Phase-head Initiated Structure Building: Its Implications for Feature-Inheritance, Transfer and Internal Merge*

Jae-Young Shim
(University of Michigan)

This paper explores an alternative way of structure building in minimalism and proposes that along with other operations in Narrow Syntax such as Feature-Inheritance and Transfer, structure building is also initiated only by phase heads. Consequently, this paper takes one step further Chomsky's (2007, 2008, 2013) generalization that all operations in Narrow Syntax are restricted to the phase level. It further investigates the implications of phase-head initiated structure building for the motivations for Feature-Inheritance and Transfer (Chomsky 2007, 2008, Richards 2007) and shows 1) that no derivation can converge at the C-I interface without Feature-Inheritance (i.e., Feature-Inheritance is necessitated to satisfy interface conditions) and 2) that the operation Transfer is a natural by-product of (Internal) Merge.

Keywords: Structure-building, Phase-head, Feature-Inheritance, Transfer, Internal Merge

1. Introduction

In minimalism (Chomsky 1995 et seq.), structure building has been assumed to proceed in a bottom up fashion by recursive application of a set-forming operation called Merge. What has been further assumed (implicitly at least) in this framework is that the derivation starts with V. Take, for example, the generation of the v*P domain of a typical transitive construction: we start with

* I am deeply indebted to Samuel Epstein, Hisa Kitahara, Daniel Seely, Tim Chou, Marlyse Baptista and Acrisio Pires, with whom I have discussed the ideas presented in this paper at various stages of their formulation. Thanks also go to the two anonymous reviewers for their valuable comments and suggestions. Needless to say, all remaining errors are solely mine.

V, merge it with its complement DP, and create the first set, {V, DP}. Then we introduce v* into the workspace and merge it with the existing structure, which gives us {v*, {V, DP}}. We complete the v*P domain by introducing another DP and merging it with {v*, {V, DP}. As a result, we have {DP, {v*, {V, DP}.

The question this paper discusses is concerned with the very beginning of the above derivation. That is, among many other lexical items, why is it V that is first chosen and introduced into the workspace? In other words, how does syntax know such is (or must be) the case?

The aim of this paper is two-fold. First, focusing on the v*P domain, I will explore an alternative way of structure-building where the first element introduced into the workspace is not V but the phase head v*. Consequently, I will argue that along with other operations in Narrow Syntax (NS) such as Feature-Inheritance and Transfer (Chomsky 2007, 2008), structure-building is also initiated only by phase heads. Second, I will examine some theory-internal implications of phase-head initiated structure building for the formulation of Feature-Inheritance and Transfer proposed in current minimalism (Chomsky 2007, 2008, Epstein et al. 2011, Richards 2007) and propose that the former is necessitated by interface conditions and the latter is a by-product of (Internal) Merge.

The organization of the paper is as follows: In section 2, I briefly overview structure building in the v*P-phase level (implicitly) assumed in minimalism (Chomsky 1995 *et seq*.) and point out some conceptual problems with it. In section 3, I introduce six conditions, all of which are either a modification or a specification of existing conditions/postulates proposed at various stages in minimalism. Based on the conditions developed in section 3, I show in section 4 how structure building can also be initiated only by phase heads. In section 5, I explore implications of phase-head initiated structure building for Feature-Inheritance and Transfer and show that these operations can be better motivated with phase-head initiated structure building. Section 6 concludes the paper.

2. Structure Building: the Puzzle

In minimalism (Chomsky 1995 *et seq*.), structure building has been assumed to proceed in a bottom up fashion by recursive application of a (two-membered) set-forming operation called Merge defined below in (1):

(1) Merge
Merge takes two objects, X and Y, to form a set {X, Y}.[1]

If X above is external to Y, Merge of X and Y is called External Merge (EM), while, if X is internal to or part of Y, Merge of X and Y is called Internal Merge (IM). I follow Chomsky (2000) in assuming that the selector becomes the label of {X, Y}, i.e., if the selector is X, X becomes the label of {X, Y}, giving us {X, {X, Y}}; if the selector is Y, however, Y becomes the label, giving us {Y, {X, Y}}.[2]

Let us now examine how a typical transitive construction as illustrated in (2) is generated by recursive application of EM, where EA and IA refer to External and Internal Argument, respectively, and the labels are indicated by underlining.

(2) Step I: $\{\underline{V}, \{V, IA\}\}$
Step II: $\{\underline{v^*}, \{v^*, \{\underline{V}, \{V, IA\}\}\}\}$
Step III: $\{\underline{v^*}, \{EA, \{\underline{v^*}, \{v^*, \{\underline{V}, \{V, IA\}\}\}\}\}\}$

In Step I, V and its complement IA[3] undergo Merge to form a set, {V, IA}, and the selector V projects to become the label of the set. Then, in Step II, v* is introduced and merges with the existing set, $\{\underline{V}, \{V, IA\}$, to form another set, $\{\underline{v^*}, \{v^*, \{\underline{V}, \{V, IA\}\}\}$, where v* becomes the label of the outcome. Finally, in Step III, EA is introduced and undergoes Merge with the existing set $\{\underline{v^*}, \{v^*, \{\underline{V}, \{V, IA\}\}\}\}$ to form yet another set $\{\underline{v^*}, \{EA, \{\underline{v^*}, \{v^*, \{\underline{V}, \{V, IA\}\}\}\}\}\}$, where v* becomes the label.

