External Merge by Phase: Its Implications for Feature-Inheritance, Transfer and Internal Merge

Jae-Young Shim[*]
University of Michigan

1. Introduction

Chomsky (2008) argues that "[…] along with Transfer, all other operations will also apply at the phase level […] that implies that IM [Internal Merge, henceforth I-Merge] should be driven only by phase heads [i.e. C and *v**]." Chomsky (2013) further specifies "all other operations" by explicitly addressing the status of External Merge (henceforth, E-Merge) with respect to the trigger of the operation (italics are mine):

> One plausible general principle is that operations are restricted to the phase level – with the *exception of EM* which is required to construct the phase in the first place: hence IM, agreement, and the operation of transfer to the interfaces.

It seems hardly the case that a sole exception to a theoretical generalization necessarily invalidates the entire theory but it can nevertheless serve as a favorable opportunity to reevaluate the generalization in the theory or at the least, to reassess the exception itself.

The aim of this paper is two-fold. First, I will explore an alternative way E-Merge proceeds to show that E-Merge is *not* an exception with respect to the trigger of the operation. Consequently, I will argue, *contra* Chomsky's (2013) claim above, that along with Transfer and I-Merge, E-Merge is also initiated only by phase heads. Second, I will examine implications of the claim that E-Merge is also initiated only by phase heads concerning the formulation of Feature-Inheritance and Transfer proposed in current minimalism (Chomsky (2007, 2008), Epstein et al. (2011), Richards (2007)) and I will redefine the operation Merge accordingly.

The paper is organized as follows: In section 2, I briefly overview structure building in the *v**P-phase level (implicitly) assumed in Chomsky

[*] 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.

(1995 *et seq*.) and point out 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 E-Merge can also be initiated only by phase heads. In section 5, I explore implications of the claim made in section 4 for Feature-Inheritance and Transfer. Section 6 concludes the paper.

2. Structure Building

2.1 Conventional Derivation

In minimalism (Chomsky 1995 *et seq*.), E-Merge has been assumed to proceed in a bottom up fashion[1] by recursive application of a (two-membered) set-forming operation called Merge[2] defined below as in (1).

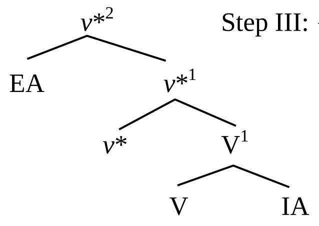
(1) **Merge**
Merge takes two syntactic objects (SOs), α and β, to form a set {α, β}.

I follow Chomsky (2000) in assuming that the label of the selector projects to become the label of the outcome of Merge. According to this assumption, when α and β undergo Merge with each other as in (1) above, either α or β becomes the label of the outcome, i.e. if the selector is α, then α projects to become the label, giving us {α, {α, β}}, whereas if the selector is β, then β becomes the label, giving us {β, {α, β}}.[3]

At some point in the derivation, the following structure in (2) is generated by recursive application of Merge, where EA and IA refer to External and Internal Argument, respectively.[4]

1. If the operation Merge, whether Internal or External, proceeds in a ‘top-down’ fashion, the resulting structure eventually violates the Extension Condition defined in Section 2.2.
2. Chomsky (2007, 2008) argue that ‘mergeability of a lexical item with another lexical item stems from an inherent property, called the Edge-Feature (EF), that any lexical item has.’ In other words, it is the (inherent) EF of a lexical item that enables the lexical item to undergo Merge with another lexical item.
3. See Section 3.4 for more on this assumption. For different approaches to labeling, see Chomsky (2013) and Collins (2002), among others.
4. C-/T-domains and many other operations such as the movement of EA to Spec-T are omitted here for simplicity but both will be discussed in Section 5. Throughout the paper, scripted numerals on a category (e.g. *v**1) are used only for expository purposes to distinguish between a head and the label of a set with the head as its member.

(2)



Step III: $\{v^{*2}, \{\text{EA}, \{v^{*1}, \{v^*, \{\text{V}^1, \{\text{V}, \text{IA}\}\}\}\}\}\}$
↑
Step II: $\{v^{*1}, \{v^*, \{\text{V}^1, \{\text{V}, \text{IA}\}\}\}\}$
↑
Step I: $\{\text{V}^1, \{\text{V}, \text{IA}\}\}$

In Step I above, V and the complement, IA[5], undergo Merge to form a set, {V, IA}, and V projects to become the label of the set.[6] Then, in Step II, *v** is introduced and merges with the existing set, $\{\text{V}^1, \{\text{V}, \text{IA}\}$, to form another set, $\{v^*, \{\text{V}^1, \{\text{V}, \text{IA}\}\}\}$, *v** becoming the label of the outcome. Finally, in Step III, EA is introduced and undergoes Merge with the set, $\{v^{*1}, \{v^*, \{\text{V}^1, \{\text{V}, \text{IA}\}\}\}\}$, to form yet another set, $\{v^{*2}, \{\text{EA}, \{v^{*1}, \{v^*, \{\text{V}^1, \{\text{V}, \text{IA}\}\}\}\}\}\}$, *v** becoming the label of the resulting set.

The question I would like to address in this section concerns the very first step in the above (conventionally-assumed) derivation where V and IA undergoes E-Merge with each other in the workspace (I will call this issue ‘the problem of genesis’ since E-Merge between V and IA is concerned with ‘the very beginning’ of any derivation). The question is: among many other lexical items (i.e. heads) in the lexicon, why is it that V is first chosen from the lexicon and introduced into workspace? In other words, how does syntax know such is (or must be) the case? These issues are discussed in the next section.

