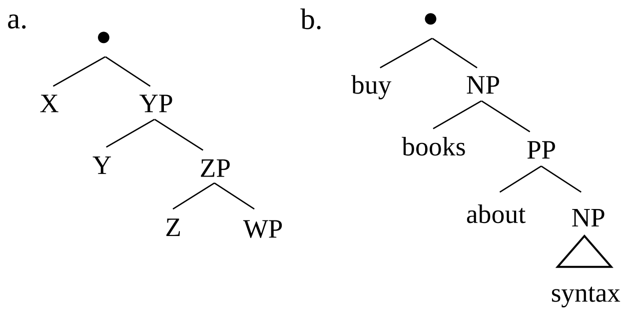
If we follow Minimalist proposals, once a head is found the algorithm terminates. This can be generalised: once the algorithm finds an object in the input that matches its target, the search halts (Chomsky, 2020b: 17). This brings up the additional problem (noted above) of having the system know what it is looking for *a priori*. 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, but, can graph-theory make things easier?

In **Section 4** we introduced the notions of *indegree* and *outdegree* of a vertex. In this context, we can define:

- The *root* of a tree 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

In the specific case of Minimalist structures, which are rigidly binary-branching and obey the *Single Mother Condition*, then the outdegree of any node is either 0 or 2, and the indegree of a node is either 0 or 1. The definitions above 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 transversals (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 the root (for labelling) or a designated internal node (for Agree) to a node with indegree 1 and outdegree 0. Let us see an example. Given a classical phrase structure tree with intermediate nodes like (27)

27)



the search sequence from the root until the first node with outdegree 0 would be Σ = <●, X> (i.e., Σ = <●, buy>), both under depth-first and breadth-first algorithms. Assuming Chomsky’s LA, this identifies ● as XP (VP in (27b)).

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:

28) 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 (29):

29)

If a set like {XP, YP} gives enough information to create a graph like (29) (not an uncontroversial question; see Branan & Erlewine, 2021), suppose that the label of the new node created by this merger is to be determined by MS. If (29) is a digraph, *e*<XP, WP> ≠ *e*<WP, XP>. A walk through (29) would contain the sequence <●, XP, WP> but not <WP, XP, ●>. 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, where search halts. The depth-first sequence would be (30):

30) Σ = <●, XP, X>

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

31) Σ' = <●, XP, YP, X>

In neither case is there an ambiguity: when the algorithm finds an object that matches the target, the search stops. In both cases, depth-first and breadth-first, X is found, and the search should stop. It is not necessary to compare the distance between the root and X vs. the root and Y; if MS is only search (i.e., it cannot do anything other than probe structure sequentially and compare each node visited with a target value; see e.g. Preminger, 2019), no ambiguity arises[24]. If the system can ignore a target and

[24] Alternatively, a parallel search may be conducted in order to preserve the ambiguity (as in Ke, 2021).

keep looking in a different branch of the graph, it is a departure from the simplest case that should (in accord with the Minimalist viewpoint) be independently justified. This may be the case if the specification of the target of the search includes the requirement that, in addition to being a node with outdegree 0, the target have a specific featural makeup: for example, the interpretation of a filler-gap dependency may be construed as a search[25]. In that case, and assuming that filler and gap are distinct nodes, the gap must be identical to the filler (but see Gärtner, 2002: 87, ff., 96) and bear a *wh*-feature. Consider the underlying structure of a *wh*-interrogative:

32) [What [C [Mary T [[Mary] read [what]]]]]?

If C searches for something 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 target. Similarly, for reconstruction *what* needs to find a copy of itself which has not checked its uninterpretable *wh* feature.

