Alan Hezao Ke

Michigan State University

Department of Linguistics, Languages and Cultures

619 Red Cedar Road B-230 Wells Hall

East Lansing, MI 48824, USA

kehezao@msu.edu

Minimal Search in Agree and Labeling

Abstract. This paper develops a formal definition of Minimal Search to evaluate the idea that Agree and labeling could be reduced to Minimal Search. Different aspects of the search algorithm in Minimal Search, i.e., breadth-first vs. depth-first search, parallel vs. serial search, global vs. modular search are compared, and reasons for choosing between each of these pairs are given based on detailed examinations of their theoretical and empirical consequences. This paper argues, based on the formal definition of Minimal Search, that Agree and labeling can only be partially unified by Minimal Search: the search algorithms in Agree and labeling can be unified by Minimal Search, but the values of the search targets and search domains are determined by Agree and labeling independently. This paper (re)defines Agree and labeling based on Minimal Search to capture both the similarities and differences between these two operations, and consequential solutions to several theoretical and empirical difficulties faced by the standard approach, including problems associated with the $<\phi, \phi>$ label and the Invisibility of Lower Copies Hypothesis, are discussed.

Keywords: Minimal Search, Agree, labeling theory, 3rd factor, search algorithm

1. Introduction

An influential idea in recent Minimalist theory is that Agree and labeling are simply Minimal Search. Minimal Search has been proposed to be a 3rd factor principle (Chomsky 2005), as it is a principle of efficient computation which language, a computational system, may very likely observe.

This assumption has an important conceptual motivation. Currently, researchers seem to agree that the syntactic component of the language model includes three operations: Merge, Agree and labeling. If Agree and labeling can be removed from the system and be replaced by the 3rd factor Minimal Search without a significant empirical loss, the system would be considerably simplified.

Chomsky (2013) has made this assumption explicit. Chomsky (2013) writes, "The simplest assumption is that LA (labeling algorithm) is just Minimal Search, presumably appropriating a third-factor principle, as in Agree and other operations." (p. 43). In Chomsky (2015a), he continues this discussion, "Optimally, projection should be reducible to a labeling algorithm LA, a special case of Minimal Search (like Agree), which in turn falls under MC (minimal computation)." Chomsky therefore hypothesizes that an optimal system would have Agree and the labeling algorithm derived from a more general, independently motivated third-factor principle, i.e., minimal computation. This conjecture, if successful, could unify the labeling algorithm and Agree by resorting to an arguably third-factor operation, Minimal Search. Consequently, since Minimal Search as a third factor is freely and universally available, it might be the case that labeling algorithm and Agree are also freely available (Chomsky 2005).

A serious implementation of the above conjecture requires a formal definition of Minimal Search. However, to the best of my knowledge, Minimal Search has not been formally defined (to cover both labeling and Agree).[1] The concept "Minimal Search" is an interaction of two other important concepts: minimality (of computation) and a search algorithm. The idea of minimality in Agree has been around for a while (as we will see in Chomsky's (2000 definition of Agree and relevant discussion below), but no formal search algorithm has been provided so far to implement it. I thus provide a definition of Minimal Search in the next section, based on which it is clear that labeling and Agree cannot be completely reduced to Minimal Search; they can only be partially unified by Minimal Search. On the one hand, the search algorithm in labeling and

[1] EKS (Epstein, Kitahara and Seely) defined a version of Minimal Search for labeling, and Jihu Kim (p.c.) provided a definition of Minimal Search for labeling which has the potential to be extended to Agree. The definition proposed here couches in a search algorithm in computer science, which distinguishes it from EKS and Kim's definitions.

Agree can be captured by a definition of Minimal Search that I will argue for in the rest of this paper. On the other hand, labeling and Agree are essentially different in their search target and search domain; they are thus needed for independent reasons and they serve distinct purposes. I will then discuss the implications of this particular definition with respect to Agree, labeling, and other empirical domains, and demonstrate how the modified theory of labeling and Agree addresses a few problematic cases for the standard approach. The last section concludes the paper.

2. Defining Minimal Search

2.1 Insights from previous studies

As previously mentioned, Chomsky (2013) does not give a formal definition of Minimal Search; however, he does illustrate how Minimal Search works in labeling and Agree by examples. Let us consider some examples for labeling as in (1a, b).

(1) a. $\{_\alpha$ kick, {the ball}}

b. $\{_\beta$ $\{_\delta$ a, {little boy}}, $\{_\kappa$ is, {kicking the ball}}}

It seems that for the labeling of the set α in (1a), Minimal Search needs to look into α, and only the head member *kick* is returned. In this case, Minimal Search seems to look for a head and ignore *the ball*, the set member of α. This is reasonable because heads can be considered bundles of features, and sets do not carry features,[2] if we do not assume feature projection or percolation, following Chomsky (2013) and Chomsky (2015b). There is thus a natural distinction between heads and sets. In other words, Minimal Search in this case could search for any features. As long as it finds a feature, it returns the head bearing that feature, and then the search is immediately terminated. If Minimal Search is not terminated at this point and can continue to look into {the ball}, *the* would be returned as the label in addition to *kick*, which is not desirable. Therefore, this establishes a very important principle in Minimal Search, which is previously implicitly assumed but unnoted, as far as I know: Minimal Search terminates whenever a target is found. I will return to this principle later.

For (1b), Minimal Search needs to find a label for the set β, and thus looks into it. It cannot find any feature in β; instead, two sets are found. As a result, Minimal Search looks into the two sets, δ and κ. The

[2] I should thank Hisatsugu Kitahara for bringing this to my attention in one of our informal discussions.

heads it returns for these two sets are *a* and *is*. Note that this result is only possible if Minimal Search looks into δ and κ simultaneously. Otherwise, since Minimal Search terminates as soon as a target is returned (as pointed out in the previous paragraph), if Minimal Search serially searches into δ and then κ, only *a* will be returned. If, on the other hand, Minimal Search serially searches into κ and then δ, only *is* will be returned. Under serial search, there is no chance for *a* and *is* to be returned simultaneously. This is by no means a welcome result. Therefore, we must adopt a search algorithm that can look into the two sets simultaneously (i.e., by applying parallel search, not serial search), as I will argue in more details later.

Furthermore, according to Chomsky (2013), the features in agreement between *a* and *is*, that is, the ϕ-feature pair <3SG, 3SG>, will be taken as the label of β. Again, the search needs to be terminated once the heads (*a* and *is*) are found and their prominent agreeing ϕ-features are extracted; otherwise, the head of *little boy* (whatever it is) and that of *kicking the ball* will also be returned as labels, which is not what we want.[3]

Note that based on the labeling operation for structures such as (1b), Chomsky (2013, 2015b) could potentially derive the EPP (the Extended Projection Principle), which requires every (finite) TP to have a subject in its specifier position, potentially with additional assumptions.[4] This is because when the *v*P-internal subject is at its base-generated position, the *v*P cannot be properly labeled given that the *v*P-internal subject and the *v* head do not share agreement features, according to Chomsky (2013, 2015b). A solution to this problem is to move the *v*P-internal subject to a higher position. Since T agrees with the *v*P-internal

[3]This strongly suggests that relevant ϕ-features must be all accessible on D; otherwise Minimal Search would need to search into the complement of D to find all the phi-features. If we assume DP is a phase and the phase head complement is transferred at the point when the phase head enters in the derivation (Chomsky 2000), *little boy* would have been transferred at the derivational step in (1b) and thus would be no longer accessible to Minimal Search. In this circumstance, we will still have to assume that all ϕ-features must be available on D, since D is the only accessible head in DP. By contrast, the standard theory of transfer (Chomsky 2008) does not remove the *v* head of *kicking the ball* and it should be available for search: this will cause unexpected problems to the labeling of (1b) if search is not terminated once a target is returned.

[4]Chomsky (2015b) works out the derivation of the EPP for finite TPs, as he assumes that the T head in English is too weak to be the label of a TP, and therefore requires a SpecTP to be in an agreement relation with T, so that the features in agreement (the <ϕ, ϕ> pair) can be the label. A non-finite T then does not require a SpecTP because they cannot be in an agreement relation.

subject, an agreement relation which will be discussed in the next paragraph, the *v*P-internal subject can move to the specifier of T′ (={T, *v*P}), where a legitimate label, i.e., $<\phi, \phi>$, is possible.

Now the question is how T can agree with the *v*P-internal subject and how Minimal Search is involved in this agreement. This is accomplished by the operation Agree. Below is Chomsky's (2000) definition of Agree, with minor modifications due to Chomsky (2004)[5]:

(2) *Definition of Agree* (Chomsky 2000)

Agree is a syntactic operation taking place between a probe P and a goal G in the domain of P, D(P), between which a Matching relation holds.

a. Matching is identity of feature attributes;

b. D(P) is the sister of P;

c. Locality reduces to "closest c-command";

(3) *Definition of closest c-command*

A matching feature G is closest to P if there is no G′ in D(P) matching P s.t. G is in D(G′).

Three aspects of the above definition of Agree are worth noting: (i) Agree requires a probe, which will be the search target in my definition of Minimal Search below; (ii) the probe looks into a domain, which will be the search domain in my formalization; (iii) Agree observes a locality constraint, namely, closest c-command, which is likely what "minimal" means primarily in the term "Minimal Search."

To illustrate Chomsky's definition of Agree, let us use (4) as an example. In addition, a direct comparison between Agree in (4) and labeling in (1) may help us find out what exactly the mechanism is that is shared between labeling and Agree. (4a) is the derivation step before the T head, *be*, agrees with the D head of the *v*P-internal subject *this student* in (4b).[6]

[5] Chomsky (2000) considers matching as feature identity. However, Chomsky (2004) points out correctly that the matching between an uninterpretable feature and a corresponding interpretable feature is not feature identity but feature non-distinctness. I revised this part of definition as "matching is identity of feature attributes," excluding feature values in matching, to address the problem of the original definition of Agree.

[6] I assume that all relevant ϕ-features are accessible on the D head, although not all ϕ-features are base-generated on D, following Sigurðsson (2017). See Footnote 2.1 for another reason why it is necessary for ϕ-features to be accessible on D from the perspective of the labeling algorithm.

(4) a. [$_{T'}$ be [$_{vP}$ [$_{DP}$ this student]$_j$ [$_{v'}$ writing the dissertation]]]

b. [$_{TP}$ [$_{DP}$ this student]$_j$ is [$_{vP}$ t$_j$ [$_{v'}$ writing the dissertation]]]

After *be* merges into the derivation, as shown in (4a), a Minimal Search is initiated, and it looks into the *v*P set for a head bearing valued ϕ-features corresponding to those unvalued ones on *be*. No heads, but two sets, i.e., the DP and v', are found. So Minimal Search has to continue to search into these two sets. Both D and *v* are found. I assume *v* does not bear matching ϕ-features, either because ϕ-features in English are base-generated at V or feature inheritance from *v* to V has been completed at this derivation step (Richards 2007). Finally, the head *the* in the DP is returned, and Minimal Search terminates.