2.2 Problem of Genesis

One possible answer to the question raised in 2.1 might be to assume that syntax somehow knows that the derivation will crash (or will be interpreted as gibberish)[7] if it makes other choices since they will all

5. If the 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.

6. Although (E-)Merge is said to take place between V and IA in Step I, it seems that one must be introduced into the workspace before the other given the selectional relation between V and IA. That is, the introduction of V seems to have to precede that of IA since it is V that selects IA, not vice versa.

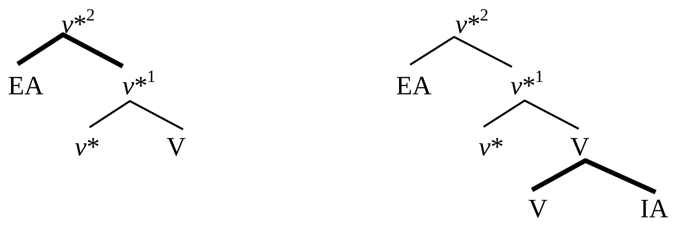
7. A derivation crashes if it violates conditions of interfaces. For example, if a derivation reaches an interface with one or more unchecked (or unvalued) uninterpretable features, then it crashes at the interface. However, if a derivation fully satisfies interface conditions but is semantically nonsensical, it is regarded as (a

eventually 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.

To take an example, let us suppose that for generation of a transitive construction such as *John killed the bug*, *v** instead of V is chosen first from the lexicon and put into the workspace. Suppose further that V is subsequently chosen from the lexicon and undergoes Merge with *v** to satisfy a selectional feature[8] of *v**. At this point, we need two more heads to satisfy the remaining selectional features of *v** and V, namely, a D/N (= EA) for *v** and another D/N (= IA) for V. However, the Extension Condition above prevents IA from undergoing E-Merge with the existing structure. To see how the Extension Condition prevents this E-Merge, let's suppose that E-Merge between EA and $v*^1$ precedes that of IA with V,[9] leading to the structure in (4a).

(4) a. E-Merge between EA and $v*^1$ b. E-Merge between IA with V



Conforming to the Extension Condition *Violating* the Extension Condition

The Extension Condition does not prevent E-Merge between EA and $v*^1$ as shown in (4a) because this E-Merge extends the existing structure (i.e.

semantic) gibberish. A famous example of such semantic nonsense is the sentence "colorless green ideas sleep furiously" (Chomsky 1957). The sentence is perfectly grammatical (i.e. it does not violate interface conditions) but it is a semantic nonsense.

8. Selectional features of a head H include features either for (thematic) argument(s) that H takes or for a head that H subcategorizes for. This will be discussed in more detail in Section 3.

9. Reversing the order does not make any difference with respect to the violation of the Extension Condition as long as E-Merge of IA with V follows E-Merge of *v** with V.

{*v**[1], {*v**, V}}). However, the next step in (4b), where V and I undergo E-Merge with each other, would lead to a violation of the Extension Condition since this E-Merge would not extend the existing structure[10].

It is not just *v** as shown in (4) but all choices other than V will eventually create a structure that would lead to a violation of the Extension Condition. However, the assumption 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 runs into a problem because it invokes look-ahead properties.

Another possible answer to the problem of genesis 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)[11], and then EA (= D/N)), which will all ultimately lead to a violation of the Extension Condition). If we accept that C_{HL} is governed by the principle of computational efficiency[12], this type of answer putting more computational burden on C_{HL} does not seem satisfactory, either.

2.3 Exceptional Status of E-Merge

Let us consider the following proposals Chomsky (2008) makes regarding I-Merge with respect to phase heads (italics are mine):

> 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* [except for E-Merge] …

If all operations are indeed triggered only by phase heads as he suggests above, it should be natural to assume that E-Merge is not an exception to

10. Chomsky (2000) argues that E-Merge of IA as in (4b) violates the Extension Condition but satisfies a "reasonable condition" which he calls Local Merge.
11. In fact, it is not clear how a derivation proceeds if the first chosen lexical item is IA.
12. According to Chomsky (2013), "[w]e don't have a comprehensive theory of computational efficiency, but quite a few things are pretty obvious that are going to enter into such a theory. One, for example, that less is better than more."

this generalization as, after all, it is an operation. Chomsky (2013), however, explicitly states that quite otherwise is the case (see the quote from Chomsky (2013) in Introduction on page 1).

Claiming that there is an exception to a generalization does not necessarily mean that the generalization is not true. Rather, as Chomsky (2013) suggests, it shows that the exception is “a valuable stimulus” to “further inquiry” and natural steps towards such further inquiry should include revaluation of the nature of the exception and reexamination of the generalization accordingly. This is what I will discuss in next section.

3. External Merge by Phase

I will first clarify technical terms for my alternative account of how E-Merge proceeds. Then, I will discuss the conditions that I will employ for my account.

3.1 Selectional Features/Requirements

I will assume a group of selectional features distinct from the rest of the features that a head can bear. To be more specific, selectional features of a head H include: 1) features for (thematic) argument(s) that H takes and 2) features for another head that H subcategorizes for[13]. 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 noun phrase (as its complement)[14], whereas the head *v** has two, i.e. the feature that requires a noun phrase for its argument and V for its subcategorization. I will call a head H with these selectional features a “Selector” and indicate it informally as $H_{[\text{I want X}]}$.