The puzzle in the above derivation (conventionally-assumed) in current minimalism is, among many other lexical items (i.e., heads) in the lexicon (or in the Numeration), why is it that V is first chosen and introduced into workspace? In other words, how does syntax know in advance that such is (or must be) the case? One possible answer to this question might be to assume that syntax 'somehow' knows that the derivation will crash (or will be interpreted as gibberish) if it makes other choices than V since they will all eventually create a

---

[1] The internal structure of X or Y can be either simple (i.e., a head) or complex (i.e., an outcome of Merge). I will refer to the former as a 'lexical item' and the latter as a 'syntactic object (SO)'.

[2] See section 3.3 for more on this assumption. For different approaches to labeling, see Chomsky (2013) and Collins (2002), among others.

[3] If IA itself is a set (e.g., {D, N}), what V merges with is the set {D, N}. Throughout the paper, however, IA (and EA) is assumed to be a simple lexical item like *John* and often used interchangeably with D unless otherwise mentioned.

structure that would lead to a violation of the Extension Condition defined as in (3):

(3) Extension Condition (Chomsky 1995)
Merge must extend the root of the structure it applies to.

Suppose, for example, that EA (instead of V) is chosen first and subsequently undergoes EM with v*, creating {v*, {EA, v*}}. Suppose further that V is later introduced and undergoes EM with the existing structure. There seem to be at least two different ways this EM of V can proceed. One is that V undergoes EM with the existing set, {<u>v</u>*, {EA, v*}}, creating another set, {<u>v</u>*, {V, {<u>v</u>*, {EA, v*}}}}. Although the resulting structure conforms to the Extension Condition (3), it is not what we would want to generate because the C-I interface would incorrectly interpret the head V as the Spec of v*. The other option would be that V undergoes EM with the head v*. This would create a structure {<u>v</u>*, {EA, {<u>v</u>*, {v*, V}}}}, which violates the Extension Condition (3) because merging V with the head v* does not extend the root of the previously-generated structure. Assuming, however, that syntax thus somehow knows 'in advance' which derivation will lead to a violation of the Extension Condition and for that reason, syntax must start with V (and IA), immediately and inevitably runs into a problem because it invokes look-ahead properties.[4]

3. Phase-head Initiated Structure Building

Concerning the operations in NS with respect to phase heads, Chomsky (2008) proposes the following generalization (author's italics)[5]:

> It is also natural to expect that along with Transfer, all other operations will also apply at the phase level. That implies that IM should be driven only by phase heads [i.e., C and v*]. If *only phase heads trigger operations ...*

---

[4] Another possible answer to the question might be to argue that all other possible derivations are indeed tried out and only one (converging) derivation survives among them. Although this solution does not invoke look-ahead issues as the first answer does, it will impose more complexity on the Computational System of Human Language (C_{HL}, Chomsky 1995) since in the worst-case scenario, C_{HL} will need to try out three alternative derivations (i.e., v*, IA(=D/N), and EA(=D/N)), which will all ultimately lead to a violation of the Extension Condition.

[5] Similar proposals are found in Chomsky (2007, 2013) as well.

If the above proposal is indeed on the right track, it should be (at least) conceptually natural to assume that structure-building, an operation in NS, is also triggered only by phase heads. Taking Chomsky's generalization one step further, I will show in what follows that structure-building can also proceed with a phase head.

3.1. Selectional Features and Their Satisfaction

Let us first clarify technical terms for an alternative account of how structure building proceeds. First, I assume a group of features of a head H distinct from the rest of the features of it, calling the former 'selectional features' of H. To be more specific, selectional features of H include: 1) features for (thematic) argument(s) that H takes and 2) features for another head that H subcategorizes for.[6] In a simple transitive structure, for example, the head V (immediately dominated by v*) bears only one selectional feature, namely, the feature that requires a DP/NP (as its complement)[7], whereas the head v* has two, i.e., the feature that requires a DP/NP for its argument and the feature that requires V for its subcategorization. I will call a head H with these selectional features a 'selector'. Hence, V and v* are a selector (while a DP is not). I further assume that these selectional features are uninterpretable so that a structure will crash at the interfaces if it reaches the interfaces with unsatisfied selectional feature(s). Therefore, all selectional features of a selector must be satisfied before a derivation reaches each of the interfaces.

If it is true that features for argument and for subcategorization indeed belong to the same group of features (i.e., selectional features) as I assume, it should be (at least) conceptually natural to assume that these features are satisfied in the same manner, i.e., in the same geometric configuration. I thus propose the following condition on the structure generated to satisfy selectional features (of a head):

(4) C-commanding Condition on Selection Structure[8]

---

[6] These selectional features encompass what Collins' (2002) calls Theta(X, Y) and Subcat(X, Y) relations where X is the head that requires Y.

[7] I will put aside the situation where this type of V takes a clause as its complement since the issue here is not concerned with the categorical status of complement.

[8] We define c-command as follows:

X c-commands Y iff

(i) neither dominates the other, and

A selector must c-command *all* the selectees it selects for.