Let us now summarize the roles Minimal Search plays in labeling and Agree. First, it looks into a set for some specific features (Agree) or any kind of features (labeling). Second, it applies iteratively to every available set, and it is terminated whenever a target is returned. To state it more formally, Minimal Search is a search algorithm that looks into a certain *search domain* (sets) for certain *search target* (features). The algorithm will find a syntactic object X before the syntactic objects c-commanded by X, as implied by Chomsky's definition of Agree in (2) which relies on the concept of closest c-command. The search is minimal because it is terminated as soon as a target is returned.[7]

2.2 *The proposal*

When considering what Minimal Search in Syntax would be like I always think of visual search. Suppose we are looking for a specific type of animal in box α. This is an instance of guided visual search where search is guided by features (Wolfe 2010). As shown in Figure 1, two boxes, β and γ, are contained inside α, and then δ is embedded in β and ρ in γ. Imagine that all the boxes have an opaque cover which must be removed before the searcher can see inside.

What algorithm is an efficient search algorithm that is appropriate for visual search in this situation? Well, even before we look for an appropriate algorithm, we must first know what things/features we are searching for. That is, we must have a search target in mind. Suppose our search target is any cat. We then also need to determine the domain(s) we search into. The selection of search domain interacts with

[7] Similar conceptualization of minimality has been previously formalized in Agree and Move, with a representational flavor, such as closest c-command, Minimal Link Condition (Chomsky 1995), and Relativized Minimality (Rizzi 1990). See Branan and Erlewine (to appear) for relevant discussion.

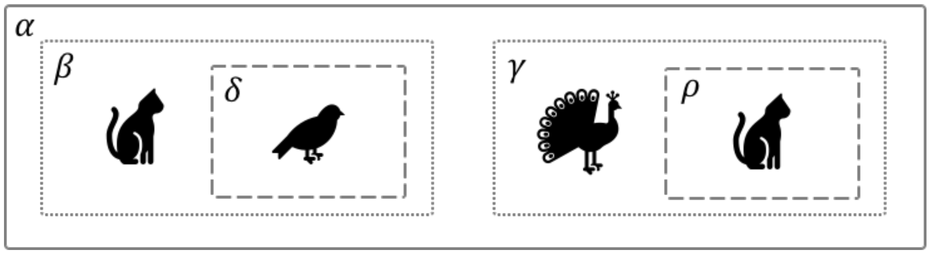

Figure 1: Visual Minimal Search

the search algorithm. If our search is serial, after searching into α, we may first search into β and then γ, or the other way around. However, if we can distribute our cognitive resources for different searches and thus be able to conduct searches in parallel, then an alternative algorithm is that we open box β and γ simultaneously.[8]

There is a problem. In β you find a cat and in γ a peacock. Do you terminate search once you find the cat in β, or do you continue to search for a cat in ρ? This depends on the nature of your search. For serial search, there is probably no reason to continue the search once a target is returned. For parallel search, if the results of search are sent to your central processing systems, then you are aware that the search target has been found. You will terminate your search as you have achieved your search goal. By contrast, if each search in your parallel search (e.g., the search into β vs. the search into γ) is implemented in Fodorian modules that are informationally encapsulated and are of limited central access (Fodor 1983), then you are not aware of the results of searches that are carried out in parallel, and thus you will search into ρ further. Let us call the former search algorithm the global search algorithm, and the latter the modular search algorithm. The one I will adopt in the definition of Minimal search below is the global search algorithm. In other words, Minimal Search in Syntax terminates search whenever a higher syntactic object is returned, potentially consistent with what Chomsky states in Chomsky (2019): "[D]on't use deep search if minimal search already works."

If the modular search algorithm is adopted, we would need to search into every box in Figure 1 if there are more embedding boxes. That is, more boxes organized in a hierarchical structure will add to the computational burden and will lead to a massive search. Note that massive search is conducted even after

[8] Compared to parallel search which generally finds the target faster, serial search in this instance may reduce the number of boxes that we need to open if we happen to start the search by looking into β and find a cat right away. However, this accidental nature of efficiency should not be in principle a basis of an argument for serial search. I will argue later that Syntax by nature may favor parallel search over serial search.

the target has been found. This is intuitively not reasonable for visual search given Figure 1, and it is surely not computationally efficient, incompatible with the idea that Minimal Search is a 3rd factor computational principle. Linguistic evidence against modular search will be provided in Section 4.

Therefore, we want Minimal Search in the visual domain to be terminated as soon as the search target is returned, and this applies to both parallel and serial search. This is consistent with what we have seen above regarding the search algorithm in labeling and Agree. In addition, the visual search example also confirms the necessity and importance of search target and search domain.

With what we have learned from Minimal Search in the visual domain, let us now turn to a definition of Minimal Search in Syntax. As shown in (5), the current definition of Minimal Search includes a search algorithm that applies iteratively to a search domain (SD) to look for a search target (ST).

(5) *A formal definition of Minimal Search*

MS = <SA, SD, ST>, where MS = Minimal Search, SA = search algorithm, SD = search domain (the domain that SA operates on), ST = search target (the features that SA looks for).

Search Algorithm (SA):

a. Given SD and ST, matching against every head member of SD to find ST.

b. If ST is found, return the heads bearing ST and go to Step (c); Otherwise, get the set members of SD and store them as a list L.

   i. If L is empty, search fails and go to Step (c); otherwise

   ii. assign each of the sets in L as a new SD and go to Step (a) for all these new SDs in parallel.

c. Terminate search.

The description of the search algorithm in (5) can be spelled out more explicitly, e.g., in terms of pseudo-codes, which I include in the appendix. Parallel instead of serial search is adopted in the definition, the reason of which will be discussed in detail in Section 2.4. Such an algorithm can be implemented in a programming language that allows parallel operations.

The definition uses lists instead of sets to store the new SDs and the returned heads. Lists are usually associated with linear order; however, I do not assume precedence relations between the SDs or the returned heads in a list.

2.3 An example

As an illustration of the Minimal Search as just defined, let us walk through an example. As shown below in Figure 2, a Minimal Search is initiated to search for an ST = feature [F], in the SD = set α.

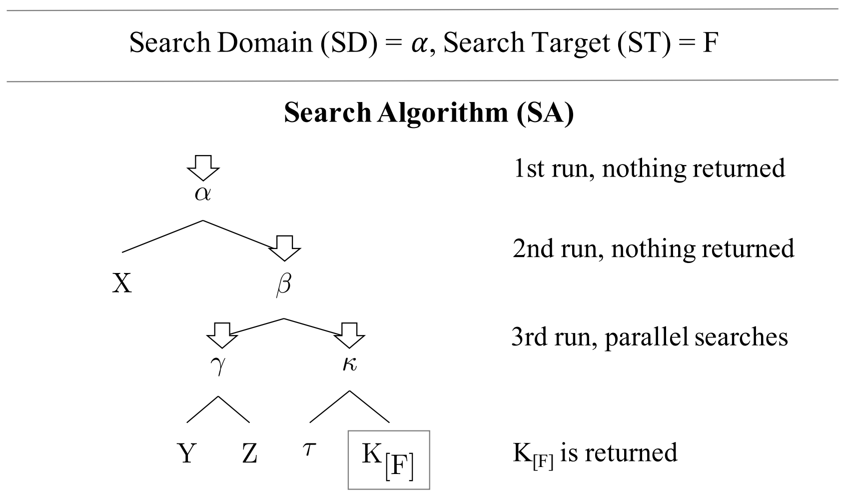

Figure 2: An illustration of Minimal Search

Three runs of search are conducted before the target is found. In the first run, Minimal Search looks into α. X, the head member of α, is returned by the first run of search. However, since it does not bear the SD [F], and its co-member is a set, the search is not terminated. The set member of α, that is, the set β, is then assigned as a new SD.[9] This is the second run. The second run of search finds only set members, i.e., γ and κ, and these two set members are stored as a list L. The two sets in L are then assigned as the new SDs for the third run of search, and two independent Minimal Searches are initiated in parallel. The head $K_{[F]}$ is found inside the set κ as it bears the ST. $K_{[F]}$ is finally returned by the search algorithm. The search over the set γ fails and is terminated without returning a result.

It is worth noting that the minimality in the (breadth-first) search is naturally (and partially) captured by *storing sets as a list L* in the definition (5b), not by, for instance, keeping track of/counting the levels of sets that Minimal Search looks into. All the sets which are at the same level will be stored in the same list, and new Minimal Searches are conducted to look into all these sets simultaneously in parallel, as desired.[10]

[9] Note that the Minimal Search defined in (5) does not allow a single Minimal Search applying to different sets. In order to search into a different set, a new Minimal Search must be initiated taking that set as a new SD.

[10] Natural language may enforce restrictions on the amount of items that are stored in a search list. See Footnote 2.4 for a comment regarding the search list.

2.4 Why breadth-first not depth-first, parallel not serial?

There are two popular search algorithms in graph or tree search in computer science, breadth-first and depth-first search. In the case of traversing a search tree, a breadth-first algorithm explores all the daughters of a starting node before it searches any deeper. In Figure 3, breadth-first search visits the daughters of the starting node N1 before moving on to any of the daughter nodes of N2 or any node deeper. After all the daughters of Node 1 are visited, it then continues to traverse all the daughter nodes of Node 2 and Node 3. So on and so forth, until all nodes in the search tree are visited and the algorithm is terminated.

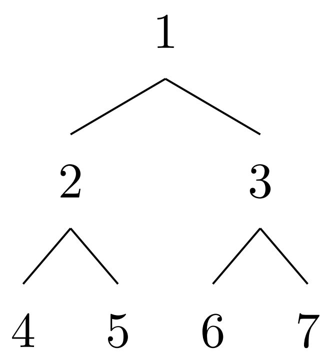

Figure 3: Breadth-first search

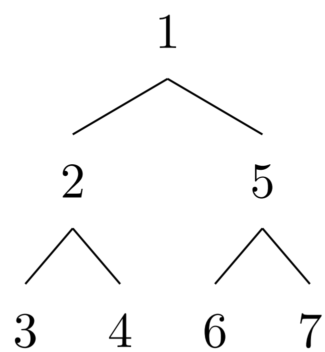

Figure 4: Depth-first search

On the contrary, a depth-first algorithm exhaustively searches down along certain nodes before back-tracking. Take the tree in Figure 4 for an example. Starting from Node 1, depth-first search first visits a certain daughter of Node 1, e.g., Node 2, and then a daughter node of Node 2, e.g., Node 3, before it visits all the daughters of Node 2 and 3. Depth-first search reaches a terminal node (leaf node) there, so it back-tracks to another daughter of Node 2, i.e., Node 4; then another daughter of Node 2, i.e., Node 5; and then a daughter of Node 5; so on and so forth, until it exhausts all the nodes dominated by Node 1.