If an adequately restrictive meta-theory of features is devised, such a possibility is not unreasonable. Indeed, Preminger (2019) formalises an approach to sequential MS where only featural (but not relational) information matters when determining whether the SO being evaluated is a suitable target: this is compatible with our approach to structure building. What to do after a target has been found is a different matter from the search itself: it is copied and re-merged at the root, satisfying the Extension Condition and the Single Mother condition (movement)? Is a new edge created between the root and the target, such that walks in the graph become *trails* and nodes may have indegree greater than 1 (multidominance)? We propose that the format of the representation can be simplified with respect to classical Minimalist diagrams, and this also allows for MS to apply unambiguously. If the output of Merge is a digraph, and each node corresponds to a uniquely indexed expression (Sarkar & Joshi, 1996; Karttunen & Kay, 1985; Gorn, 1967), *what* in Spec-C and *what* in Compl-V are the same node: in this way, chains and copies are effectively eliminated. If the featural composition of

[25] Not just in transformational models. For example, LFG's treatment of long-distance dependencies, using functional uncertainty, defines a dependency path as a regular expression. 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.

what changes in the derivation, it is again possible to define a search sequence through the digraph for purposes of reconstruction, but here the nodes remain distinct. We can build a digraph in the workspace stepwise (recall that arcs go from predicates to their arguments and that, following McKinney-Bock & Vergnaud, 2014, two relations may correspond to an arc: *selection* or *checking*):

33) a. *e*<read, what> ⟶ Selection

b. <*e*<*v*, read>, *e*<read, what>>

c. <*e*<*v*, Mary>, *e*<*v*, read>, *e*<read, what>> (at this point, having saturated the valency of the predicate, we can read off grammatical functions from the ordered set of arcs: *Mary* as a Subject and *what* as an Object)

d. <*e*<T, Mary>, *e*<*v*, Mary>, *e*<*v*, read>, *e*<read, what>>

e. <*e*<C_{wh}, T>, *e*<T, Mary>, *e*<*v*, Mary>, *e*<*v*, read>, *e*<read, what>>[26]

At this point, we can define a search sequence triggered by C_{wh}:

34) $\Sigma = \langle C_{wh}$, T, Mary, *v*, read, what$\rangle$

If expressions are assigned uniquely identifying addresses in the workspace (Krivochen, 2022), then a new edge can be created from C to *what* (under Minimalist assumptions, *what* is copied and re-Merged, delivering a multiset; digraphs pose no such complications):

f. <*e*<C, what>, *e*<C, T>, *e*<T, Mary>, *e*<*v*, Mary>, *e*<*v*, read>, *e*<read, what>>

Checking:

C :: +wh
What :: D -wh

35) Reconstruction search sequence: $\Sigma = \langle \text{what}_{Probe}$, C, T, Mary, *v*, read, $\text{what}_{Goal}\rangle$

In neither breadth-first nor depth-first searches (in any of the three transversal possibilities) is {XP, YP} problematic: if SO are defined in graph-theoretic terms, the search parameters can be characterised unambiguously. We can capture argument structure and define a unique sequential

[26] Note that *Mary* is multidominated: excluding multidominance under graph theoretic assumptions requires additional stipulations. Given a unique address system for nodes, no 'copies vs. occurrences' problem arises (cf. Collins & Groat, 2018).

search transversal by appealing to no additional mechanisms or operations, only Merge(X, Y) = *e*<X, Y>.

In (29) above both XP and YP are *accessible* for the search algorithm, but it does not mean that it is an ambiguous configuration that needs to be reorganised: either XP or YP will be probed into first in a sequential search (see Kural, 2005: 373-376 for an explicit implementation of linearisation-as-transversal that also finds no issue with ‘symmetry points’; contra Collins, 2017: 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.

Mapping (29) into a structure where either XP or YP have moved is certainly possible; indeed, it is the preferred way in the recent Minimalist literature to dissolve a symmetry point. However, the justification for such movement is not in the search algorithm, since there is nothing wrong with the configuration itself for purposes of MS, as we have just shown. The argument is equally valid if the search starts above ●. For example, take (36):

36) ?

DP *v*P

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 -T- DP is internally Merged at the root, leaving behind a copy, and (iii) the object in (36) is counter-cyclically labelled *v*P because the copy would be ‘invisible’ for the labelling algorithm (we mentioned some additional complications entailed by (iii) under (24)). 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 Kural, 2005; Kremers, 2009 for a view from linearisation). 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; furthermore, a stipulation that copies are ‘invisible’ for labelling is needed.