Condition (4) implies that operations in selection structure can only be defined in terms of c-command, not of such other relations as m-command or Spec-Head relation.

3.2. The Operation SELECT and the Summoning Condition

Following Chomsky (1995), I assume an operation SELECT but unlike Chomsky's, our SELECT 'directly' access the lexicon, chooses a head, and puts it into the workspace. In other words, our SELECT does not access an intermediate buffer such as Numeration as proposed in Chomsky (1995). Despite the difference, both come free as suggested in Chomsky (1995):

> Note that no question arises about the motivation for application of Select [...] If Select does not exhaust the numeration, no derivation is generated [...] The operations Select and Merge are "costless."

Although I agree that the motivation for the operations (i.e., Select and Merge) themselves are conceptually natural, I will put a restriction on the operation SELECT and propose that like any other operations, it can be initiated only by phase heads (i.e., either C or v*). In other words, I assume that the only lexical items in the lexicon that are visible to the initial search by SELECT are phase heads.[9]

A question that immediately arises at this point is, how can non-phase heads then be chosen from the lexicon if phase heads are the only legitimate lexical items that can be accessed by SELECT? I propose the following condition on the operation SELECT to address this issue.

(5) Summoning Condition on SELECT
SELECT can access a non-phase head H only if H is required to satisfy a

---

(ii) the first branching node that dominates X dominates Y.

For different approaches to c-command, see, among others, Epstein (1995) where c-command is derivationally defined as a consequence of the application of Merge.

[9] One might wonder if it is a mere (extras) stipulation that the initial search by SELECT can only see phase heads. If we consider, however, the pivotal roles of phase heads in current minimalism as the initiator of NS operations (e.g., Feature-Inheritance, Transfer), the idea is not much of a stipulation.

selectional feature of a head that has already been introduced into workspace.

Once a phase head is introduced into a workspace by SELECT and non-phase heads are subsequently accessed and introduced into the workspace by the Summoning Condition,[10] EM begins to operate on them so that the Selectors and the heads summoned undergo EM to satisfy selectional features of the Selectors. From this perspective, the function of EM can be taken to construct a structure where all the selectional requirements of a head are satisfied, and what motivates EM is selectional features of the head.[11].

3.3. Projection, Labeling and Restrictions on EM

I follow the basic idea in Chomsky's (2000) claim that "the label is predictable and need not be indicated: the label of selector projects" but propose a more restricted and refined version of Projection/Labeling as follows:

(6) Saturated-Selector Projection
Selector projects but can do so only when the selector no longer has any selectional feature left unsatisfied.

(6) is more restricted than Chomsky's claim above in that a Selector with more than one selectional feature does not project each time one of its selectional features is satisfied. It is more refined at the same time since (6) determines not only what to project but also when to project, i.e., a Selector projects only when all of its selectional features are satisfied.

I take the label of a set to be a signal to the operation EM that shows the set is now ready to participate in further EM as a unit. This implies that a member of a set cannot be accessed by EM if the set already has a label. I assume that it is computationally more efficient and thus more desirable for EM

10 The idea behind the Summoning Condition is not identical but similar to that of Feature-Inheritance (Chomsky 2007, 2008) where T, which is not an inherent probe, is assumed to be able to act as a probe only after it inherits ϕ-features from C.

11 This implies that the Edge-Feature proposed in Chomsky (2007, 2008) is not necessary (at least) for EM in our framework. It further suggests that the expletive *there* in English cannot be introduced by EM as it is not required by a selectional feature as we defined it. In fact, there have been proposals that the expletive *there* is base-generated in Spec-D and subsequently moves to Spec-T to satisfy the EPP-feature of T. See Waller (1997), among others. I will put aside this interesting issue here without further discussion.

to access the label of a set if the set has one rather than to access a member inside the set since in this case, an inside member is more deeply embedded than the label and thus more search is required of EM. I thus propose the following computationally motivated condition on EM that puts a rationale in otherwise stipulated Extension Condition as defined in (3):

(7) Label-over-Member Condition on EM[12]
EM must access the label of a set if the set has one.

Chomsky (2000) claims that "[p]roperties of the probe/selector α must be satisfied before new elements of the lexical subarrary are accessed to derive further operations." Modifying Chomsky's claim, Collins (2002) proposes the following Locus Principle (italics added):

(8) Locus Principle
Let X be a lexical item that has one or more probe/selectors. Suppose X is chosen from the lexical array and introduced into the derivation. Then *the probe/selectors of X must be satisfied before any new unsaturated*[13] *lexical items are chosen from the lexical array.* Let us call X the locus of the derivation.

Let us consider how the above Locus Principle blocks unwanted derivations such as (9a) and (9b) (taken from Collins (2002)):

(9) a. {I′ will {VP John arrive}}
b. (C, {I′ will {VP John arrive}})

Suppose that a derivation reaches the stage in (9a), where EM of I with VP creates I′. Suppose further that at the next stage in (9b), C is chosen and introduced into the workspace. EM of C with I′ is blocked by the Locus Principle (8) since at this stage I′ still has (at least) one more feature to be satisfied (i.e., its EPP feature) and C also has its own features to be satisfied (e.g., its subcategorization feature). In other words, "two unsaturated lexical items [i.e., C and I′] occupy the workspace simultaneously" and therefore, "the

12 A reviewer pointed out how (7) can be reinterpreted in terms of the Labeling Algorithm proposed in Chomsky (2013) where labeling is no longer a prerequisite for entering into computation. A very interesting question but I will put aside this issue for future research.