Given an SD $\alpha = \{\beta, \gamma\}$, where Greek letters represent sets, a breadth-first search explores all the members of α before visiting the members of β and γ. On the other hand, a depth-first search goes to a member of α, for instance, β, and visits the members of β before exploring γ. The search algorithm in the definition of Minimal Search (5) is breadth-first (a parallel version).

Why do we adopt breadth-first search rather than depth-first search? Depending on the data structure, both the breadth-first and depth-first can be computationally efficient. However, two considerations make a breadth-first algorithm a preferable option for labeling and Agree. First, the hierarchical nature of syntactic structures is one of the most important aspect of language and linguistic analysis, and the c-command relations are a most crucial instance of the hierarchical nature of language. Minimal Search must be compatible with this fundamental property of human language, and it must satisfy relevant restrictions. Breadth-first search respects the c-command relations as it looks into all nodes that are at a higher level before visiting any nodes at lower levels. By contrast, depth-first search does not exhaust all the items at a specific level first before visiting lower levels. The searching process in depth-first search cannot always track the hierarchical order of the levels, as there must be times that the algorithm traverse back from a lower level to a higher level. Indeed, depth-first search, to some extent, ignores some critical aspects of hierarchical structure. In particular, depth-first search cannot always capture the asymmetric c-command relations (it primarily captures the containment relations): it may reach a c-commandee before accessing its c-commanders.[11] For instance, in Figure 4, Node 3 and 4 are visited before their c-commander Node 5 is visited. On the other hand, if Node 5 is explored before Node 2, Node 6 and 7 will be visited before their c-commander Node 2 (and there is no principled reason to force the algorithm to visit Node 2 before Node 5 and vise versa).

In principle, Minimal Search would need to return as early as possible a node containing the search target if the node is structurally closest to the starting node for search; structural closeness is defined in terms of c-command: If A c-commands both B and C, B is closer to A than C iff B c-commands C. A breadth-first algorithm thus serves exactly this purpose. Due to the fact that depth-first search will visit a c-commandee before visiting its c-commanders, closest-c-command as found in Chomsky's definition of Agree in (2) cannot always be faithfully captured by Minimal Search if we adopt depth-first search.

In addition, importantly, depth-first search in fact also cannot give desired results with regard to labeling. In the labeling of the {XP, YP} construction, the label will never be {X, Y} if depth-first search algorithm is adopted without further modifications, as it will be either that XP and all heads inside XP are visited before YP is explored, or that YP and all heads inside YP are explored before XP is visited. Neither of these two

[11] C-command is defined as below:

(6) *Definition of c-command*

α c-commands β iff

a. neither α nor β dominates the other, and

b. the first branching node that dominates α also dominates β.

choices naturally give us {X, Y} or {ϕ, ϕ}. Making it worse, a depth-first Minimal Search may first visit XP before visiting H in the {H, XP} construction and returns X instead of H. The result is that the label for {H, XP} would be not H but X! This is certainly not what we are looking for.

We have argued that breath-first search rather than depth-first search is preferred for the Minimal Search algorithm. Why do we then employ a *parallel* breadth-first instead of a *serial* breadth-first search algorithm? The reason is that once set theory is adopted for structure building, linear order becomes a peripheral property. This is because the members of a set are standardly assumed to be symmetric with regard to their linear order. Linear order is not specified until syntactic objects are linearized at the SM interface. As a consequence, there is no principled reason to search into a specific member of a set before another set. One may argue that the algorithm can randomly search into any one of the set members. However, this will not give us the right result: in terms of {XP, YP} constructions, if XP or YP is randomly searched into first, either X *or* Y, never both X *and* Y, will be returned, for I have argued in length that in general the algorithm must terminate once it encounters a target. Consequentially, we will never be able to find X and Y *simultaneously* in the case of {XP, YP} without adding further (likely unjustified) complications to the search algorithm.

Therefore, the non-ordering nature of set members favors the parallel breadth-first search algorithm, both theoretically and empirically, since such an algorithm looks into both set members simultaneously in parallel, without making a distinction between them; and a natural result of this parallel search is that X and Y in the {XP, YP} construction can be returned simultaneously.

Finally, the Syntax component has no way of determining which member of a set is more important than the other; therefore the algorithm cannot determine which member to search into first based on their status of significance. The head-hood information, which probably gives some sense of "importance" to a certain member of a set, is not available until labels are computed at the time when syntactic objects are sent to CI for interpretation (see below for discussion of the timing of labeling).[12]

2.5 *Minimal search-based Agree and labeling*

Above, I gave a formal definition of Minimal Search. In this section, we will give a Minimal Search-based definition for labeling and another for Agree. It will become clear that Agree and labeling can be partially but not completely reduced to Minimal Search.

As shown in (5), Minimal Search involves three components, the SD, ST, and SA. The SA is invariant and identical for all cases of Minimal Search, no matter whether it is for Agree or labeling. What distinguishes Agree from labeling is the SD and ST. The SD and ST being independently assigned by different operations is a unique feature of the current definition of Minimal Search (see Ke (2019) Chapter 2 for an extended exploration of the empirical implications of this feature). Below I would like to explain how the SD and ST are determined in Agree and labeling.

2.5.1 *How to determine SD and ST?*

The SDs of Minimal Search for labeling and Agree are determined in accordance with the purpose of these two operations. Agree connects two heads, one carrying unvalued features and the other carrying matching valued features. In order for two heads in an agreement relation to be connected to each other syntactically, they must form a syntactic relation at a certain stage of derivation. That is, they must be contained in the same root set or under a single node in a syntactic tree at a certain stage of derivation. Suppose a head X with

[12] Readers who are familiar with search algorithms in computer science may think of computation burden as a drawback of the parallel breadth-first search algorithm, mainly due to the storage of multiple search domains in a list and relevantly the conduction of multiple searches in parallel. However, Minimal Search by definition is a third factor, and therefore it is not specific to language and it potentially does not take resources from the Syntax module but instead uses general resources. Of course, general attentional resources are also restricted and limited, but Minimal Search, like many other information processors in the brain, can be carried out subconsciously without resorting to attention or even memory. Indeed, just like parallel visual search which is not dramatically slowed down by the increase of the amount of objects in the search array (Woodman et al. 2001; Horowitz and Wolfe 1998), Minimal Search may be not so strictly restricted by the amount of search domains and the amount of searches in parallel. However, I will also not go to the other extreme that Minimal Search is completely not restricted on the capacity of its search lists. In fact, language has natural mechanisms to reduce the number of search domains in the list, among which phasal transfer or Phase Impenetrability Condition (PIC) being a critical and well accepted candidate.

an unvalued feature uF, represented as $X_{[uF]}$, needs to be valued by another head Y with a corresponding valued feature vF, represented as $Y_{[vF]}$, there are three logically possible situations for X to be connected to Y at a certain point of the derivation, with regard to the c-command relations between X and Y:

(7) a. $X_{[uF]}$ merges with $Y_{[vF]}$ or a set containing $Y_{[vF]}$: $X_{[uF]}$ c-commands $Y_{[vF]}$;

b. $Y_{[vF]}$ merges with $X_{[uF]}$ or a set containing $X_{[uF]}$: $Y_{[vF]}$ c-commands $X_{[uF]}$;

c. A set α containing $Y_{[vF]}$ merges with a set β containing $X_{[uF]}$. Neither $X_{[uF]}$ nor $Y_{[vF]}$ c-commands the other.

All three options are utilized in natural language. Downward (long-distance) Agree, e.g., where the matrix verb agrees with an embedded nominal for noun classes in Tsez (Polinsky and Potsdam 2001) is an instance of (7a). Upward Agree in negative concord (Zeijlstra 2012), inflection doubling/parasitic participles (Wurmbrand 2012) and multiple case licensing (Hiraiwa 2001) are instances of (7b). And Ke (2019, ch. 3) argues that reflexive binding is an instance of (7c). We will set (7c) aside for the discussion below.

Importantly, with (7a), the Minimal Search initiated to value $X_{[uF]}$ needs to search into the c-command domain of X. That is, the SD of the Minimal Search in this case must be the sister, or co-member, of X. However, a problem immediately arises with regard to (7b), where the valuee bearing the unvalued feature, $X_{[uF]}$, is c-commanded by the valuer bearing the corresponding valued feature, $Y_{[vF]}$. Notice that there is no way for the specific formalization of Minimal Search in (5) to search "upward." Minimal search as defined in (5) always searches "downward" into its SD. This problem can be resolved with a search domain assigned as a set that contains the c-commanding valuer. Due to space limitation, we will not discuss this case here (see Ke (2019) Chapter 2 for a proposal that formalizes upward Agree as a special case of downward Minimal Search).

The ST for Agree is simply the unvalued features on the head that initiates the Minimal Search. For instance, when the purpose of a Minimal Search is to find a head with valued features matching the unvalued feature attributes [Person: , Num:] on a T head, then [Person: , Num:] are taken as the ST.[13]

On the other hand, since the labeling algorithm is meant to provide a label for a given set, the SD will be that set. For instance, if the set {kick, {the, ball}} is the set that needs to be labeled, this set is the SD of the Minimal Search for the labeling purpose.

The ST of Minimal Search in labeling is more complicated. Take (8) as an example. In the case of (8a), the ST would be any feature or any head, not a particular head or feature.

(8) a. $\{_{\alpha}$ H, XP$\}$: $\{_{\alpha}$ kick, {the ball}}

b. $\{_{\beta}$ XP, YP$\}$ = $\{_{\beta}$ $\{_{\delta}$ X, KP$\}$, $\{_{\kappa}$ Y, ZP$\}\}$:

$\{_{\beta}$ $\{_{\delta}$ a, {little boy}}, $\{_{\kappa}$ is, {kicking the ball}}}

However, for (8b), if we want to obtain a $<\phi,\phi>$ pair by means of Minimal Search, then the ST must be ϕ-features as well, a set of specific features. This assumes that the labeling algorithms, STs specifically, for (8a) and (8b) should be different. A unified labeling algorithm, by contrast, has to either (i) set the ST to any feature/head (not particular features), and therefore will return only heads for both (8a) and (8b); or (ii) set the ST to some specific features such as ϕ-features. A problem for (ii) is that when features other than ϕ-features are in agreement, e.g., in an interrogative CP, the features in agreement, i.e., the <Q, Q> pair, cannot be returned as a label, unless the ST of the Minimal Search for an interrogative CP is set to Q. Furthermore, it is important to point out that even if we want to set different STs for different searches depending on the type of prominent features in {XP, YP} constructions, we have a look-ahead problem: how do we know what type of prominent features are relevant and thus set these features as STs before searching into the search domain? In order to avoid the look-ahead problem and have a unified labeling algorithm for all the cases we discussed (we surely do!), the ST should be set to any head/any feature in all cases, and thus the label for (8b) must be <a, is> rather than a $<\phi,\phi>$ pair for an {XP, YP} construction such as (8b) (see more discussion below in Section 3.3).