I will 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 head must be satisfied before a derivation reaches each of the interfaces.

These uninterpretable selectional features of a head are different from other uninterpretable features such as φ-features of a head in that the former are always satisfied by E-Merge[15]. In other words, there is no such long-

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

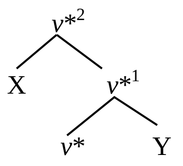
14. I will put aside the situation where this type of V takes a clause as its complement since the issue here is not concerned with the categorial status of complement.

15. If one adopts movement to a theta position as proposed in Horstein (1999), selectional features can be argued to be satisfied by I-Merge as well. However, I will

distance search (i.e. search beyond one’s own domain) in satisfaction of selectional features as found in φ-feature agreement. Further, if we assume the theoretical device called feature-inheritance[16] developed in Chomsky (2007, 2008), selectional features of a head, unlike φ-features, are never inherited by the head of its complement.

3.2 Configurational Asymmetry in Selection Structure

Assuming the conventional way E-Merge proceeds as illustrated in (2) above, let us consider the following to see what type of structure is generated when selectional features of a head are satisfied.

(5)


Both X and Y are lexical items that are selected by the head *v**. In other words, they both are externally merged to satisfy the selectional features of *v**, X for argument and Y for subcategorization. Notice that there exists a configurational asymmetry between X and Y in relation to the Selector *v** despite the fact that both X and Y are chosen from the lexicon to satisfy the same class of features of *v** that I call selectional features. The nature of the asymmetry is that Y is c-commanded by the Selector *v** but X is not. Notice further that this type of configurational asymmetry is found only in satisfaction of *v**’s selectional features simply because *v** is the only head that bears both subtypes of selectional features, i.e. selectional features for argument and for subcategorization.[17] All other heads such as T and V have selectional features only for subcategorization.

If it is true that features for argument and for subcategorization indeed belong to the same group of features (i.e. selectional features), it should be (at least) conceptually natural to assume that these features are satisfied in

---

not discuss this issue further here.

16. The mechanism, feature-inheritance, will be defined and discussed in more detail in Section 5.

17. I-Merge creates the same asymmetry as discussed above since I-Merge is always to the Spec position of a head. When a *wh*-phrase moves to Spec of C, for example, it is not c-commanded by C, whereas the complement of C (i.e. T) is. However, in the case of I-Merge, features involved are not selectional features as we defined them above. I will discuss this issue in more detail in Section 5.

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):

(6) **C-commanding Condition on Selection Structure**
A Selector must c-command *all* the heads it selects for.

Condition (6) 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.

3.3 The Operation SELECT and the Summoning Condition

Following Chomsky (1995), I will call SELECT the operation that accesses the lexicon, chooses a head, and puts it into the workspace. The only difference between our SELECT and Chomsky's is that ours 'directly' accesses the lexicon, whereas Chomsky's accesses an intermediate buffer called Numeration where all the heads to be used for a derivation are first placed.[18] However, 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 SELECT and propose that like any other operations, it can be initiated only by phase heads (i.e. either C or *v**).[19] In other words, I assume that the only lexical items in the lexicon visible to the initial search by SELCT are phase heads.

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.

18. It is not clear, however, what mechanisms are involved in forming a Numeration.
19. The choice between C and *v** does not matter as long as the derivation meets the condition of Full Interpretation (Chomsky 1995) that requires a derivation contain only interpretable features to converge at each of the interfaces. See Section 5.2 for more on this issue.

(7) **Summoning Condition on SELECT**
SELECT can access a non-phase head H only if
(i) H is required to satisfy a selectional feature of a phase head itself, or
(ii) H is required to satisfy a selectional feature of another non-phase head that has already been introduced into workspace.

The structure created by SELECT with the Summoning Condition above will look similar to the set called Numeration (or Lexical Array (Chomsky 2000)) in that SELECT will eventually introduce into the workspace all the necessary heads for a derivation (see Section 4 below) but unlike the Numeration (or Lexical Array), the heads introduced by SELECT do not form an independent set on their own; they are directly put into workspace.

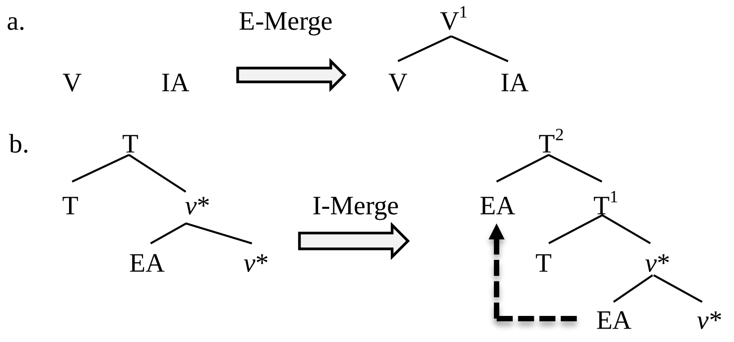
Once a phase head is introduced into a workspace by SELECT and non-phase heads are subsequently accessed and introduced into the workspace by the Summoning Condition (i) and (ii), E-Merge begins to operate on them to merge the Selectors with the heads summoned in order to satisfy selectional features of the Selectors. From this perspective, the function of E-Merge can be taken to construct a structure where all the selectional requirements of a head are satisfied, and what motivates E-Merge is selectional features of the head[20].