13 A lexical item that contains at least one unsatisfied probe or selector is unsaturated. (Collins 2002).

derivation is ruled out by the Locus Principle." Adopting the basic idea in Chomsky's claim and Collins'Locus Principle, I propose the following principle:

(10) Repulsion Principle
Two selectors, each bearing one or more unsatisfied selectional features, cannot undergo EM with each other.

Notice that the above Repulsion Principle is a weaker version of Collins' Locus Principle since the former does allow more than one selector with unsatisfied selectional feature(s) to be introduced into the same workspace, whereas the latter preempts this possibility. The result, however, is predictively identical, i.e., they both block the possibility of EM between two selectors each of which bears one or more unsatisfied selectional features ('unsaturated lexical items' in Collins' terms).

Finally, I propose a condition on the interpretation based on the Label Accessibility Condition (LAC) proposed in Epstein, Kitahara and Seely's (2011, henceforth EKS):

(11) Label Accessibility Condition (LAC)
Only the label of an entire syntactic object, the root, is accessible to Narrow Syntax.

EKS (2011) proposes the LAC, arguing that 'LAC itself is deducible since any system must access something and given third factor considerations,[14] access is made with least effort. I assume EKS' LAC without further discussion but modify it to a condition on interpretation at the interfaces as below:

(12) Single Label Condition on Interpretation
In order to be interpreted at the interfaces, an expression must have a single label that dominates all its constituents.

It may seem that the six conditions that I have proposed in this section impose more complexity on NS but as I mentioned at the end of section 1, each of these conditions is either a modification or a specification of an existing condition. Therefore, they will not add more complexity to NS.

---

[14] Chomsky (2005) argues that "the third factor [...] include[s] principles of efficient computation."

To sum up, I list all the six proposed conditions below:

(4) C-commanding Condition on Selection Structure
A Selector must c-command all the lexical items it selects for.

(5) Summoning Condition on SELECT
SELECT can access a non-phase head H only if H is required to satisfy a selectional feature of a head that has already been introduced into workspace.

(6) Saturated-Selector Projection
Selector projects but can do so only when the selector no longer has any selectional feature left.

(7) Label-over-Member Condition on Merge
Merge must access the label of a set if the set has one.

(10) Repulsion Principle
Two selectors, each bearing one or more unsatisfied selectional features, cannot undergo EM with each other.

(12) Single Label Condition on Interpretation
In order to be interpreted at the interfaces, an expression must have a single label that dominates all its constituents.

4. Derivation

Consider now how the selection structure of typical transitive constructions such as *John loves Mary* is built under the conditions I have proposed so far.[15] First, v* is introduced into a workspace by the operation SELECT as we assume that phase heads are the only lexical items visible to the initial search by SELECT.[16] Subsequently, non-phasal lexical items are accessed and introduced into the same workspace under the Summoning Condition (5): V and D_{John} (= EA) are introduced since they both are required by the selectional features of v*; D_{Mary} (= IA) is subsequently introduced into the workspace as it is required by the selectional feature of V. We now have four lexical items in our workspace, namely, v*, V, D_{John}, and D_{Mary}. (13) below lists two of conceivable EMs between these four lexical items[17]:

---

[15] I assume the categorical status of proper nouns (e.g., *John*, *Mary*) to be D and represent them as $D_{Proper\ Noun}$ in tree diagrams.

[16] One may wonder why SELECT chooses v* first rather than C and what regulates the choice. Although it is unlikely that the choice (or order) between v* and C will make any difference, I will limit my discussion to the v*P-domain mainly due to space limitations.

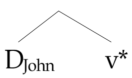
(13) a. EM Option I: {v*, V}, {D_{John}, D_{Mary}}
b. EM Option II: {v*, D_{John}}, {V, D_{Mary}})

Option I is ruled out by the Repulsion Principle in (10). That is, v* and V cannot undergo EM with each other since at this point, each has their own unsatisfied selectional features. D_{John} and D_{Mary} also cannot undergo EM with each other as neither D_{John} nor D_{Mary} carries any selectional features as we defined them. In contrast, no conditions developed so far prevents Option II: v* can undergo EM with D_{John} since the former has an unsatisfied selectional feature (i.e. a feature requiring an External Argument), whereas the latter does not bear any unsatisfied selectional features. In the same vein, V can undergo EM with D_{Mary} because V has its own unsatisfied selectional feature, whereas D_{Mary} does not bear any.[18]. Therefore, the two instances of EM in Option II are legitimate.

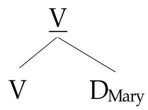
EM between V and D_{Mary} results in projection of the selector V (Condition (6)), whereas EM of v* with D_{John} will not have a label since at this point, the selector v* still has one more selectional feature to be satisfied, namely, its selectional requirement for subcategorization. The structures constructed so far by the two instances of EM look as follows:

(14) First two structures created by EM:

a.



b.