The important issue is this: 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* transversal (Kural, 2005), and make sure that the left or right branches are always, consistently, searched first (so-called 'directionality parameters'). The 'ambiguity' of {XP, YP} situations is an artefact of set-theoretic commitments, not a necessary formal property of structural descriptions or of MS.

The same formal argument is valid for {X, Y} situations: a node dominating two nodes with indegree 1 and outdegree 0 each. 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, etc.). Presumably, there could be other cases, depending on how much silent structure is allowed. For example, the bolded sequences in (37a-b) should involve two terminal nodes:

37) a. I **love her** (pronounced /lʌvɚ/)
b. John **reads it**

Under bare phrase structure assumptions, *love* and *her* should be two heads. If this is correct, then monotransitive verbs with pronominal 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 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 (38a) (which gives no information about argument structure or hierarchy) we have the local object in (38b) (note that the valency of *love* is not saturated yet, for there is still an argumental slot to fill):

38) a. {love, her}

b.

$$\begin{tikzcd} \text{love} \arrow[d, bend right] \\ \text{her} \end{tikzcd}$$

Only (38b) 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.

8. Conclusions

The aim of this paper was to explore the possibilities to define MS as a search algorithm and its interaction with structure building. We argued that there are serious difficulties in defining MS for unordered set-theoretical structural descriptions, given current assumptions about Merge and its output. We contend that the set-theoretic commitments of Minimalism conspire against a definition of sequential search algorithms over the output of Merge[27]. We thus propose to replace

39) Merge(X, Y) = {X, Y}

with

[27] Branan & 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. The same set-to-graph translation seems necessary in Ke (2019) and Milway (2021).

40) Merge(X, Y) = *e*<X, Y>

Where X takes Y as an argument (for X and Y elements in the workspace). In this view, syntactic structures are ordered sets of arcs. This has a number of consequences, of which we will highlight two: first, sequential MS can apply unambiguously, since an order is defined over the set of syntactic objects. Second, there is no ‘removal’ operation concomitant to Merge, since the workspace never contains *e*<X, Y> plus X and Y (in contrast to what happens under set-theoretic assumptions; see Chomsky, 2020, 2021). 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, some assertions in the Minimalist literature with respect to the ‘problems’ posed by {XP, YP} and {X, Y} for purposes of MS must be revised: if a well-defined MS renders these situations unambiguous, then it is not MS that should be reformulated (the opposite conclusion is reached in Milway, 2021; Ke, 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. The phenomena that are supposedly characterised in terms of those ambiguities will thus need to be defined in different terms: note that the same argument form is used in current Minimalism against so-called ‘extensions of Merge’ such as Multidominance (Chomsky, 2019).

Digraphs (can) comply with the current desiderata for syntactic structure like Minimal Yield (Chomsky, 2020b: 18) by not introducing intermediate nodes (since the result of Merge(X, Y) includes no node other than X and Y), and at the same time allows us to bring back argument structure to the core of syntax. Crucially for our argument, Chomsky-Collins sets not provide enough information for their unambiguous translation into digraphs or ordered arrays, which are necessary for search algorithms to apply non-randomly.

The approach presented here, based on the idea that structural descriptions for natural language sentences are digraphs and that search algorithms can be defined in terms of walks through those digraphs, must of course be refined. By way of conclusion, we want to formulate some issues that set the agenda for future research. A crucial one pertains to *sequential* vs. *parallel* search; Ke (2019; 2021) argues for the latter on the grounds that it best implements Chomsky's LA; Preminger (2019), Branan & Erlewine (2021) and us prefer sequential search, for different reasons. The choice depends to a great extent on what the format for structural descriptions is, and what they are designed to capture. Ke (2019: 44, ff., 2021) argues in favour of a parallel breadth-first interpretation of MS on the grounds that (i) depth-first, allegedly, cannot capture some aspects of hierarchical structure by ignoring c-command relations in the definition of a transversal, and (ii) the results of Merge are unordered sets, and breadth-first is more consistent with symmetric relations (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 at any point). Furthermore, if vertices may have an indegree greater than 1, relations should be redefined in terms of *trails* (which allow for a vertex to be walked on more than once), not *paths* (which do not; see also Kayne, 1984: 131-132).