[13]I leave open for now the question whether all the unvalued features should be combined as a single ST and initiate a single Minimal Search, or each of them should initiate an independent Minimal Search. For most cases, these two options do not make a distinction. For exceptional cases where we may need to split the unvalued features, please see e.g., Béjar and Rezac (2003) and Preminger (2009). The current definition of Minimal Search is in principle compatible with the split analysis. A mixed analysis with a combined ST in some cases but with split STs in some other cases may also be possible.

To summarize, I have shown briefly that the *search algorithm* in Agree and labeling can be unified by the search algorithm of Minimal Search. However, the *values* of the SD and ST of Minimal Search are independently determined by Agree and labeling, and thus the SD and ST values for Agree are different from those for labeling. Minimal search for Agree and labeling thus serve distinct purposes. Details of this argument will be presented later in Section 3.

2.5.2 *MS-Agree and MS-labeling*

We are now ready to give definitions of Minimal Search-based Agree and Minimal Search-based labeling, which I will henceforth abbreviate as MS-Agree and MS-labeling:

(9) Definition: Minimal Search-based Agree (MS-Agree)

Agree = Minimal Search + Valuation

i. **Minimal search**

a. Input to Minimal Search:

- SD = co-member of $H_{[uF]}$
- ST = feature attributes of uFs (e.g., [Person: , Num:])

b. Output of MS: heads bearing the ST

ii. **Valuation**

- Trigger: the head bearing uFs
- Valuation: copy the value of the ST in the Minimal Search output to the corresponding uFs in the trigger

Below, I give an example to illustrate the implementation of MS-Agree. *Be* in (10) can agree with *a man*, or more accurately, the determiner *a*, the head of *a man*. This is a case of long-distance agreement. The steps of MS-Agree are shown in (11).

(10) Example: *(there) be likely to be a man outside*

(11) a. SD = {likely, {to be a man outside}}

b. ST = ϕ-feature attributes of the unvalued ϕ-features on *be*

c. Search into the SD, the target is not found

d. Set {to, {be a man outside}} as the SD

e. Search into the SD, the target is not found

f. Set {be, {a man outside}} as the SD

g. Search into the SD, the target is not found

h. Set {{a man}, outside} as the SD

i. Search into the SD, the target is not found

j. Set {a, {man}} as the SD

k. Search into the SD, the target is found on *a*, return *a*

l. Valuation: copy feature values to *be* from the matching features on *a*

Furthermore, a formal definition for MS-labeling is presented below:

(12) Definition: Minimal Search-based labeling (MS-labeling)

The output of Minimal Search is taken as the label of the set to be labeled.

i. Input to Minimal Search:

- SD = the set to be labeled
- ST = any feature/head[14]

ii. Output of Minimal Search:

- Heads

The labeling algorithm is straightforward. Let us use the examples in (1), repeated below in (13), as an illustration.

(13) a. $\{_{\alpha}$ kick, {the ball}} $\longrightarrow$ label = kick

b. $\{_{\beta}$ $\{_{\delta}$ a, {little boy}}, $\{_{\kappa}$ is, {kicking the ball}}} $\longrightarrow$ label = {a, is}

What is different from the standard theory of labeling is that now we have <a, is> rather than <ϕ, ϕ> as the label of the structure in (13b). I will return to the implications of this difference later.

[14]This is equal to the statement that ST is not a set and ST is not empty.

An important question regarding the current formalization of labeling is whether labels themselves can be the input to the labeling algorithm. This question should be taken seriously because, based on our assumptions, labels are heads returned by Minimal Search. If they are heads, why can't they behave like regular heads in labeling? For instance, suppose the goal is to provide a label for the set $\alpha = \{_{\alpha}\mathrm{X}, \{_{\beta}\mathrm{Y}, \{_{\kappa}\mathrm{Z}, \ldots\}\}\}$. If the label of β has already be computed, that is, Y has been taken as its label, and if we further assume that labels feed Minimal Search as its input for the labeling of α, α is then equivalent to $\{\mathrm{X}, \mathrm{Y}\}$. We now have a labeling problem, which does not exist if labels are not used as legitimate objects for labeling, as in that case the label of α should be simply X. An essential reason that including labels as input to MS-labeling causes problems is because this obscures the distinction between heads and sets. The head–set distinction is an important assumption that makes the labeling algorithm work in the first place. Otherwise even for simple cases such as {H, XP}, H cannot be simply returned as the label.

A possible way to avoid taking labels as input to MS-labeling is that we can simply stipulate that Minimal Search does not take labels as input. However, then we owe an explanation as to why labels are, on the one hand, exactly heads yet, on the other hand, not usable by Minimal Search. A better solution could be that labels are computed top-down rather than bottom-up. That is, the labeling algorithm does not apply throughout the derivation, but only right before transfer, taking the syntactic object undergoing transfer (Chomsky 2015b) as its SD. Take $\alpha = \{_{\alpha}\mathrm{X}, \{_{\beta}\mathrm{Y}, \{_{\kappa}\mathrm{Z}, \ldots\}\}\}$ again as an example. The label of α is computed before the label of β. Consequently, the computation of the label for the superset α will not be affected by the label of its set member β. This is a favorable result. Such a solution is compatible with the assumption that labels are crucial for interpretation purposes at the interfaces, not for Syntax. In other words, labels are not computed in the course of derivation, for example, right after each Merge operation. Another important consequence of this solution is that labels cannot be used for MS-Agree as well, given the assumption that MS-Agree may be initiated right after the Merge of a head with unvalued features. Details of the consequences of MS-Agree and MS-labeling will be discussed in the next section.

3. Implications of MS-Agree and MS-labeling

3.1 Agree and labeling can only be partially reduced to Minimal Search

Recall that one of the primary motivations for giving a formal definition of Minimal Search is to evaluate the appealing conjecture proposed by Chomsky (2013, 2015a): both Agree and labeling can be reduced to

Minimal Search.

With the formal definition of Minimal Search in (5) and the definitions of MS-Agree and MS-labeling in (9) and (12), it is quite clear that Agree and labeling cannot be completely reduced to Minimal Search. Our definition of MS-Agree in (9) shows that Agree uses Minimal Search to find the target. However, besides Minimal Search, Agree must resort to the second step, Valuation, in order to copy the valued features from the target to the corresponding unvalued features on the trigger. Our definition of MS-labeling in (12), on the other hand, is essentially Minimal Search. But still labeling is an independent linguistic operation that takes the output of Minimal Search as the label. In other words, labeling specifies a way to use the output of Minimal Search. Minimal search itself does not give instructions or put any restrictions on how to use its search output, as the output could feed either Agree or labeling.

In addition, there is another distinction between the definition of Agree (9) and the definition of labeling (12); namely, the specific *values* of SDs and STs for Minimal Search are determined by Agree and labeling independently. The assignment of the values of SDs and STs thus serves distinctive purposes in Agree or labeling. Therefore, even if we take the Valuation step in Agree as a separate operation that occurs at SM/PF and not as part of the syntax of Agree, so that both labeling and the syntactic part of Agree are essentially Minimal Search, we still cannot reduce Agree and labeling fully to Minimal Search, given that the SDs and STs are assigned by Agree and labeling separately.

It is worth noting that with Minimal Search, we do unify a core component of Agree and labeling. The search algorithms for Agree and labeling is the Minimal Search algorithm. This is partially in accordance with Chomsky's intuition regarding a possible reduction of Agree and labeling to Minimal Search.

Therefore, I conclude that although Agree and labeling both involve Minimal Search, they are linguistic mechanisms that are independently needed, arguably for interface interpretation purposes. If we take Agree and labeling away from Syntax, Minimal Search as a general search algorithm will not know for what and into which domain it should search. This is probably a general mechanism underlying the application of 3rd factor principles in language: 3rd factor principles serve general purposes, and their specific application in language must interact with linguistic operations for them to accomplish particular linguistic purposes. In the case of MS-Agree and MS-labeling, the input to Minimal Search is provided by specific linguistic operations (Agree or labeling), whereas the output of Minimal Search is handled by these linguistic operations.

3.2 Agree consists of two sub-operations

In the definition of MS-Agree in (9), Agree consists of two sub-operations: Minimal Search and Valuation. The separation between Minimal Search and Valuation is a natural consequence of the definition of Minimal Search, as Minimal Search does not in any way tell us how to copy the valued features on the returned head to the unvalued features that have initiated the search. The only thing that Minimal Search does is to search for certain features in some domain. When a head bearing that feature is found, the head is returned as the output. The output will then feed other operations such as Valuation.

Proposals separating Agree into two independent sub-operations can be found in Arregi and Nevins (2012), Bhatt and Walkow (2013), Smith (2017), Kučerová (2018), and Bjorkman and Zeijlstra (2019). The separation can potentially solve a debate on the nature of Agree: The Minimal Search part likely happens in Syntax, but the Valuation part can be done at the SM/PF interface. If the above authors are on the right track, the definition of Minimal Search advocated here will derive the distinction between the search operation and the Valuation operation in Agree, which would have been otherwise an additional assumption in the previous studies.

3.3 <ϕ, ϕ> cannot be a label

Furthermore, as we have mentioned, if we have a fully unified labeling algorithm, we cannot on the one hand obtain a head as the label for instances such as {kick, {the, ball}}, and on the other hand return a ϕ-feature pair for instances such as {{a, {little boy}}, {is, {kicking the ball}}} (see (8)). One way to return a ϕ-feature pair in the second case is setting the ST to ϕ-features (see Chomsky 2013, p. 13). In fact, according to this approach, for any type of feature pairs such as <ϕ, ϕ> and <Q, Q>, the search algorithm needs to set that type of features as the ST. In other words, the search algorithm has to be assigned an ST whose value depends on the specific type of the SD. This is a look-ahead feature that is not desirable, and I believe it cannot be easily implemented by a simple search algorithm. A possibility to avoid this look-ahead problem is that Minimal Search always looks for some specific types of features. The type of features that is most suitable for labels is of course categorical features. However, if Minimal Search always searches for categorical features, it cannot look for ϕ-features or Q-features. Consequently, this algorithm turns out to be almost identical to the one advocated in this paper.