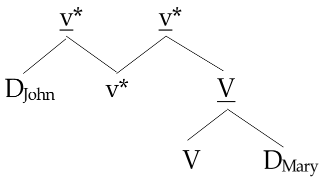


Consider now how the next stage of the derivation proceeds to satisfy the remaining selectional feature of v*. Condition (7) forces v* in (14a) to merge not with V but with $\underline{V}$ in (14b). Once v* undergoes EM with $\underline{V}$, v*now projects to become the label of the outcome since all of its selectional features have now been fully satisfied (Condition (6)). Consequently, the following structure is generated:

[17] We do not assume that there is any order in the two instance of EM in (13a) and (13b), although they are described as if there were an order.

[18] v* and V can undergo EM with different noun phrases. That is, v* merges with D_{Mary} and V with D_{John}. In this case, however, what we get is '*Mary loves John*', not '*John loves Mary.*'

(15)



One noticeable peculiarity about the structure in (15) is that the head v* is immediately dominated by two labels simultaneously. In other words, there is no single node in the structure dominating all the constituents.[19] [20] . This is a clear violation of condition (12).

Let's turn to next section to discuss in more detail the two-peaked structures created by EM in our analysis with respect to (its implications for) the operations Feature-Inheritance and Transfer proposed in Chomsky (2007, 2008).

5. Implications: Feature-Inheritance and Transfer

5.1. Feature Inheritance and Transfer in Chomsky (2007, 2008) and Richards (2007)

Chomsky (2007, 2008, 2013) proposes that all operations in NS are triggered only by uninterpretable features (or probes) of phase heads (i.e., C and v*). He further claims that when the derivation reaches a stage where C merges with T, uninterpretable ф-features on C are passed down to its complement's head T by the mechanism he calls Feature-Inheritance[21], whereas the (uninterpretable) EF of C remains in-situ. Chomsky (2008) deduces the rationale behind Feature-Inheritance from considerations of the C-I interface conditions, arguing that the C-I interface requires NS to structurally distinguish between A- and A′-positions and that Feature-Inheritance is the simplest mechanism that fulfills

---

[19] In set-theoretic notations, the structure in (15) would be represented as follows: {{D_{John}, v*}, {v*, {V, D_{Mary}}}, where v* exists as a member in the two sets simultaneously. I will discuss this issue in more detail in Section 5.2.

[20] This type of two-peaked structure, however, is not unique to our analysis but is also argued to be created in structures generated by countercyclic IM (see EKS 2011). Citko (2005, 2008) also employs two-peaked structures created by her Parallel Merge to better account for the so-called across-the-board wh-questions such as *what did Mary write t_{what} and John review t_{what}*. However, I will not discuss this approach further here.

[21] In other words, ф-features are no longer lexically inherent to T in his system but they are syntactically derivative. He argues the same for the relation between v* and V, that is, ф-features originate from v* and in the course of NS derivation, they are passed down to V by Feature-Inheritance. For expository purposes, however, I will focus on Feature-Inheritance in the C-T domain.

this C-I imposed requirement.

Richards (2007) attempts to find an alternative account of the motivation for Feature-Inheritance on the basis of the following two premises:

(16) Premise 1:
Valuation and Transfer of uninterpretable features must happen together.[22]
Premise 2:
The edge and nonedge (complement) of a phase are transferred separately.

Uninterpretable features must be deleted before they reach the C-I interface. Otherwise, the derivation will crash at the interface. However, once valued, these uninterpretable features are indistinguishable from their matching interpretable counterparts, so if Transfer takes place after valuation, these indistinguishable uninterpretable features cannot be deleted, leading to a crash at the C-I interface. The problem remains the same even if Transfer occurs before valuation since a derivation with transferred unvalued uninterpretable features still crashes at the C-I interface. To tackle this timing dilemma, he argues that 'valuation must be part of Transfer (Premise 1).' In other words, Transfer and valuation takes place simultaneously. Otherwise, no derivation can converge.

Premise 2 states that as soon as all operations in the C phase-level (PH below) have been completed, the complement of the phase head C ('nonedge' in Richards' terms), i.e., TP is transferred to each of the interfaces, whereas the phase head C and its Spec, collectively called "the edge", remain in the workspace and they are carried over to the next phase.

(17) $\boxed{\begin{array}{c}\text{[PH (= CP)} \quad \text{C}_{[u\phi]}\text{]} \\ \text{edge}\end{array}}$ $\boxed{\begin{array}{c}[_{\text{TP}} \text{ Spec} \quad \text{T} \ldots \text{]]} \\ \text{nonedge}\end{array}}$
Transfer

In a framework without C-to-T Feature-Inheritance, uninterpretable ϕ-features (indicated [uϕ] in (17)) would get valued not on T but on C. However, as shown in (17), what is transferred at the point of this feature valuation is not C (or CP) but TP. In other words, uninterpretable ϕ-features on C cannot be transferred at the point of valuation and this leads to a violation of Premise 1. He thus argues that 'feature-inheritance is the only device that can reconcile

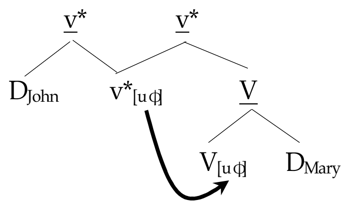
22 Premise 1 was equally pointed out by Esptein and Seely (2002), as Richards notes.