3.4 Projection, Labeling and Restrictions on E-Merge

In this section, I will first examine two types of Merge, E-Merge (8a) and I-Merge (8b), with respect to Projection/Labeling proposed in Chomsky (2000) and based on this examination, I will propose a modified version of Projection/Labeling. Then, I will examine the Extension Condition in Chomsky (2000) and Collins' (2002) Locus Principle, proposing two conditions on E-Merge. Consider first the following structures:

20. This implies that the Edge-Feature proposed in Chomsky (2007, 2008) is not necessary (at least) for E-Merge in our framework. It further implies that the expletive *there* in English cannot be introduced by E-Merge 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.

(8) a.



In (8a), E-Merge creates a set containing V and IA to satisfy the selectional feature of V and the Selector V becomes the label of the outcome (i.e. {V, IA} becomes {V^1, {V, IA}}). (8b) shows an example of I-Merge where EA moves to SPEC-T to satisfy the EPP-feature of T. Here again, the label of the outcome is the Selector T with the EPP-feature requiring EA. From this observation, we can argue that when two lexical items undergo (E-/I-)Merge with each other, it is always the Selector that projects to become the label of the outcome. 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:

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

(9) 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 and it is more refined since (9) 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 E-Merge that shows the set is now ready to participate in further E-Merge as a unit. This implies that a member of a set cannot be accessed by E-Merge if the set already has a label. I assume that it is computationally more efficient and thus more desirable for E-Merge 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 E-Merge. I thus propose the following computationally motivated condition on E-Merge that puts a rationale in otherwise stipulated

Extension Condition:

(10) **Label-over-Member Condition on E-Merge**
E-Merge 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 are mine):

(11) **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*[21] *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 (12a) and (12b) (taken from Collins (2002)):

(12) a. {I' will {VP John arrive}}
b. (C, {I' will {VP John arrive}})

Suppose that a derivation reaches the stage in (12a), where (E-)Merge of I with VP creates I'. Suppose further that at the next stage in (12b), C is chosen and introduced into the workspace. (E-)Merge of C with I' is blocked by the Locus Principle since at this stage I' still has (at least) one more feature to be satisfied (e.g. 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 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:

(13) **Repulsion Principle**
Two Selectors, each bearing one or more unsatisfied selectional features, cannot undergo E-Merge with each other.

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

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 E-Merge between two Selectors each of which bears one or more unsatisfied selectional features ('unsaturated lexical items' in Collins' terminology).

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

(14) **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[22], access is made with least effort. I assume EKS' LAC without argument but modify it to a condition on interpretation at the interfaces as below:

(15) **Single Label Condition on Interpretation**
An expression must have a single label to be interpreted at the interfaces.

It may seem that the six conditions that I have proposed in this section impose more complexity on Narrow Syntax 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 do not add more complexity to Narrow Syntax.

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

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

(7) **Summoning Condition on SELECT**
SELECT can access a non-phase head H only if
(i) H is required to satisfy a selectional feature of a phase head itself, or
(ii) the head is required to satisfy a selectional feature of another non-phase head that has already been introduced into workspace.

22. Chomsky (2005) argues that "[…] the third factor […] includ[es] principles of efficient computation."

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

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

(13) **Repulsion Principle**
Two Selectors, each bearing one or more unsatisfied selectional features, cannot undergo E-Merge with each other.

(15) **Single Label Condition on Interpretation**
An expression must have a single label to be interpreted at the interfaces.

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.[23] First, *v** is introduced into a workspace by SELECT as we assume that phase heads are the only lexical items visible to the initial search by SELECT.[24] Subsequently, non-phasal lexical items are introduced into the same workspace under the Summoning Condition (7): under (7i), V and D_{John} (= EA) are introduced since they both are required by the selectional features of *v** and under (7ii), 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}. (16) below lists two of conceivable E-Merges between these four lexical items[25]:

(16) a. E-Merge between *v** and V (i.e., {*v**, V}) and
E-Merge between D_{John} and D_{Mary} (i.e. {D_{John}, D_{Mary}})
b. E-Merge between *v** and D_{John} (i.e. {*v**, D_{John}}) and
E-Merge between V and D_{Mary} (i.e. {V, D_{Mary}})

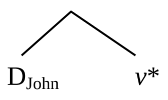
(16a) is ruled out: *v** and V cannot undergo E-Merge with each other since each has their own unsatisfied selectional features (see Repulsion

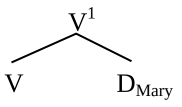
23. I assume the categorial status of proper nouns (e.g., John, Mary) to be D and represent them as DProper Noun in tree diagrams.
24. One may ask why SELECT chooses *v** first rather than C and what regulates the choice. For this, see Section 5.3.
25. We do not assume that there is any order in the two instances of E-Merge in (16a) and (16b), although they are described as if there were ordered.

Principle in (13) above). D_{John} and D_{Mary} cannot undergo E-Merge with each other either as neither D_{John} nor D_{Mary} carries any selectional features as we defined them. In contrast, no conditions developed so far prevents E-Merge in (16b): *v** can E-Merge with D_{John} since the former has an unsatisfied selectional feature (i.e. feature requiring the External Argurment), whereas the latter does not bear any unsatisfied selectional features. In the same vein, V can undergo E-Merge with D_{Mary} because V has its own unsatisfied selectional feature, whereas D_{Mary} does not bear any.[26] Therefore, the two instances of E-Merge in (16b) are legitimate.[27]

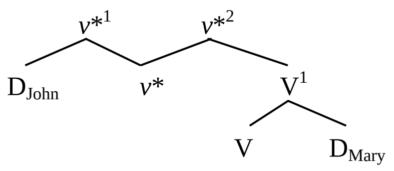
E-Merge between V and D_{Mary} results in projection of the Selector V (Condition (9)), whereas E-Merge of *v** with D_{John} will not have a label since the Selector *v** still has one more selectional feature to be satisfied, namely, its selectional requirement for subcategorization. The structures constructed so far by the two instances of E-Merge look as follows:

(17) First two structures created by E-Merge:

a. 

b. 