Another way to return a <ϕ, ϕ> pair from {{a, {little boy}}, {is, {kicking the ball}}} is to add another operation to MS-labeling: after *a* and *is* are returned by Minimal Search, MS-labeling compares their features and returns their shared features as a feature pair. The feature pair is then taken as the label. In terms of the visual search task in Figure 1 above, an analogue would be that we are searching for animals first. Once we find two animals, we compare their features. If both of them are hairy, we can use the shared feature <hairy, hairy>, or "hairy things," to label them. This complicates the labeling algorithm as it adds an additional comparison operation. This additional operation is vacuous for cases such as {kick, {the, ball}}, since only one head *kick* will be returned. I will argue below that the additional comparison operation is not necessary; in fact, its application causes the loss of important categorical information (e.g., "hairy animals" becomes "hairy things" in visual search). In terms of Agree, such a complication in the search algorithm is not needed or even irrelevant at all; thus it is hard to imagine how such a complication can be added to a general definition of Minimal Search for labeling and Agree technically. As will be discussed below, this operation also causes other problems. In a word, the additional comparison operation is unnecessary and is inadequate to be included in the definition of MS-labeling.

By contrast, if for both {kick, {the, ball}} and {{a, {little boy}}, {is, {kicking the ball}}} we have the same ST, that is, ST = any feature (not a particular one) or any head, then what is returned in {{a, {little boy}}, {is, {kicking the ball}}} must be a head pair as well. That is, <a, is> will be returned as the label.

There may be a question as to what the label <a, is> actually is. The same question is equally applicable to Chomsky's (2013) $<\phi, \phi>$ pair. A possible advantage of $<\phi, \phi>$ is to show directly to the interface the shared features between two heads. However, I will argue below such a "decision" of providing information to interfaces should not be made in Syntax. Instead, the interfaces should determine what <a, is> is when interpreting the pair. $<\phi, \phi>$ as a "premature" abstraction of shared features between two heads leads to significantly information loss. Labels such as <a, is> in fact contain richer information than labels such as $<\phi, \phi>$; for example, the former includes the categorical information (in addition to the shared ϕ-features) whereas the later does not. Details aside, one possible way to interpret <a, is> at the CI interface is that this label tells us its syntactic category, as the heads contain categorical features. Therefore, <a, is> includes the information <D, T>. This is a desirable result. This is consistent with the original and most important motivation of the labeling theory (Chomsky 2013): labels tell the interfaces what the objects are in terms of their syntactic categories.

Furthermore, taking categorical features instead of the ϕ-features as the label solves an independent problem caused by the <ϕ, ϕ> label (Shim 2018). Narita (2012) points out that if the label for a TP is <ϕ, ϕ>, then it requires the uninterpretable ϕ-features on T to be deleted upon transfer to the CI interface, rendering it visible to the CI interface. That is, on the one hand, ϕ-features on T must be deleted at CI; on the other hand, the same features must be present as part of the label at CI. This conflict is naturally circumvented under the current proposal: MS-labeling returns a head pair instead of a ϕ-feature pair for a finite TP.

3.4 Resolving problems of labeling and invisible copies

A related problem is how we can force the movement of the *v*P-internal subject to SpecTP. Recall that Chomsky (2013) achieves this by resorting to the idea that ϕ-features in agreement could be taken as a legitimate label for TP. If the subject and T head are not in an agreement relation, the TP set is labelless. The same reasoning applies to wh-movement. In the following example, if the wh-phrase *who* moves to SpecCP, then the CP can be labeled as <Q, Q>. Movement of any other syntactic objects to SpecCP renders a labelless set.

(14) $[_{\text{CP}}$ who$_{[\text{vQ}]}$ $[_{\text{C}'}$ does$_{[\text{uQ}]}$ John like <who>]]

It is worth noting that although Chomsky's labeling theory has the benefit of deriving the movement of *v*P-internal subject to SpecTP, it faces some potential empirical problems. There are two kinds of problems: (i) If the *v*P-internal subject stays at Spec*v*P and does not move away from that position, it causes a problem because *v*P will be labelless given the *v*P-internal subject does not agree with *v* (as Chomsky (2013, Footnote 35) points out); (ii) if a DP stays at SpecTP but does not agree with it, it also causes a problem because the TP will be labelless. In fact, the problem touches on a broad range of constructions: all types of {XP, YP} constructions will be without labels if XP and YP (or X and Y) do not agree. I will turn back to this topic later and demonstrate these problems with examples, and I will argue that the MS-labeling approach is immune to these problems.

For the MS-labeling approach, what may be difficult is that nothing prevents a DP which is not in an agreement relation with the T or C head to stay at SpecTP or SpecCP, as the label does not have a direct relation with the features in agreement. To overcome this difficulty, I will argue below that the labeling problem related to TP and CP is in fact not only a CI interface issue, but also an SM interface issue, assuming that labels are input to both interfaces (at least the categorical information needs to be available to both interfaces). That is, the problem may be not that TP and CP are labelless if their two members are not in an agreement relation. Instead, the problem is whether a <D, T> pair or a <D, C> pair can be considered a legitimate SM relation. For example, if a <D, T> pair is <a$_{[v\phi]}$, be$_{[u\phi]}$>, where *a* and *be* are not in an agreement relation, then the pair <a$_{[v\phi]}$, be$_{[u\phi]}$> cannot be licensed as a legitimate SM relation, as the unvalued features on *be* could have been valued via an agreement relation with those on *a* but this is not the case. This is a natural requirement on the pair <a, be> if we consider an agreement relation (especially the Valuation component of Agree) also an SM relation.[15]

The same reasoning applies to the labeling of the CP in (14); namely, the SM interface would need to see that *who* and C_{does} are in an agreement relation to license <who, C$_{\text{does}}$> as a legitimate SM relation, so that (for instance) the SM interface can assign a proper intonation contour to *who* and C_{does} and phrases labeled by them. The SM restriction on labels is summarized as below:

(15) *SM Condition on Labels* (to be revised)

<X, Y> is a legitimate label at SM only when X and Y exhibit an agreement relation.

(15) takes care of all empirical cases (in English) that Chomskyan labeling theory covers with feature pairs as labels. In particular, it derives the EPP from an SM condition as it forces the raising of *v*P-internal subject to SpecTP where the SM Condition on Labels (15) can be satisfied. In addition, (15) seems to also solve the invisibility problem regarding lower copies of movement. Recall that according to Chomsky (2013), the *v*P-internal subject in (16) needs to move further to solve the labeling problem with *v*P, as *v*P will be labelless if the subject stays in *v*P since it does not agree with *v*.

(16) {$_{\text{TP}}$ a little boy, {T$_{be}$, {$_{v\text{P}}$ <**a** little boy>, {$v_{\mathbf{be}}$, {kicking the ball}}}}}

A similar analysis could apply to MS-labeling, although the *v*P in (16) will not be labelless; instead, it has a label <a, v_{be}>, which may encode a θ-relation at CI. The point is, <a, v_{be}> cannot be licensed at SM due to the above SM Condition on Labels in (15). Therefore the *v*P-internal subject needs to move to SpecTP, creating a legitimate label at SM: <a, T$_{is}$>.

However, this is not the end of the story for both Chomsky's approach and the approach advocated in this paper so far, because the lower copy of the subject is at Spec*v*P, and it is exactly the same with its higher copy at SpecTP. So Chomsky (2013) has to stipulate the following condition on lower copies:

[15] See Footnote 3.4 for discussion of reasons why an SM condition could exist in the first place.

(17) *Invisibility of Lower Copies*

Lower copies are invisible to labeling.

Nevertheless, there is no principled reason why a lower copy cannot serve to determine the label of a set of which it is a member (Abe 2016; Mizuguchi 2019). Another critical question arises due to (17): If the *v*P-internal subject is invisible to labeling, then how it becomes visible again at CI? Note that the *v*P-internal subject *a little boy* must be visible to CI because it encodes the original θ-role position.[16]

Fortunately, MS-labeling offers a reasonable solution to the above problem caused by the Invisibility of Lower Copies Hypothesis in (17). In fact, we do not have to assume (17) at all, which will free us from another unnecessary stipulation. I assume, following Nunes (1995) and Bošković and Nunes (2007), that all the copies except one must be deleted at SM as a result of the following independent requirement on linearization: linear order is determined by asymmetric c-command relations (Kayne 1994). Multiple copies give rise to symmetric c-command relations with other syntactic objects. For example, in the sentence (16) above, after moving from Spec*v*P to SpecTP, *a little boy* occurs in two syntactic positions. There is not an asymmetric c-command relation between *a little boy* and e.g., T_{be}, because *a little boy* at SpecTP c-commands T_{be}, but T_{be} c-commands the lower copy of *a little boy* at Spec*v*P. This causes a linearization problem. A way to avoid this linearization problem is to delete all but one of the copies at SM.

Turning back to the example in (16), if the lower copy at Spec*v*P is deleted at SM due to independent linearization issues, which leads to the deletion of all its phonological features, leaving syntactic features such as categorical features unaffected, then the label for *v*P after the movement of the *v*P-internal subject is $<$D, $v_{be}$$>$. I suggest that $<$D, $v_{be}$$>$ composes an exception to the SM Condition on Labels, as D does not have any phonological content, and should not be subject to such an SM condition. We can now revise the SM Condition on Labels to make it explicit that cases such as $<$D, $v_{be}$$>$ are not subject to the SM condition.[17]

(18) *SM Condition on Labels* (to be further revised)

Given two heads X, Y, if X, Y have phonological content then X and Y must exhibit an agreement relation for <X, Y> to be a legitimate label at SM.

[16] I should thank Samuel Epstein for bringing this to my attention in our discussion of another project.

[17] See Takita (2020) for a similar way to derive the Invisibility of Lower Copies Hypothesis.

A note should be in place before we move on: by (18) I do not claim that phonological null heads cannot agree. Rather, if either X or Y is phonologically null, the {X, Y} label is exempt from (18) and thus will not cause a problem as far as (18) is concerned.

However, this revised version of the SM Condition on Labels (as well as the standard theory of labeling) still encounters a problem with clausal gerunds:

(19) a. [$_{\text{XP}}$ **David/him** being the chairman] has meant more work for all of us

b. [$_{\text{XP}}$ **those people/them** being the committee members] has meant more work for all of us

c. [$_{\text{XP}}$ **John/Him** leaving the family] devastated my mother. (modified from a sentence found in Corpus of Contemporary American English (COCA))

d. [$_{\text{XP}}$ **Bill/me** asking the question] changes the answer.

XPs in these examples are previously analyzed either as a verbal phrase that is selected by a nominal head (e.g., Abney 1987), or specifically TPs (e.g., Pires 2006, 2007)/special IPs that have some nominal properties (e.g., Reuland 1983; Baker 2005). There is no evidence of agreement between the specifier (subject) of a gerund and any head inside the gerund, e.g., T/auxiliary, Asp, *v*, or V. It also remains unclear how the accusative case is assigned to the specifier of gerund.[18]:

(20) Question: Who wants to try this game?

(21) Answer:

a. Me/*I.

b. Just me/*I.

c. Me/*I too.