Premise 1 and 2 and thus ensure convergence at the interfaces.' Without Feature-Inheritance, no derivation is ever possible beyond the first phase level.

5.2. Feature-Inheritance and Transferin the v*P-domain

Before discussing how Transfer and Feature-Inheritance can be incorporated into our framework and consequently how Merge can be reinterpreted in our system, let us first consider our final (C-I offending) structure (15), repeated here as (18).

(18)



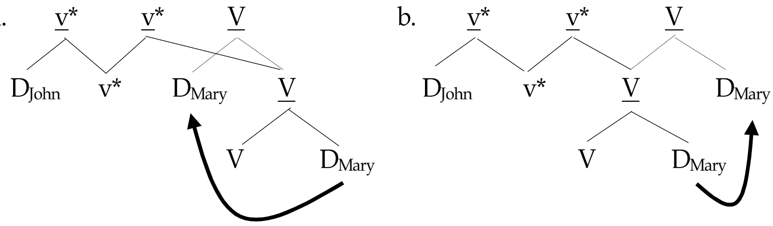
I follow Chomsky (2008) in assuming that (uninterpretable) φ-features, which originate from v*, are inherited by V and they induce an EPP effect. In other words, once φ-features are inherited by V, they trigger movement of D_{Mary} in (18).[23] A question that immediately arises at this point is, where does D_{Mary} move to?

Since the movement of D_{Mary} is triggered not by v* but by V, D_{Mary} must be somehow merged with V. However, this movement has nothing to do with selectional features of V (hence, the name Extended Projection Principle). In other words, the movement of D_{Mary} is not driven to satisfy 'selectional' requirements of V itself but rather, if we adopt Chomsky's (2007, 2008) Feature Inheritance, it is a requirement that is added to V by v* in the course of the derivation the requirement is not inherent to V. Therefore, it should be natural to assume that the movement of D_{Mary} need not abide by our Condition (4) which requires the selector to c-command all the lexical items it selects for, i.e., IM of D_{Mary} is exempt from Condition (4). Where does it then move to? Below are some of the conceivable landing sites for D_{Mary}.

---

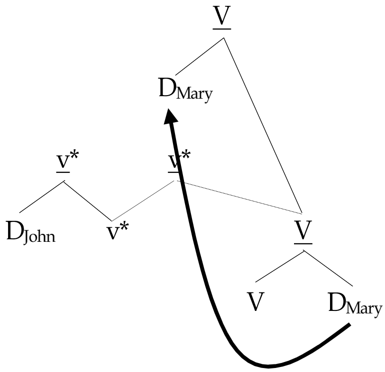
[23] It is not clear, though, whether EF (on v* and/or V) is also involved in the movement of D_{Mary}. It might be the case that unlike EF on v*, EF on V is somehow defective in that it cannot attract D_{Mary} without the aid of φ-features but it can (or must) once φ-features are inherited by V. Interesting but I will leave this issue open. See, among others, Ouali (2008) regarding what can happen to the features on a phase head after they are inherited by its complement head.

(19) a. b.



In (19a), where D_{Mary} moves to the Spec-V position, the movement of D_{Mary} creates even more peaks so that the resulting structure still cannot be interpreted at C-I interface (Condition (12)). The situation does not improve in (19b), where D_{Mary} moves rightward.[24] Below is the structure that I propose is created after the movement of D_{Mary}:

(20)



As discussed in Section 4, the head v* in (20) is immediately dominated by two labels simultaneously. In set-theoretic notations, the status of v* before the movement of D_{Mary} can be represented as below in (21a), where v* occurs as a member of two sets simultaneously[25]:

(21) a. $\{\{D_{John}, v^*\}, \{v^*, \{V, D_{Mary}\}\}\}$ Before movement of D_{Mary}

b. $\{D_{John}, v^*\}, \{D_{Mary}, \{\cancel{v^*}, \{V, D_{Mary}\}\}\}$ Movement of D_{Mary}

c. $\{D_{John}, v^*\}, \{D_{Mary}, \{V, D_{Mary}\}\}$ After movement of D_{Mary}

What the movement of D_{Mary} to Spec-V does in (20) (and in (21)) is to eliminate the existing relation between the head v* and the label V (of $\{V, D_{Mary}\}$). What

24 In fact, (19a) and (19b) are exactly the same from the perspective of C-I if we adopt Chomsky's (2008) claim that "order does not enter into the generation into the C-I interface."

25 For expository purposes, labels are not indicated in (21).

this means in set-theoretic terms is that IM of D_{Mary} eliminates one of the two occurrences of v*, that is, v* from the set {v*, {V, D_{Mary}}} as shown in (21b) and consequently, the structure in (21c) is created after IM of D_{Mary}. I take this eliminative operation by IM whereby a member of a set gets eliminated to be a trigger for the operation Transfer. In other words, Transfer gets activated via IM, transmitting the structure where IM has taken place to the interfaces. This in turn implies that Transfer occurs only when this type of eliminative IM takes place.