Consider now how the next stage of the derivation proceeds to satisfy the remaining selectional feature of *v**. Condition (10) forces *v** in (17a) to merge not with V but with V^1 in (17b). Once *v** undergoes E-Merge with V^1, *v** now projects to become the label of the outcome since all of its selectional features have now been satisfied (Condition (9)). Consequently, the following structure is generated:

(18) 

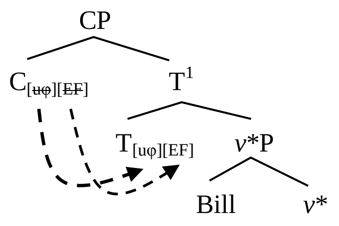
One noticeable peculiarity about the above structure is that the head *v**

26. *v** and V can undergo E-Merge with different noun phrases. That is, *v** merges with DMary and V merges with DJohn. In this case, however, what we get is '*Mary loves John*', not '*John loves Mary*.'

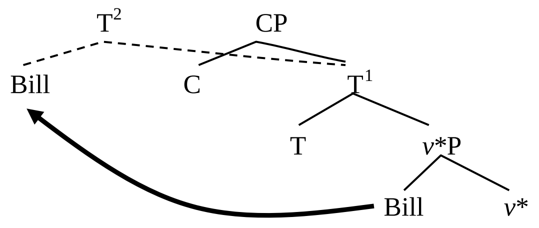
27. Another possibility is that both *v** and V undergo E-Merge with the same noun phrase, that is, either with DJohn or with DMary. I will discuss this issue in more detail in Section 5.3.

is immediately dominated by two labels simultaneously, namely, *v**^{1} and *v**^{2}. In other words, there is no single node in the structure dominating all the constituents.[28] 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 I-Merge by EKS (2011).[29] Consider the following structure to see how two-peaked structures are generated by I-Merge in EKS (2011):

(19) a.



b.



Following Chomsky (2007, 2008), EKS assume that (uninterpretable) φ-features on C are transmitted to T. They further assume that Edge-Feature (EF) on C is also transmitted to T as shown in (19a). As a result, T bears double EF, one inherent to it and the other inherited from C, and this is argued to induce an EPP effect to raise the subject *Bill* as in (19b).[30] In EKS's analysis, however, the subject *Bill* cannot be countercyclically infixed to create a Spec-position in T because the creation of a Spec-T position after C has been E-Merged to T violates their LCR condition defined as in (20):

28. In set-theoretic notation, the structure in (18) would be represented as follows: {DJohn, *v**}, {*v**, {V, DMary}, where *v** exists as a member in two sets simultaneously. I will discuss this issue in more detail in Section 5.2.
29. 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 twhat and John review twhat*. However, we will not discuss this approach further here.
30. It is not clear in EKS (2011) what mechanism enables a single noun phrase to delete two EFs (i.e. double EF) and how C without EF can attract a *wh*-phrase.

(20) The Law of Conservation of Relations (LCR)
In N[arrow]S[yntax], syntactic relations (among existing items) cannot be altered throughout a derivation.

To satisfy the EPP (i.e., double EF on T) and also observe LCR, the subject *Bill* must move to Spec-T, creating a two-peaked structure as shown in (19b). As a result, T^1 is now immediately dominated by both CP and T^2. Since there is no single label in (19b), then given EKS' LAC in (14), NS cannot access the structure and consequently, the derivation must halt.

To address this problem, EKS propose that the operation Transfer can rescue the halted derivation by removing one of the two peaks, i.e. T^2, from NS so that the derivation can continue.[31] EKS thus seek to deduce the necessity, timing and category undergoing Transfer from geometric abnormality, i.e. the two-peaked configuration created by countercyclic I-Merge of the subject to Spec-T.

Let's turn to next section to discuss two-peaked structures created by E-Merge in our analysis in more detail with respect to its implications for the operation Transfer as well as Feature-Inheritance proposed in Chomsky (2007).

5. Implications: Feature-Inheritance and Transfer

I will first briefly overview the operation Transfer and Feature-Inheritance developed in Chomsky (2007, 2008). Then, I will overview Richards' (2007) argument against Chomsky (2008) with respect to the motivation for Feature-Inheritance.[32] Finally, I will address the implications of our analysis for Feature-Inheritance and Transfer and redefine the operation Merge.

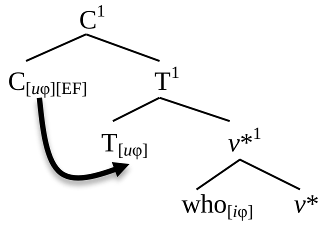
5.1 Feature Inheritance and Transfer in Chomsky (2007, 2008) and Richards (2007)

Chomsky (2007, 2008, 2013) proposes that all operations in Narrow Syntax are triggered by uninterpretable features (or probes) of phase heads (i.e. C and v*) and that E-Merge is an exception to this generalization. He further claims that when the derivation reaches a stage where C is E-Merged with T, uninterpretable φ-features on C, represented as [*u*φ] in the tree diagram (21) below, are passed down to its complement's head, T, by the

31. They are not clear about what ensures the removal of T2, not of CP.
32. Chomksy (2007) adopts Richards' (2007) argument on the motivation of feature-inheritance.