[18]If clausal gerunds occur in the complement of verbs or prepositions, the specifier of the gerunds must be accusative; if clausal gerunds occur in adjunct positions, including so-called absolute constructions (Reuland 1983), either nominative or accusative specifiers are allowed; if otherwise the clausal gerunds occur in the subject position of the matrix TP, a construction that we focus on here, it seems accusative specifier is dominating, and nominative specifier is very rare, according to the author's preliminary survey of relevant sentences in COCA. See a relevant discussion of case assignment in this last type of gerund in Pires (2006, pp. 58-60). A possibility is that the accusative case on the specifiers of the gerunds in (19) is an expression of default case due to (attempted) case assignment failure, an instance similar to default accusative case observed in ellipsis and gapping, as discussed by Schütze (2001)

d. And me/*I!

e. Me/*I next!

f. Not us/*we.

g. Me/*I neither.

h. I/*Me do (too).

(22) Question: What did everyone eat?

(23) Answer: Me/*I, beans; him/*he, rice; them/*they, carrots.

(Schütze 2001: 211–212)

Schütze (2001) points out a contrast between (21a–g) and (21h): in the former instances "Infl is missing in some deeper sense," and accusative case on the DP in the subject position occurs; however, in the later case, where Infl is present, only nominative case is allowed. That is, if Infl does not exist in the answers in (21), no agreement or case assignment will occur, and the subject will be left within its original position, Spec*v*P.[19] This does not cause labeling failure in Syntax. Schütze (2001) makes a similar observation with regard to gapping (22, 23). Following Schütze (2001) and integrating insight from Preminger (2014), I assume that default case is a consequence of case assignment failure, and therefore case as well as ϕ-feature agreement cannot provide a way to properly label the XPs in (19). In sum, for reasons discussed above, the XPs marked in (19) are potentially problematic for Chomsky's (2013) labeling theory, as it incorrectly predicts that the XPs in (19) will be filtered out by labeling failures.

Similarly, clausal gerunds as complements of prepositions and verbs also cause a problem to the standard theory of labeling:

(24) a. Voters might be weary of [$_{XP}$ **Democrats/them/*they** controlling the White House].

b. This tape shows [$_{XP}$ **volunteers/them/*they** counting the collection money at a crusade].

The grammaticality of the sentences in (19) and (24) thus indicates that a <D, X> label does not give rise to problems at CI or SM although no agreement relation is attested between D and X.

[19] Based on similar ellipsis constructions in answers such as (21a–g) which license negative polarity items, Den Dikken et al. (2000) argue that the answers are in fact in the specifier of a projection lower than TP, e.g., *v*P. See Van Craenenbroeck and Den Dikken (2006) and citations therein for arguments along similar lines.

Existential constructions could also be a problem. Take (25) for an example. If the DP *a child* stays in the specifier position of the small clause SC in (25) and does not agree with the head of SC, whatever it is, the SC will be labelless according to Chomsky's (2013) labeling theory.

(25) [$_{\mathrm{TP}}$ there is [$_{\mathrm{SC}}$ [$_{\mathrm{DP}}$ a child] [$_{\mathrm{PP}}$ in the room]]]

On the other hand, if we assume *a child* raises to the matrix SpecVP and agrees with a null V, this problem may be resolved. (25) is not a problem to my analysis either, because P does not bear ϕ-features in English and it cannot involve in an agreement relation with a D head, exempting the label <D, P> from the SM Condition on Labels.

In addition, why can *there* in (25) stay at SpecTP? Is there an agreement relation between *there* and *is*? It may be argued that *there* is inserted after Syntax to satisfy the EPP or *there* bears at least some ϕ-features (e.g., Chomsky 2000). These analyses can circumvent the problem under discussion for Chomsky's (2013) labeling theory if *there* agrees with *is* so that the agreeing features can be taken as a label. However, we then are forced to assume that *is* agrees with *there* in addition to the fact that it also agrees with *a child*, an exceptional agreement pattern that is not seen elsewhere in English and thus calls for an explanation. (25) is not a problem for MS-labeling, if we assume that *there* does not bear ϕ-features (see e.g., Chomsky 1995, p. 342) (which is the preferred null hypothesis) or *there* is inserted after Syntax, which makes it impossible to be involved in an agreement relation with *is*.

To capture the above observations, the SM Condition on Labels is revised to its final version in (26).

(26) *SM Condition on Labels* (final)

Given two heads X, Y, if X, Y could form an agreement relation at SM, then X and Y must exhibit an agreement relation for <X, Y> to be a legitimate label at SM.

This interpretation principle at SM says that if X and Y could be in an agreement relation at SM, then there must be a phonological manifestation of this agreement relation in order to license the label <X, Y> at

SM.[20] In other words, if X and Y are not possible to be in an agreement relation at SM, the SM condition in (26) does not apply. We have thus far discussed two cases where two heads are *not* able to form/exhibit an agreement relation at SM: (a) either X or Y is phonologically null; (b) either X or Y's features or feature values are not accessible for agreement.

Relevantly, (27) renders additional support to the SM Condition on Labels defined in (26). As long as *Batman* is assigned (accusative) case (by the C head *for*), it can stay at SpecTP, although there is no obvious evidence of agreement between *Batman* and the T head *to*.

(27) It is time [$_{\text{CP}}$ for [$_{\text{TP}}$ Batman/him/*he to retire]].

It is easy to see that the standard theory of labeling incorrectly predicts that such an agreement relation is necessary to avoid labeling failure, if the infinitive T *to* is assumed to bear no ϕ-features (see Chou and Fernández-Salgueiro (2020) for a different assumption).

This issue naturally extends to ECM cases, under the assumption that there is no agreement between an infinitive T and its specifier yet the infinitive TP does not cause labeling failure:

(28) Bill believes [$_{\text{TP}}$ it [$_{\text{T'}}$ to be a dishonest account]].

Another phenomenon that supports (26) comes from languages with quirky subjects (e.g., subjects that are dative case marked). (29a) indicates that the verb/T agrees with the subject when it bears a nominative

[20]A question immediately arises: why do we need this SM condition at the first place? The intuition underlying the two versions of SM conditions in (18) and (26) is that whenever an {X,Y} label is formed, it signals an instance of structural dependency. As long as such structural dependency *can be phonologically manifested* given a particular morphosyntactic system, such a phonological manifestation is required. This does not imply that every language must have phonological manifestation of structural dependency, which is obviously incorrect. Instead, the SM condition requires a phonological manifestation only when a given language employs morphosyntactic means in a particular structural dependency. Such phonological manifestation of structural dependency will be potentially accessible to child language acquisition. For sure such requirement does not apply to languages that do not have, for instance, morphosyntactic ϕ-features, as such phonological manifestation is not possible. It is also not applicable to other cases where such phonological manifestation is impossible, for instance, either X or Y in <X, Y> do not have morphosyntactic features. In this paper, I discuss a set of circumstances where the SM condition (26) does not lead to a problem as long as X or Y is phonologically null or lacks accessible feature values.

case, but this agreement does not occur when the subject is a quirky subject; that is, when the subject bears a dative case (29b). Instead, a default agreement is triggered.

(29) *Icelandic* (Sigurðsson 1996, p. 1)

a. Strákarnir leiddust/*leiddist.
boy.the.PL.**NOM** walked.hand.in.hand.**3PL**/*3SG
'The boys walked hand in hand.'

b. Strákunum leiddist/*leiddust.
boy.the.PL.**DAT** were.bored.**3SG**/*3PL
'The boys were bored.'

Importantly, it has been argued convincingly that quirky subjects, like standard nominative subjects, are also at the position of SpecTP. Quirky subjects can pass subjecthood tests such as raising, reflexivization, subject-verb "inversion" (e.g., in questions), ECM, subject control and Conjunction Reduction, etc. (Zaenen et al. 1985; Sigurðsson 1992). Therefore, (29b) is problematic to Chomsky's standard theory of labeling, as the quirky subject at SpecTP does not agree with the T head. By contrast, MS-labeling is exempt from this problem because the SM Condition on Labels (26) is not applicable to (29b) because the feature values on the quirky subject are assumed to be inaccessible for the Valuation operation in MS-Agree.[21]

In fact, the SM Condition on Labels as defined in (26) solves another general problem to Chomskyan labeling theory: labeling failure is again incorrectly predicted for TPs (including obligatory control constructions), gerunds, XP inside PPs and *while* phrases, all of which include a PRO in their specifier positions but no agreement or case assignment is involved:

(30) a. [$_{TP}$ PRO to sleep under the stars] is the only thing Eli wants now.

b. [$_{XP}$ PRO wasting resources] is the first thing I would avoid.

[21]Different proposals have been made to explain why no agreement/default agreement is triggered with quirky subjects in Iceland (see Preminger 2014: Ch. 8 for a review). A comprehensive review and evaluation of these approaches is beyond the scope of this paper, but it is worth noting that a common property of all these proposals is that the values of the ϕ-features on quirky subjects are inaccessible for agreement. That is, although the ϕ-features (feature attributes) on quirky subjects may cause blocking effects in Agree, I assume that the values of these phi-features are not accessible to be copied to the trigger, which leads to default agreement on the trigger. This is consistent with the definition of MS-Agree (9) in this paper, which, as we have seen, shows a natural distinction between Minimal Search and Valuation.

c. [$_{PP}$ before [$_{XP}$ PRO getting off the bus]], I lost my cell phone.

d. John kept walking slowly, [while [$_{XP}$ PRO drenching the road with insecticides]]. (Pires 2007: 199)

For all these cases, the presence of PRO causes a labeling problem for the standard theory. Note that these instances of PROs are not able to be formalized as lower copies of movement, no matter whether they are controlled or not (Hornstein 2001; Pires 2006). Therefore, even if we adopt the movement approach to PRO in general and embrace the assumption that the lower copies are invisible, the problems do not go away. By contrast, these examples do not pose any problems to the SM condition in (26), since PROs are phonologically null and thus (26) does not apply.[22]

Before I conclude this section, it must be noted that the SM Condition on Labels defined in (18) and (26) generates distinct empirical predictions: (18) predicts that *v*P-internal subject needs to move out of *v*P (and the specifier of any higher heads that do not agree with it) because <D, *v*> does not count as a legitimate phonological relation at SM–this is similar to the empirical predictions that Chomskyan labeling theory generates; however, (26) tolerates <D, *v*> and will not rule it out at SM mainly because *v* is assumed to be free of ϕ-features, and therefore cannot agree with the *v*P-internal subject, which renders (26) inapplicable. As a consequence, (26) cannot derive the raising of *v*P-internal subject from labeling theory. This is arguably reasonable as we have seen cases where the subject can stay inside *v*P. It would be better to generally not force the subject to move out from *v*P via labeling theory, and explain exceptional cases with more specific constraints.