Another issue 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 & 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* [specifiers, complements, adjuncts], *and to then consider the shape and size of the search space separately*

Consider, for instance, the approach to non-configurationality in Hale (1983), Austin & 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 oversees 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 *à la* Ke may be preferable. In a phrase structure system that is more rigidly organised (‘configurational’), depth-first seems to be more convenient (Kremers, 2009). Choosing between depth-first and breadth-first MS may be a source of cross-linguistic variation.

The definition of MS in this paper aims to contribute to the debate of the proper format of syntactic representations, siding with those who follow a graph-theoretic approach. Search algorithms are not exclusive to Minimalist syntax, as relations between nodes in a graph may be modelled using these tools. We propose that if MS is as central to operations in Minimalism as the recent literature suggests, then it may be both theoretically and empirically fruitful to consider graphs as contenders for the format of structural descriptions.

9. References

Arikawa, Koji. 2019. Graph Theory Teaches Us Something About Grammaticality. *Prague bulletin of mathematical linguistics* 112. 55-82.

Austin, Peter & Bresnan, Joan. 1996. Non-configurationality in Australian aboriginal languages. *Natural Language and Linguistic Theory* 14. 215–268.

Bach, Emmon. 1964. *An Introduction to Transformational Grammars*. New York: Holt, Rinehart & Winston.

Bauke, Leah & Blümel, Andreas. 2017. Introduction. In Bauke, Leah & Blümel, Andreas (eds.) *Labels and Roots*, 1-16. Berlin: Walter de Gruyter.

Bell, Sarah. 1983. Advancements and ascensions in Cebuano. In David Perlmutter (ed.) *Studies in Relational Grammar 1*. Chicago: University of Chicago Press. 143-218.

Boeckx, Cedric. 2008. *Aspects of the syntax of agreement.* London: Routledge.

Branan, Kenyon & Erlewine, Michael Yoshitaka. 2021. Locality and (minimal) search. To appear in Grohmann, Kleanthes & Leivada, Evelina (eds.) *Cambridge Handbook of Minimalism*. https://ling.auf.net/lingbuzz/005791

Chomsky, Noam. 1994. Bare phrase structure. *MIT occasional papers in linguistics* 5.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 2001. Derivation by Phase. In Michael Kenstowicz (ed.) *Ken Hale: A Life in Language*. Cambridge, Mass.: MIT Press. 1-52.

Chomsky, Noam. 2004. Beyond Explanatory Adequacy. In Belletti, Adriana (ed.) *Structures and Beyond*, 104–131. Oxford: OUP.

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

Chomsky, Noam. 2015. Problems of Projection: Extensions. In Di Domenico, Elisa & Hamann, Cornelia & Matteini, Simona (eds.) *Structures, Strategies and Beyond: Studies in honour of Adriana Belletti*, 1-16. Amsterdam: John Benjamins.

Chomsky, Noam. 2019. Some Puzzling Foundational Issues: The Reading Program. *Catalan Journal of Linguistics*. 263-285.

Chomsky, Noam. 2020a. The UCLA lectures. https://ling.auf.net/lingbuzz/005485

Chomsky, Noam. 2020b. Minimalism: where we are now, and where we are going. Lecture at 161st meeting of Linguistic Society of Japan. Ms.

Chomsky, Noam & Gallego, Angel & Ott, Dennis. 2019. Generative Grammar and the Faculty of Language: Insights, Questions, and Challenges. *Catalan Journal of Linguistics*. 229-261.

Citko, Barbara. 2008. Missing labels. *Lingua* 118. 907–944.