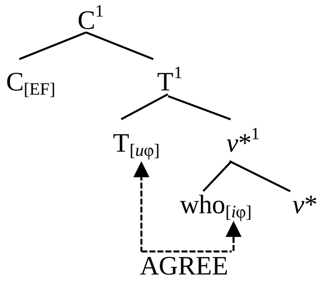
mechanism he calls Feature-Inheritanc[33], whereas the (uninterpretable) EF of C remains in-situ.

(21)



After inheriting φ-features from C[34], T can now act as a probe seeking a goal with corresponding but valued interpretable φ-features, represented as [*i*φ] above, in its c-command domain. If such a matching goal is found, the operation AGREE takes place under Minimal Search between the φ-features of T and those of the goal (i.e. *who* in (21)), the former getting valued by the latter as in (22):

(22)

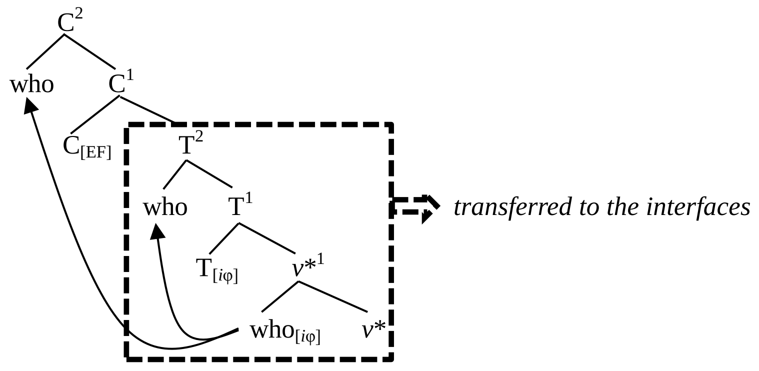


33. In other words, φ-features are no longer lexically inherent to T in his system but they are syntactically derivative. He argues the same for the relation between *v** and V, that is, φ-features originate from *v** and in the course of NS derivation, they are passed down to V by Feature-Inheritance.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

34. It is not clear in Chomsky (2007, 2008) exactly how Feature-Inheritance is accomplished. For example, what happens to φ-features on C after they are inherited by T? Are they deleted or copied, or do they just no longer exist on C? However, Chomsky (2013, fn. 47) explicitly claims that "inheritance has to be understood as copying", meaning that φ-features still remain on C. I will not adopt this approach but assume instead that φ-features no longer exist on C once they are inherited by T. For related issues, see Ouali (2008) among others.

After AGREE, T raises the goal to its Spec[35]. Independently, the EF of C also seeks a goal and if found, the goal is raised to Spec-C. Finally, the complement of the phase head C, namely, T^1, is sent to each of the interfaces by the operation called "Transfer" and the valued but C-I uninterpretable φ-features of T get removed before they reach the C-I interface, thereby avoiding C-I crash. All these operations are depicted in (23):

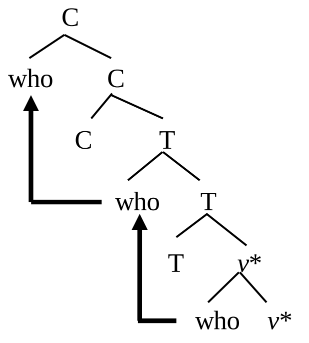
(23)



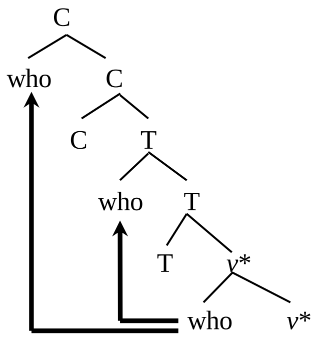
Chomsky (2008) deduces the rationale behind Feature-Inheritance from considerations of the C-I interface conditions, arguing that the C-I interface requires Narrow Syntax to *structurally* distinguish between A and A' positions and that Feature-Inheritance is the simplest mechanism that fulfills this C-I imposed requirement. He further maintains that Feature-Inheritance eliminates from the grammar non-uniform chains as seen in (24a):

35. Chomsky (2008) relates φ-feature inheritance to EPP. He thus states that "another question is whether inheritance is obligatory or optional. For C-T, that raises familiar questions about universality of EPP."

(24) a. Without Feature-Inheritance



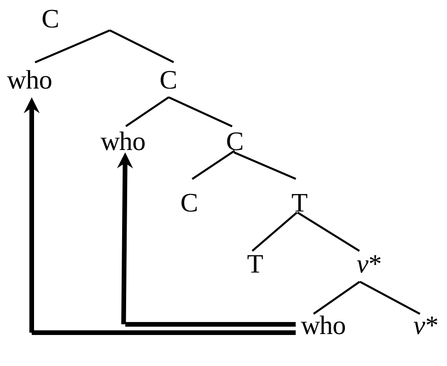
b. With Feature-Inheritance



In a framework without the Feature-Inheritance mechanism shown in (24a), the subject *who* in Spec-*v** moves to Spec-C via Spec-T, creating a non-uniform chain, (A', A, A). That is, the chain consists of two heterogeneous chains, namely, an A-chain (from Spec-*v** to Spec-T) and an A'-chain (from Spec-T to Spec-C). However, a system with Feature-Inheritance as in (24b) creates two independent movements for the subject *who*. In other words, the movement of *who* to Spec-T takes place independently of its movement to Spec-C, resulting in creating two unrelated but uniform chains, i.e. an A-chain and an A'-chain.