Another difference between (18) and (26) is that (26), unlike (18), allows the subject to be optionally phonologically null, and therefore it does not require an overt subject. Such a feature may be consistent with the idea that null syntactic elements can also satisfy the EPP (e.g., Holmberg 2005), yet meanwhile it does not capture the obligatoriness of overt realization of the subject of finite TP in English, which is likely because pro-drop is a language/construction-specific property (Doner 2019). In fact, the current discussion

[22] In addition to the empirical evidence discussed above for (26), please be referred to Mizuguchi (2019) for discussion of some other problematic cases for the standard labeling theory where {XP, YP} structures do not involve agreement yet do not lead to labeling failure, including partial wh-movement in German, object shifting in Icelandic, and in-situ subjects in German. As far as I can see, none of these cases composes a problem to MS-labeling because these are instances where (26) is not applicable.

seems to point to the direction that the attempt to derive the EPP (which has a broad and complicated distribution cross-linguistically; see Doner (2019)) completely from labeling theory alone is probably empirically untenable and theoretically inappropriate: empirically, we have seen that there are many cases where a specifier can stay inside a phrase without causing any labeling issues, although the specifier does not agree with the head of that phrase; theoretically, the standard theory must assume that lower copies are invisible, which is not motivated in a principled way. Consequently, if (26) is to be adopted, the EPP must be explained with some mechanism other than labeling, to which I have no definite proposal to offer at this point (see e.g., Richards (2016) for an interesting alternative to derive the EPP). However, I also acknowledge that if we could explain away examples in (19) to (30) and show that these examples are orthogonal to labels at the SM interface, then we may consider sticking to (18) instead.

Possible evidence for the SM Condition on Labels defined in (18) but against that in (26) is from Epstein et al. (2014). EKS (Epstein et al. 2014) suggest that (31) can be filtered out by a labeling failure under the standard labeling theory, because *a* and *to* in TP2 are not in an agreement relation and thus TP2 cannot be properly labeled.

(31) *[$_{\text{TP1}}$ there [$_{\text{T}'1}$ seems [$_{\text{TP2}}$ [$_{\text{DP}}$ a man] [$_{\text{T}'2}$ to be outside]]]]

It must be noted that the SM Condition on Labels defined in (18) can also capture (31) along the same lines. Samuel Epstein (p.c.) indicates that (26), however, may not be able to exclude (31). This is because (26) predicts that <a, to> is exempted, because *to* does not bear ϕ-features and therefore it cannot be in an agreement relation with *a*.

In summary, it seems the above evaluation generally favors the SM Condition on Labels defined in (26). However, there are also substantial empirical gains if we adopt the SM Condition on Labels in (18) instead. I will thus leave the final verdict to future research.

3.5 Other consequences

Finally, besides the points we mentioned above, we have also pointed out that in order to keep the head–set distinction, labels cannot feed Minimal Search, and therefore labels cannot be used for MS-Agree.[23] In addition, as we have mentioned, a natural way to avoid labels feeding into the labeling algorithm is to compute labels from top down. This is consistent with the overall idea that labels are for interpretative purposes at interfaces and "must take place at the phase level, as part of the Transfer operation (Chomsky 2015b, p. 6; see also Chomsky 2008 and Ott 2011)." Minimal search will then search down from the whole syntactic object that is to be transferred.

4. A potential question

In this section, I would like to address a potential question which touches on the parallel search aspect of the definition of Minimal Search in this paper: What if the parallel searches do not communicate?

As mentioned in Section 2.2, a potential formalization different from the global search algorithm advocated in this paper is the modular search algorithm, according to which parallel searches do not communicate with one another regarding their search results. They are like modules whose information cannot be directly accessed from other modules. However, I have identified a main drawback of that approach, as illustrated with visual search: the algorithm does not terminate even after a search target has already been returned, due to the fact that under this algorithm searches do not communicate and they do not know after a target has been returned. Therefore, as we have seen in the example of visual search previously, the modular search algorithm will necessarily continue to exhaust all possible sets and subsets that are available for search even after a target has already be found, thus violating the principle of computational efficiency.

Below I will highlight a few empirical problems associated with the modular algorithm in Minimal Search for Agree. The problems that are highlighted below are essentially caused by the very nature of modular search: it violates computational minimalism.

[23] As a result, a system that feeds labels to Agree is potentially problematic. For instance, to account for Cyclic Agree, Béjar and Rezac (2009) suggest that the label of v', i.e., the v head, can act as a head with unvalued features that probes into its sister, Spec*v*P. Although such an analysis is an ingenious use of a variant of labeling algorithm proposed in Chomsky (1995, 2000), it loses the set–head distinction and cannot be formalized under MS-labeling and probably any other possible formalization of Minimal Search that requires the set–head distinction. The empirical basis of the Cyclic Agree analysis can be covered with MS-Agree, details of which can be found in Ke (2019).

Imagine that T with [uϕ] searches for a head with matching features. Using modular search, the search target is set to ϕ-features attributes as those on T and the search domain is *v*P. The modular algorithm would find *both* the subject and the object.

(32) $\{T_{[u\phi]}, \{_{vP}\ \{_{DP}\ \text{the}_{[v\phi]}\ \text{girl}\}, \{_{v'}\ \text{read}, \{\text{the}_{[v\phi]}\ \text{book}\}\}\}\}$

This is because the searches into DP and *v*′ are parallel searches and they do not communicate with each other in terms of their search results, according to the modular search algorithm. Consequently, after confirming that the D head of the subject *the girl* bears the search target and *read* does not bear the search target, the search into *v*′ will continue to look for the search target. The search terminates only after *v*′ and all accessible subsets of *v*′ are exhaustively visited and the head of the object *the book* is returned.

Such a result is not what we want. Note that the hierarchical nature of language is one of language's most crucial characteristics. The subject c-commands the object; however, the modular algorithm returns both the subject and the object (more accurately their heads), thus ignoring one of the most important features of language. Empirically, if we adopt the modular algorithm, we must have a way to prevent Minimal Search from returning the object. One may say that the object is not available for Minimal Search for some special reason (e.g., it has been assigned case or it has been transferred when T enters the derivation[24]). But we know from other languages that the object must be generally available for Minimal Search as long as the subject does not bear the search target. This has been clearly shown in the following contrast in Icelandic (taken from Sigurðsson 1996):

(33) a. Við lásum/*las bókina.
we.NOM read.**1PL**/*3SG the-book.ACC
'We read the book.'

b. Henni leiddust strákarnir.
her.DAT bored.**3PL** the-boys.NOM.PL
'She found the boys boring.'

When the subject bears the nominative case, T agrees with it (33a). However, when the subject bears the "inherent" dative case, T does not agree with the subject but can agree with the object which bears the nominative case (33b).

[24]Note that according to Chomsky (2015b) the internal argument is not transferred when T enters the derivation, only its lower copy is.

Cyclic Agree is another case where the object is available only when the subject is not available for Agree. Béjar and Rezac (2003), Rezac (2004), and Béjar and Rezac (2009) argue that agreement in Georgian occurs in two cycles. In the first cycle, a head with unvalued features (the trigger) probes downward to find a head matching the unvalued features. If the search in the first cycle is not successful, the trigger may instead search upward. Béjar (2003) and Béjar and Rezac (2003) assume that person and number features are represented separately in Georgian: *v* bears a [uPerson] feature and T bears a [uNum] feature. (34a) shows that, when the subject is plural, $T_{[uNum]}$ agrees with the subject and $v_{[uPerson]}$ agrees with the object.

(34) Georgian (Rezac 2004, p. 72)

a. m-xedav-t
   1-see-PL
   'You(.PL) see(.PL) me.'

b. g-xedav-t
   2-see-PL
   'I see(.PL) you(.PL).'

By contrast, (34b) exemplifies that, when the subject is singular, $v_{[uPerson]}$ still agrees with the object, but $T_{[uNum]}$ now agrees with the object instead of the subject. Béjar and Rezac suggest that the singular number feature on the subject is underspecified or not present in syntactic representations in Georgian, so downward searching from $T_{[uNum]}$ skips the subject and expands the search domain down to the object. The contrast between (34a) and (34b) strongly support the idea that if two syntactic objects α and β both bear the search target and if α asymmetrically c-commands β, α rather than β will be found by Minimal Search for agreement purposes. Only when α is not accessible for Minimal Search could β then be returned. The global search algorithm for Minimal Search thus yields correct predictions. This is in fact resonant with Chomsky's (1993) Shortest Movement Condition, his (1995) Minimal Link Condition, and Rizzi's (1990) Relativized Minimality. By contrast, the modular algorithm makes wrong predictions as it allows both α and β be returned even if α asymmetrically c-commands β.

5. Conclusions

This paper proposes a formal definition of Minimal Search to evaluate the idea that both Agree and labeling can be reduced to Minimal Search. Different aspects of the search algorithm, e.g., breadth-first vs. depth-first search, parallel vs. serial search, global vs. modular search are discussed, and reasons for making choices between each of these pairs are given based on detailed examinations of their theoretical and empirical consequences. Furthermore, MS-Agree and MS-labeling are defined based on the definition of Minimal Search. I have shown that MS-Agree and MS-labeling are clearly two distinct operations: although they share the same search algorithm, they serve different purposes and involve different values of search domain and search target. In addition, Agree consists of two sub-operations: Minimal Search and Valuation, whereas labeling does not share the Valuation operation. Therefore, Agree and labeling can be only partially but not completely reduced to Minimal Search.

With a formal definition of MS-labeling, it becomes clear that <ϕ, ϕ> cannot be a label without further complications and assumptions beyond the simple definition of MS-labeling, and head pairs that carry categorical information such as <D, T> are taken as legitimate labels for TPs. Besides obtaining a unified labeling algorithm, MS-labeling also solves various problems. <ϕ, ϕ> as a label for a finite TP is problematic because the ϕ-features on T will be deleted upon transfer to CI, whereas <D, T> circumvents this problem. In addition, deriving the EPP with <ϕ, ϕ> alone is empirically inadequate as this theory requires (i) the specifier of a phrase to agree with the head of the phrase in general, for which many exceptions are found across different phenomena and languages, and (ii) lower copies to be invisible to Minimal Search, which has been argued not to be well motivated theoretically. This paper proposes that if an SM condition on labels is adopted, several empirical challenges to the standard theory of labeling can be overcome and meanwhile the unnecessary stipulation that lower copies are invisible to Minimal Search can be dispensed with, as desired.

6. Appendix: Pseudo-code for the SA in Minimal Search

Note that in the pseudo-code I obtain the results of the parallel Minimal Search algorithm in (5) by applying the Multi_Find function, which aggregates a list of returned heads by executing the Find function in multiple processors simultaneously, assuming that multiple processors are available for Minimal Search.

GLOBAL VARIABLE:

new_SD = List()

FUNCTIONS:

1. List(x, y):

return: [*x*, *y*]

2. Member(*x*={*y*, *z*}):

    return: List(*y*, *z*)

3. Multi_Find(*x*, *y*=[*a*, *b*, *c*...]):

    return:

        List(Multi_processor(Find(*x*, *a*), Find(*x*, *b*), Find(*x*, *c*), ...))