Collins, Chris. 2002. Eliminating Labels. In Epstein, Samuel & Seely, Daniel (eds.) *Derivation and Explanation in the Minimalist Program*, 42-64. Oxford: Blackwell.

Collins, Chris. 2017. Merge(X, Y) = {X, Y}. In Bauke, Leah & Blümel, Andreas (eds.) *Labels and Roots*, 47-68. Berlin: Walter de Gruyter.

Collins, Chris & Groat, Erich. 2018. Distinguishing Copies and Repetitions. http://ling.auf.net/lingbuzz/003809

Collins, Chris & Stabler, Edward. 2016. A Formalization of Minimalist Syntax. *Syntax* 19(1). 43-78.

Cormen, Thomas & Leiserson, Charles & Rivest, Ronald & Stein, Clifford. 2001. *Introduction to Algorithms*. Cambridge, Mass.: MIT Press.

Dipert, Randall. 1982. Set-Theoretical Representations of Ordered Pairs and Their Adequacy for the Logic of Relations. *Canadian Journal of Philosophy* 12(2). 353-374.

Dowty, David. 2003. The dual analysis of adjuncts/complements in Categorial Grammar. In Ewald Lang, Claudia Maienborn and Cathrine Fabricius-Hansen (eds.) *Modifying Adjuncts*. Berlin: Mouton de Gruyter. 33-66.

Epstein, Samuel. 1998. Un-Principled Syntax and the Derivation of Syntactic Relations. In Epstein, Samuel & Hornstein, Norbert (eds.) *Working Minimalism*, 317-345. Cambridge, Mass.: MIT Press.

Epstein, Samuel & Seely, Daniel. 2006. *Derivations in Minimalism*. Cambridge: CUP.

Epstein, Samuel & Kitahara, Husatsugu & Seely, Daniel. 2013. Structure building that can't be! In Uribe-Etxebarria, Myriam & Valmala, Vidal (eds.) *Ways of structure building*, 253-270. Oxford: OUP.

Epstein, Samuel & Kitahara, Husatsugu & Seely, Daniel. 2015. Labeling by Minimal Search: Implications for Successive-Cyclic A-Movement and the Conception of the Postulate "Phase". In *Explorations in Maximizing Syntactic Minimization*, 201-221. London: Routledge.

Epstein, Samuel & Kitahara, Hisatsugu & Seely, Daniel. 2020. Unifying Labeling under Minimal Search in "Single-" and "Multiple-Specifier" Configurations. *Coyote Papers: Working Papers in Linguistics* 22. 1-11.

Even, Shimon & Even, Guy. 2012. *Graph algorithms*. Cambridge: CUP.

Fong, Sandiway & Ginsburg, Jason. 2019. Towards a Minimalist machine. In Berwick, Robert & Stabler, Edward (eds.) *Minimalist parsing*, 16-38. Oxford: OUP.

Fukui, Naoki & Narita, Hiroki. 2014. Merge, labelling, and projection. In Carnie, Andrew & Sato, Yosuke & Siddiqi, Daniel (eds.) *The Routledge Handbook of Syntax*, 3-23. London: Routledge.

Gärtner, Hans-Martin. 2002. *Generalized transformations and beyond*. Berlin: Akademie Verlag.

Gärtner, Hans-Martin. 2014. Strange Loops: Phrase-Linking Grammar Meets Kaynean Pronominalization. *Linguistische Berichte* 24. 1-13.

Gärtner, Hans-Martin. 2021. 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*. https://doi.org/10.1007/s10849-021-09342-x

Ginsburg, Jason. 2016. Modelling of problems of projection: A non-countercyclic approach. *Glossa* 1: 7. 1-46. http://dx.doi.org/10.5334/gjgl.22

Gorn, Saul. 1967. Handling the growth by definition of mechanical languages. *Proceedings of the April 18-20, 1967, spring joint computer conference*, 213–224. New York: Association for Computing Machinery.

Hale, Kenneth. 1983. Warlpiri and the grammar of non-configurational languages. *Natural Language and Linguistic Theory* 1. 5-47.