Richards (2007) argues against Chomsky (2008)'s claim that Feature-Inheritance is motivated by the C-I imposed structural distinction between A and A' positions on the grounds that the distinction can be established in a simpler way without resorting to Feature-Inheritance, as shown in (25):

(25)



If lexical items are allowed to have more than one specifier (i.e. multiple specifiers), C can raise *who* in Spec-*v** to its first Spec position and then to its outer Spec position as shown in (25). Therefore, the structural distinction between A and A’ positions can be established without the mechanism Feature-Inheritance.

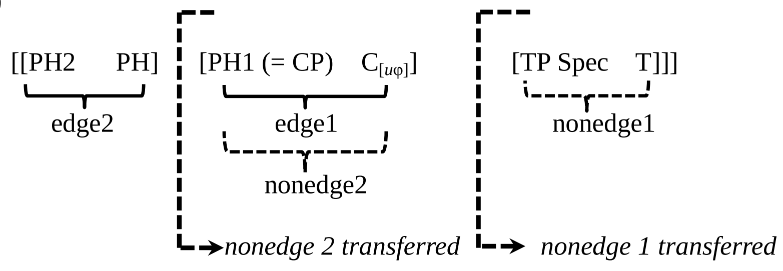
Richards (2007) attempts to find an alternative account of the motivation for Feature-Inheritance on the basis of the following two premises:

(26) **Premise 1:**
Valuation and Transfer of uninterpretable features must happen together.[36]
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 (PH1 below) have been completed, the complement of the phase head C (‘nonedge’ in Richards’ (2007) 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 (PH2 below).

(27)



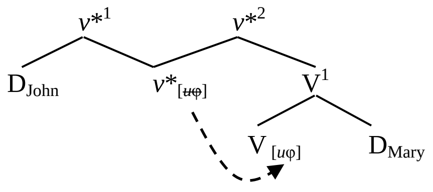
36. Premise 1 was originally pointed out by Epstein and Seely (2002), as Richards notes.

In a framework without C-to-T Feature-Inheritance, uninterpretable φ-features would get valued not on T but on C. However, as shown in (27), what is transferred at the point of this 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 violates Premise 1. He thus argues that 'feature-inheritance is the only device that can reconcile 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 Transfer in the v*P-domain*

Before discussing how Transfer and Feature-Inheritance can be incorporated into our framework and consequently how Merge is reinterpreted in our system, let's first consider our final (C-I offending) structure (18), repeated here as (28).

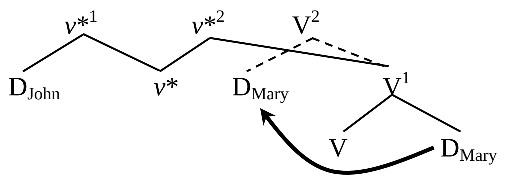
(28)



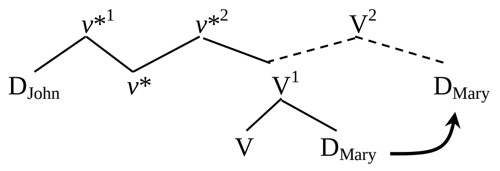
I follow Chomsky (2008) in assuming that uninterpretable φ-features on *v** are inherited by V and they induce an EPP effect. A question that immediately arises at this point is, Where does D_{Mary} move to?

Since the movement of D_{Mary} is triggered not by *v** but by V, D_{Mary} must be 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 (6) which requires Selector to c-command all the lexical items it selects for, i.e. I-Merge of D_{Mary} is exempt from Condition (6). Where does it then move to? Below are some of the conceivable landing sites for D_{Mary} (see (34) below for other possible derivations):

(29) a.

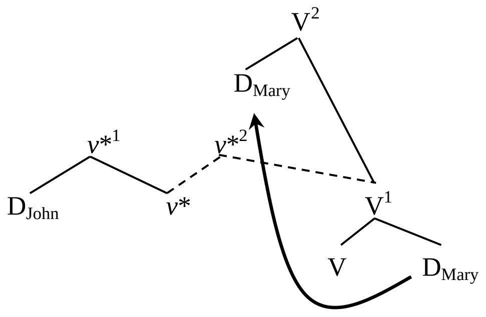


b.



(29a), where D_{Mary} moves to the Spec-V position, looks exactly the same as the structure EKS (2011) proposes and it creates even more peaks so that the resulting structure still cannot be interpreted at C-I interface (Condition (15)). The situation does not improve in (29b), where D_{Mary} moves rightward.[37] Below is the structure that I propose is created after the movement of D_{Mary}:

(30)



As discussed in Section 4, $v*$ in (30) is immediately dominated by two labels simultaneously, namely, $v*^1$ and $v*^2$. In set-theoretic notation, the status of $v*$ before the movement of D_{Mary} can be represented as below in (31a), where $v*$ occurs as a member of two sets simultaneously[38]:

37. In fact, (29a) and (29b) 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."

38. For expository purposes, labels are not indicated in (31).

(31) a. {D_{John}, *v**}, {*v**, {V, D_{Mary}}} Before movement of D_{Mary}
b. {D_{John}, *v**}, {D_{Mary}, ~~{*v**~~, {V, 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 (30) above is breaking the existing link between *v** and V^1. What this means in set-theoretic terms is that I-Merge of D_{Mary} eliminates one of the two occurrences of *v**, that is, *v** from the set {*v**, {V, D_{Mary}}} as shown in (31b) and consequently, the structure in (31c) is created after I-Merge of D_{Mary}. I take this eliminative operation by I-Merge whereby a member of a set gets eliminated to be a trigger for the operation Transfer. In other words, Transfer gets activated via I-Merge, transmitting the structure where I-Merge has taken place to the interfaces. This in turn implies that Transfer occurs only when eliminative I-Merge takes place, which I will further discuss in Section 5.3 below.