This type of relation-breaking (or member-deleting) IM may seem to violate the No Tampering Condition (NTC) proposed in Chomsky (2005, 2007, 2008) because it involves modifying the existing structure by eliminating the (existing) relation between the head v* and the label V. If we consider the following claims in Chomsky (2008), however, this type of IM is not unjustified:

(22) No Tampering Condition (NTC)
Merge of X and Y leaves two SOs unchanged.

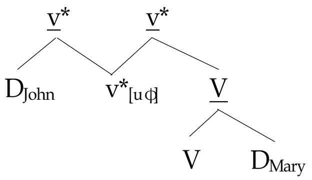
(23) Strong Minimalist Thesis (SMT)
Language is an optimal solution to interface conditions that FL [the Faculty of Language] must satisfy.

(24) SMT might be satisfied even where NTC is violated - if the violation has a principled explanation in terms of interface conditions (or perhaps some other factor).

An expression must have a single label for it to be interpreted at the interfaces (see Condition (12)) but we have just seen that the offending structure in (18) has no other alternatives to satisfy this interface-driven condition (hence conforming to SMT in (23)) than to remove either of the two peaks. Furthermore, IM we propose is not the only operation that violates NTC defined in (22). Take, for example, feature-inheritance from v* to V and subsequent AGREE between V and a noun phrase. The former operation adds new features (i.e. ϕ-features) to V, modifying (the existing) featural specifications of V. The latter also changes the featural values of ϕ-features added to V since the operation AGREE renders unvalued ϕ-features of V valued by those of a noun phrase. Therefore, IM with a built-in eliminative ability can be justified (at least conceptually) even if it violates NTC.

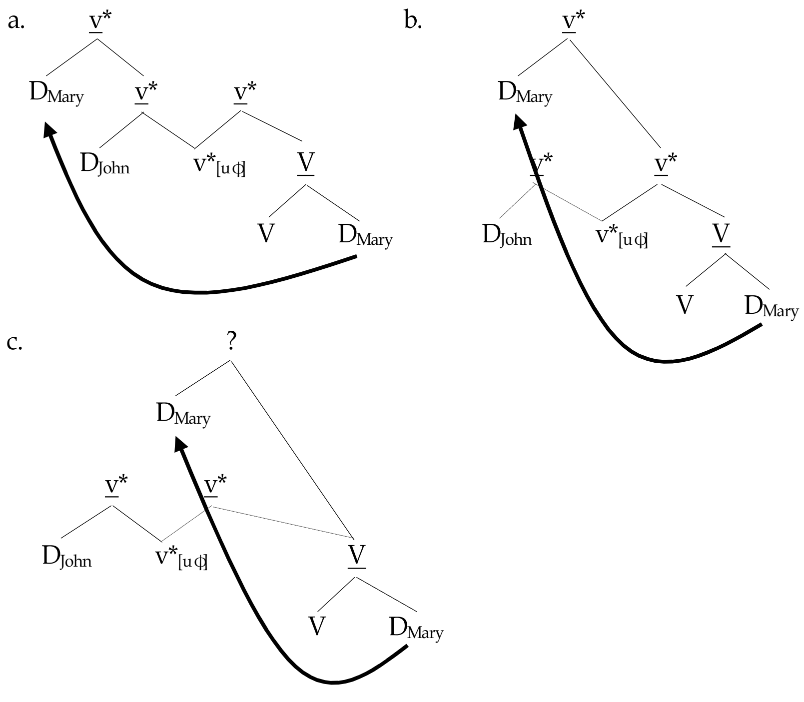
Now, let us examine implications of our version of IM with a built-in eliminative operation as concerns the motivation for the mechanism Feature-Inheritance. Consider first the structure prior to the movement of D_{Mary}:

(25)



Without the operation Feature-Inheritance, uninterpretable φ-features would stay on v* and raise D_{Mary} to its Spec. Because the movement of D_{Mary} is triggered by unintepretable features on v*, D_{Mary} must somehow merge with v*. Below are three conceivable structures that can be created by the movement of D_{Mary}.

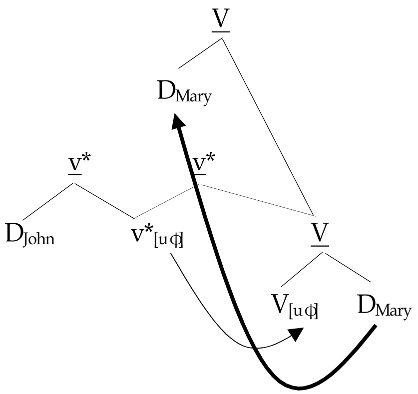
(26)



(26a) shows that D_{Mary} moves to Spec-v*. Notice, however, that IM of D_{Mary} does not involve any eliminative process and thus no structure can get transferred. Therefore, the derivation crashes due to a violation of Condition (12), i.e., it (still) does not have a single label that dominates all the lexical items. (26b), where D_{Mary} moves by eliminating the relation between D_{John} and v*, is also ruled out as D_{John} cannot participate in further EM and thus it will eventually reach the C-I interface without having its uninterpretable Case feature valued. (26c), where