Hayashi, Norimasa. 2021. *Labels at the Interfaces: On the Notions and the Consequences of Merge and Contain*. Kyushu University dissertation.

Heim, Irene & Kratzer, Angelika. 1998. *Semantics in generative grammar*. Oxford: Blackwell.

Hopcroft, John & Ullman, Jeffrey. 1969. *Formal Languages and their relation to Automata*. London: Addison-Wesley.

Hornstein, Norbert. 2009. *A theory of syntax: minimal operations and Universal Grammar*. Cambridge: CUP.

Kahane, Sylvain & Lareau, François. 2016. Word ordering as a graph-rewriting process. In Foret, Annie & Morrill, Glyn & Muskens, Reinhard & Osswald, Rainer & Pogodalla, Sylvain (eds.) *Formal Grammar*, 216–239. Berlin: Springer.

Karttunen, Lauri & Kay, Martin. 1985. Structure sharing with binary trees. *Proceedings of the 23rd Association for Computational Linguistics*, 133-136. Chicago: University of Chicago.

Kayne, Richard. 1984. *Connectedness and Binary Branching*. Dordrecht: Foris.

Kayne, Richard. 1994. *The antisymmetry of syntax*. Cambridge, Mass.: MIT Press.

Kayne, Richard. 2018. The place of linear order in the language faculty. Ms. https://as.nyu.edu/content/dam/nyu-as/linguistics/documents/Kayne%200118%20Venice%20The%20Place%20of%20Linear%20Order%20in%20the%20Language%20Faculty.pdf

Ke, Alan. 2019. *The syntax, semantics, and processing of agreement and binding grammatical illusions*. University of Michigan PhD dissertation.

Ke, Alan. 2021. Minimal Search in Agree and Labelling. Ms. https://ling.auf.net/lingbuzz/006123

Keenan, Edward & Bernard Comrie. 1977. Noun phrase accessibility and universal grammar. *Linguistic Inquiry* 8. 63–99.

Kitahara, Hisatsugu. 2020. 'Multiple specifier' configurations revisited. *Reports of the Keio Institute for Cultural and Linguistic Studies* 51. 207-216.

Knuth, Donald. 1998. *The art of computer programming. Vol. 3: Sorting and Searching*. Reading, Mass.: Addison-Wesley.

Komachi, Masayuki & Kitahara, Hisatsugu & Uchibori, Asako & Takita, Kensuke. 2019. Generative procedure revisited. *Reports of the Keio Institute of Cultural and Linguistic Studies* 50. 269-283.

Kremers, Joost. 2009. Recursive linearization. *The Linguistic Review* 26. 135-166.

Krivine, Jean-Louis. 1971. *Introduction to axiomatic set theory*. Dordrecht: Reidel.

Krivochen, Diego Gabriel & Andrea Padovan. 2021. Lexicalised locality: local domains and non-local dependencies in a lexicalised tree adjoining grammar. *Philosophies **2021***, 6(3), 70. https://doi.org/10.3390/philosophies6030070

Krivochen, Diego Gabriel. 2022. *Syntax on the Edge: a graph theoretic analysis of English and Spanish sentence structure*. Ms. Under review.

Kunen, Kenneth. 2013. *Set theory*. London: College Publications.

Kural, Murat. 2005. Tree Traversal and Word Order. *Linguistic Inquiry* 36(3). 367-387.

Langendoen, D. Terence. 2003. Merge. In Carnie, Andrew & Willie, Mary & Harley, Heidi (eds.) *Formal Approaches to Function in Grammar: In Honor of Eloise Jelinek*, 307-318. Amsterdam: John Benjamins.

Larson, Bradley. 2015. Minimal search as a restriction on Merge. *Lingua* 156. 57-69.

McCawley, James. 1968. Concerning the base component of a transformational grammar. *Foundations of Language* 4. 243-269.

McCawley, James. 1982. *Thirty million theories of grammar*. Chicago: University of Chicago Press.