4. Find(*x*, *y*):

    found_heads = List()

    if *y* is a head and *y* includes *x*:

        append *y* to found_heads;

    if *y* is a set:

        join new_SD and Member(*y*)

    if *y* is a list:

        Multi_Find(*x*, y):

    return: (found_heads, new_SD)

ALGORITHM:

Start: Terminate_search = False, new_SD = List(), given ST, SD

While Terminate_search is equal to False:

    (found_heads, SD) = Find(ST, SD)

    if found_heads is not empty:

        Terminate_search = True

    else if SD is not empty:

        new_SD = List()

References

Abe, Jun. 2016. Dynamic antisymmetry for labeling. *Lingua* 174:1–15.

Abney, Steven P. 1987. The English noun phrase in its sentential aspect. Dissertation, Massachusetts Institute of Technology.

Arregi, Karlos, and Andrew Nevins. 2012. *Morphotactics: Basque auxiliaries and the structure of spellout*, volume 86. Springer.

Baker, Mark C. 2005. On gerunds and the theory of categories. *Ms., Rutgers University* URL https://sites.rutgers.edu/mark-baker/wp-content/uploads/sites/199/2019/07/gerundscattheory.pdf, manuscript.

Béjar, Susana. 2003. Phi-syntax: A theory of agreement. Dissertation, University of Toronto.

Béjar, Susana, and Milan Rezac. 2003. Person licensing and the derivation of PCC effects. In *Romance linguistics: Theory and acquisition: Selected papers from the 32nd Linguistic Symposium on Romance Languages (LSRL)*, ed. Ana Teresa Pérez-Leroux and Yves Roberge, volume 4 of *Amsterdam Studies in the Theory and History of Linguistic Science*, 49–62. Amsterdam/Philadelphia: John Benjamins Publishing.

Béjar, Susana, and Milan Rezac. 2009. Cyclic agree. *Linguistic Inquiry* 40:35–73. URL https://muse.jhu.edu/journals/linguistic_inquiry/v040/40.1.bejar.html.

Bhatt, Rajesh, and Martin Walkow. 2013. Locating agreement in grammar: An argument from agreement in conjunctions. *Natural Language & Linguistic Theory* 31:951–1013.

Bjorkman, Bronwyn M, and Hedde Zeijlstra. 2019. Checking up on (ϕ-) agree. *Linguistic Inquiry* 50:527–569.

Bošković, Željko, and Jairo Nunes. 2007. The copy theory of movement: A review from PF. In *The copy theory of movement*, ed. Norbert Corver and Jairo Nunes, volume 107, 13–74. Amsterdam/Philadelphia: John Benjamins Publishing.

Branan, Kenyon, and Michael Yoshitaka Erlewine. to appear. Locality and (minimal) search. In *Cambridge handbook of minimalism*, ed. Kleanthes K. Grohmann and Evelina Leivada. Cambridge, UK: Cambridge University Press.

Chomsky, Noam. 1993. A minimalist program for linguistic theory. In *The view from building 20: Essays in linguistics in honor of sylvain bromberger*, ed. Kenneth Hale and Samuel Jay Keyser, Current Studies in Linguistics, book section 1, 1–52. Cambridge, MA: The MIT Press.

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

Chomsky, Noam. 2000. Minimalist inquiries: The framework. In *Step by step: Essays on minimalist syntax in honor of Howard Lasnik*, ed. Roger Martin, David Michaels, and Juan Uriagereka, 89–155. Cambridge, MA: MIT Press.

Chomsky, Noam. 2004. Beyond explanatory adequacy. In *Structures and beyond*, ed. Adriana Belletti, volume 3, 104–131. Oxford: Oxford University Press.

Chomsky, Noam. 2005. Three factors in language design. *Linguistic Inquiry* 36:1–22.

Chomsky, Noam. 2008. On phases. In *Foundational issues in linguistic theory: Essays in honor of Jean-Roger Vergnaud*, ed. Robert Freidin, Carlos P. Otero, and Maria Luisa Zubizarreta, 133–166. Cambridge, MA: MIT Press.

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

Chomsky, Noam. 2015a. A discussion with Naoki Fukui and Mihoko Zushi (march 4, 2014). *Sophia Linguistica: Working Papers in Linguistics* 69–97. URL `http://digital-archives.sophia.ac.jp/repository/view/repository/00000036046`.

Chomsky, Noam. 2015b. Problems of projection: Extensions. In *Structures, strategies and beyond : Studies in honour of Adriana Belletti*, ed. Elisa Di Domenico, Cornelia Hamann, and Simona Matteini, volume 223, book section 1, 3–16. Amsterdam/Philadelphia: John Benjamins Publishing Company.

Chomsky, Noam. 2019. MIT lectures. URL `https://youtu.be/r514RhgISv0https://youtu.be/GPHew_smDjY`.

Chou, Chao-Ting Tim, and Gerardo Fernández-Salgueiro. 2020. A last resort approach to (number) feature inheritance. *Studia Linguistica* 1–23.

Den Dikken, Marcel, André Meinunger, and Chris Wilder. 2000. Pseudoclefts and ellipsis. *Studia linguistica* 54:41–89.

Doner, Julianne Margaret. 2019. The epp across languages. Doctoral Dissertation, University of Toronto.

Epstein, Samuel David, Hisatsugu Kitahara, and T Daniel Seely. 2014. Labeling by minimal search: Implications for successive-cyclic a-movement and the conception of the postulate "phase". *Linguistic Inquiry* 45:463–481. URL https://muse.jhu.edu/journals/linguistic_inquiry/v045/45.3.epstein.html.

Fodor, Jerry. 1983. *The modularity of mind*. Cambridge, MA: The MIT Press.

Hiraiwa, Ken. 2001. Multiple agree and the defective intervention constraint in Japanese. In *MIT Working Papers in Linguistics*, volume 40, 67–80. Cambridge, MA: MITWPL.

Holmberg, Anders. 2005. Is there a little pro? evidence from finnish. *Linguistic Inquiry* 36:533–564. URL https://doi.org/10.1162/002438905774464322.

Hornstein, Norbert. 2001. *Move!: A minimalist theory of construal*. Oxford: Blackwell.

Horowitz, Todd S, and Jeremy M Wolfe. 1998. Visual search has no memory. *Nature* 394:575–577.

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

Ke, Hezao. 2019. The syntax, semantics and processing of agreement and binding grammatical illusions. Dissertation, University of Michigan, Ann Arbor.

Kučerová, Ivona. 2018. ϕ-features at the syntax-semantics interface: Evidence from nominal inflection. *Linguistic Inquiry* 1–77. URL https://doi.org/10.1162/ling_a_00290.

Mizuguchi, Manabu. 2019. Ambiguous labeling and full interpretation. *Studia Linguistica* 73:563–603. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/stul.12109.

Narita, Hiroki. 2012. Remarks on the nature of headedness and compositionality in bare phrase structure. In *Proceedings of Sophia University Linguistic Society 26*, 81–126.

Nunes, Jairo Morais. 1995. The copy theory of movement and linearization of chains in the minimalist program. Dissertation, University of Maryland at College Park.

Ott, Dennis. 2011. A note on free relative clauses in the theory of phases. *Linguistic Inquiry* 42:183–192. URL http://www.jstor.org/stable/23011916.

Pires, Acrisio. 2006. *The minimalist syntax of defective domains: Gerunds and infinitives*, volume 98. John Benjamins Publishing.

Pires, Acrisio. 2007. The derivation of clausal gerunds. *Syntax* 10:165–203.

Polinsky, Maria, and Eric Potsdam. 2001. Long-distance agreement and topic in Tsez. *Natural Language & Linguistic Theory* 19:583–646.

Preminger, Omer. 2009. Breaking agreements: Distinguishing agreement and clitic doubling by their failures. *Linguistic Inquiry* 40:619–666.

Preminger, Omer. 2014. *Agreement and its failures*. Cambridge, MA: The MIT Press. URL https://muse.jhu.edu/book/35120.

Reuland, Eric J. 1983. Governing-ing. *Linguistic Inquiry* 14:101–136.

Rezac, Milan. 2004. Elements of cyclic syntax: Agree and merge. Dissertation, University of Toronto.

Richards, Marc. 2007. On feature inheritance: An argument from the phase impenetrability condition. *Linguistic Inquiry* 38:563–572.

Richards, Norvin. 2016. *Contiguity theory*, volume 73. Cambridge, MA.: MIT Press.

Rizzi, Luigi. 1990. *Relativized minimality*. Linguistic inquiry monographs 16. Cambridge, MA, US: The MIT Press.

Schütze, Carson T. 2001. On the nature of default case. *Syntax* 4:205–238.

Shim, Jae-Young. 2018. $<\phi, \phi>$-less labeling. *Language Research* 54:23–39. URL http://hdl.handle.net/10371/139743.

Sigurðsson, Einar Freyr. 2017. Deriving case, agreement and voice phenomena in syntax. Dissertation, University of Pennsylvania.

Sigurðsson, Halldór Ármann. 1992. The case of quirky subjects. In *Working papers in scandinavian syntax*, volume 49. Department of Scandinavian Languages, Lund University.

Sigurðsson, Halldór Ármann. 1996. Icelandic finite verb agreement. In *Working Papers in Scandinavian Syntax*, volume 57, 1–46. Department of Scandinavian Languages, Lund University.

Smith, Peter W. 2017. The syntax of semantic agreement in English. *Journal of Linguistics* 53:823–863.

Takita, Kensuke. 2020. Labeling for linearization. *The Linguistic Review* 37:75–116. URL https://www.degruyter.com/view/journals/tlir/37/1/article-p75.xml.

Van Craenenbroeck, Jeroen, and Marcel Den Dikken. 2006. Ellipsis and epp repair. *Linguistic Inquiry* 37:653–664.

Wolfe, Jeremy M. 2010. Visual search. *Current Biology* 20:R346—R349.

Woodman, Geoffrey F, Edward K Vogel, and Steven J Luck. 2001. Visual search remains efficient when visual working memory is full. *Psychological Science* 12:219–224.

Wurmbrand, Susi. 2012. Parasitic participles in germanic: Evidence for the theory of verb clusters. *Taal en Tongval* 64:129–156.

Zaenen, Annie, Joan Maling, and Höskuldur Thráinsson. 1985. Case and grammatical functions: The icelandic passive. *Natural Language & Linguistic Theory* 3:441–483.

Zeijlstra, Hedde. 2012. There is only one way to agree. *The Linguistic Review* 29:491–539. URL https://www.degruyter.com/view/j/tlr.2012.29.issue-3/tlr-2012-0017/tlr-2012-0017.xml.