D_{Mary} moves by eliminating the relation between v* and V and becomes Spec of V, is the most problematic derivation. As mentioned above, D_{Mary} is required by v*, not by V, and therefore, it must be connected to v*. However, this is not the case in (26c). Furthermore, if we adopt the idea of what projects is always the Selector, it is not clear how the projection would work in (26c). If we assume ϕ -feature-inheritance by V, however, all the problems found in the three derivations above disappear in (20), repeated here as (27):

(27)



In (27), ϕ-features of v* are inherited by V. What this means is that D_{Mary} must be connected with V since the requirement for D_{Mary} now resides on V. Once D_{Mary} moves to Spec-V via our eliminative IM, V projects to become the label of the outcome as V is the Selector. In this application of Merge, we now can deduce the necessity of Feature-Inheritance, not from considerations of timing between valuation and Transfer as Richards (2007) suggests and Chomsky (2007) later adopts, but from considerations of interface conditions, i.e., Narrow Syntax conforms to Interpretation Condition (15) imposed by the C-I interface even by eliminating a member from the structure(i.e., eliminative IM) and thus violating NTC: language is indeed an optimal solution to interface conditions.

I conclude this section with a modified definition of Merge:

(28) Merge

Merge takes two syntactic objects (SOs), α and β, to form a set $\{\alpha, \beta\}$. In doing so, Merge can modify an existing relation if the modification is required by interfaces.

6. Conclusion and Remaining Issues

In this paper, I proposed that along with other operations in NS, structure-building can also be initiated only by phase heads and showed that this type of phase-head initiated structure-building inevitably creates a C-I uninterpretable structure with no single label dominating all the constituents in the v*P-domain. To resolve this dilemma, i.e., phase-head initiated structure building vs. two-peaked structure in the v*P-domain created by phase-head initiated structure building, I proposed that IM can eliminate a member from a set to satisfy interface conditions and that this eliminative IM gets the operation Transfer activated. Finally, I explored the implications of phase-head initiated structure building and eliminative IM for both Feature-Inheritance and Transfer and showed that both operations can be better motivated in phase-head initiated structure building.

However, there remain some issues I will leave open here. One issue is concerned with the structure of ditransitive verbs such as put and give. Unlike typical transitive verbs, these ditransitive verbs will presumably create a three-peaked structure. For now, however, I have little to say how this three-peaked structure can be remedied by our eliminative (Internal) Merge. Another issue is concerned with the asymmetric c-command relation between the external argument and the internal argument. As we saw in previous sections, however, the external argument does not asymmetrically c-command the internal argument in our system. In fact, there seems to be no c-command relation between the two. I will leave all these interesting questions for future research.

References

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

Chomsky, N. 2000. Minimalist Inquiries: the Framework. In R. Martin et. al. eds., *Step by Step: Essays on Minimalism in Honor of Howard Lasnik*, 89-115. Cambridge, Mass.: MIT Press.

Chomsky, N. 2005. Three Factors in Language Design. *Linguistic Inquiry*, 36: 1-22.

Chomsky, N. 2007. Approaching UG from Below.In U. Sauerland and H.-M.Gärtner, eds., *Interfaces + Recursion = Language?*, 1-30. Berlin: Mouton de Gruyter.

Chomsky, N. 2008. On Phases. In R. Freiden, et. al. eds., *Foundational Issues in Linguistic Theory: Essays in Honor of Jean-Roger Vergnaud*, 89-155. Cambridge, Mass.: MIT Press.

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

Citko, B. 2005. On the Nature of Merge: External Merge, Internal Merge, and Parallel Merge. *Linguistic Inquiry*, 36: 475-496.

Citko, B. 2008. More Evidence for Multidominance. Talk given at the conference on Ways of Structure Building, University of the Basque Country.

Collins, Chris. 2002. Eliminating Labels. In Samuel D. Epstein and Daniel Seely, eds., *Derivation and Explanation in the Minimalist Program*. Oxford: Blackwell Publishing.

Epstein, S. 1995. Un-principled and the Derivational Syntactic Relations. ms. Harvard University.

Epstein, S. and D. Seely. 2002. Rule Applications as Cycles in a Level-free Syntax. In S. Epstein and D. Seely, eds., *Derivation and Explanation in the Minimalist Program*. Oxford: Blackwell.

Epstein, S., H. Kitahara and D. Seely.2011. Structure Building That Can't Be. In D. Adger and H. Borer, eds., *The Oxford Studies in Theoretical Linguistics*. Oxford: Oxford Press.

Ouali, H. 2008. On C-to-T φ-feature Transfer: The Nature of Agreement and Anti-Agreement in Berber. In R. d'Alessandro, G. H. Hrafnbjargarson and S. Fischer eds., *Agreement Restrictions*. Berlin: Mouton de Gruyter.

Richards, M. 2007. On Feature Inheritance: An Argument from the Phase Impenetrability Condition. *Linguistic Inquiry*, 38: 563-572.

Waller, B. 1997. Towards a Proper Characterization of English there. Presented in Workshop in Syntax/Semantics, MIT.

Jae-Young Shim
Linguistics Department,
440 Lorch Hall, 611 Tappan Street
University of Michigan
Ann Arbor, MI 48109-1220
USA

jaeshim@umich.edu

Received: 2014. 5. 13.
Accepted: 2014. 5. 22.