McKinney-Bock, Katherine & Vergnaud, Jean-Roger. 2014. Grafts and beyond: Graph-theoretic syntax. In McKinney-Bock, Katherine & Zubizarreta, María Luisa (eds.) *Primitive Elements of Grammatical Theory*, 207-236. London: Routledge.

Medeiros, David P. 2021. Tree transversal grammar. Ms. University of Arizona.

Milway, Daniel. 2021. Agree as derivational operation: Its definition and discontents. Ms. https://ling.auf.net/lingbuzz/005842

Moro, Andrea. 2000. *Dynamic antisymmetry*. Cambridge, Mass.: MIT Press.

Osborne, Timothy & Putnam, Michael & Groß, Thomas. 2011 Bare phrase structure, label-less trees, and specifier-less syntax: Is Minimalism becoming a dependency grammar? *The Linguistic Review* 28. 315-364.

Ott, Dennis. 2012. *Local Instability*. Berlin: de Gruyter

Ott, Dennis. 2015. Symmetric Merge and Local Instability: Evidence from Split Topics. *Syntax* 18(2). 269-303.

Postal, Paul. 2010. *Edge-Based Clausal Syntax*. Cambridge, Mass.: MIT Press.

Preminger, Omer. 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

Romeijn, Edwin. 2008. Random Search Methods. In Floudas, Christodoulos & Pardalos, Panos (eds.) *Encyclopaedia of Optimization*. Boston: Springer. https://doi.org/10.1007/978-0-387-74759-0_556

Sarkar, Anoop & Joshi, Aravind. 1997. Handling coordination in a tree adjoining grammar. Technical report, University of Pennsylvania. https://www2.cs.sfu.ca/~anoop/papers/pdf/tag-coordination.pdf

Schmerlin, Susan F. 2018. *Sound and Grammar: towards a Neo-Sapirian theory of language*. London: Brill.

Sedgewick, Robert & Wayne, Kevin. 2014. *Algorithms*. NY: Addison-Wesley.

Sobin, Nicholas. 2014. Th/Ex agreement and case in expletive sentences. *Syntax* 17. 385‑416.

Sobin, Nicholas. 2020. Directing syntactic traffic. *Syntax* 23. 214-274.

Stabler, Edward. 2011. Computational perspectives on minimalism. In Boeckx, Cedric (ed.) *Oxford handbook of minimalism,* 617–641. Oxford: Oxford University Press.

Stepanov, Arthut. 2001. Late adjunction and Minimalist phrase structure. *Syntax* 4(2). 94-125.

Sternefeld, Wolfgang. 1998. Connectivity effects in pseudo-cleft sentences. *ZAS Papers in Linguistics* 10. 146-162.

Uriagereka, Juan. 2012. *Spell-Out and the Minimalist Program*. Oxford: OUP.

Van Gelderen, Elly. 2021. *Third factors in language variation and change*. Cambridge: CUP.

Van Steen, Maarten. 2010. *Graph Theory and Complex Networks: An Introduction*. Available at https://www.distributed-systems.net/index.php/books/gtcn/gtcn/

Vercauteren, Aleksandra. 2017. Features and labeling: Label-driven movement. In Bauke, Leah & Blümel, Andreas (eds.) *Labels and Roots*, 69-90. Berlin: Walter de Gruyter.

Wilson, Robin. 1996. *Introduction to Graph Theory*. London: Adison Wesley.

Wurmbrand, Susi. 2014. The Merge Condition: A syntactic approach to selection. In Kosta, Peter & Franks, Steven & Radeva-Bork, Teodora & Schürcks, Lilia (eds.) *Minimalism and Beyond: Radicalizing the interfaces*, 130-167. Amsterdam: John Benjamins.

Zwart, Jan-Wouter. 2009. Prospects for top-down derivations. *Catalan Journal of Linguistics 9.* 161-187.

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

Zwicky, Arnold & Isard, Stephen. 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, Erik. 2022. Phase-constrained obligatory late adjunction. *Syntax* 25(1). 84-121.

Acknowledgments: to be added.