This type of link-breaking (or member-deleting) I-Merge may seem to violate the No Tampering Condition (NTC) proposed in Chomsky (2005, 2007, 2008) because it involves modifying the existing structure by breaking the link between *v** and V^1. If we consider the following claims in Chomsky (2008), however, this type of I-Merge is not unjustified:

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

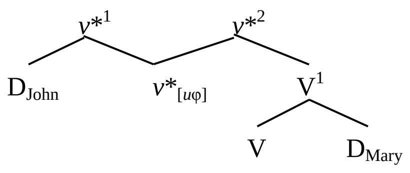
(33) **Strong Minimalist Thesis (SMT)**
Language is an optimal solution to interface conditions that FL [the Faculty of Language] must satisfy.

(34) 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 (15)) but we have just seen that the offending structure in (28) has no other alternatives to satisfy this interface-driven condition (hence conforming to SMT in (33)) than to remove either of the two peaks. Furthermore, I-Merge we propose is not the only operation that violates NTC defined in (32). 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, I-Merge with a built-in disconnecting ability can be justified (at least conceptually) even if it violates NTC.

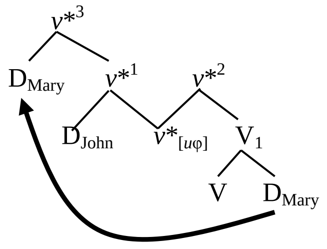
Now, let us examine implications of our version of I-Merge with a built-in eliminative operation as concerns the motivation for the mechanism Feature-Inheritance. Consider first the structure prior to the movement of $\mathrm{D}_{\mathrm{Mary}}$:

(35)

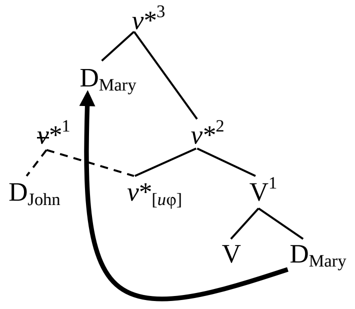


Without Feature-Inheritance, uninterpretable φ-features would stay on *v** and raise $\mathrm{D}_{\mathrm{Mary}}$ to its Spec. Because the movement of $\mathrm{D}_{\mathrm{Mary}}$ is triggered by unintepretable features on *v**, $\mathrm{D}_{\mathrm{Mary}}$ must merge with *v**. Below are three conceivable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

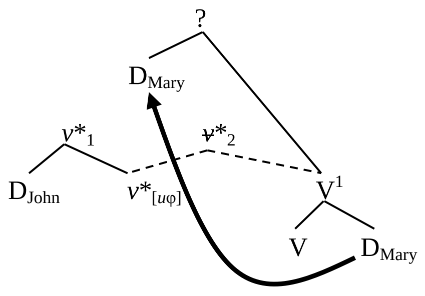
(36) a.


v^{*3}
$\mathrm{D}_{\mathrm{Mary}}$
v^{*1}
v^{*2}
$\mathrm{D}_{\mathrm{John}}$
$v^{*}_{[u\varphi]}$
V_1
V
$\mathrm{D}_{\mathrm{Mary}}$

b.

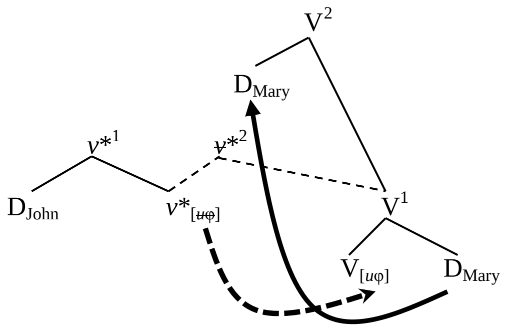

v^{*3}
$\mathrm{D}_{\mathrm{Mary}}$
$\cancel{v}^{*1}$
v^{*2}
$\mathrm{D}_{\mathrm{John}}$
$v^{*}_{[u\varphi]}$
V^1
V
$\mathrm{D}_{\mathrm{Mary}}$

c.


?
$\mathrm{D}_{\mathrm{Mary}}$
v^{*}_{1}
$\cancel{v}^{*}_{2}$
$\mathrm{D}_{\mathrm{John}}$
$v^{*}_{[u\varphi]}$
V^1
V
$\mathrm{D}_{\mathrm{Mary}}$

(36a) shows that $\mathrm{D}_{\mathrm{Mary}}$ moves to Spec-*v**. Notice, however, that I-Merge of $\mathrm{D}_{\mathrm{Mary}}$ does not involve any eliminative process and thus no structure can get transferred. Therefore, the derivation crashes due to a violation of Condition (15), i.e. it (still) does not have a single label that dominates all the lexical items. (36b), where D_{Mary} moves by breaking the link between D_{John} and *v**, is also ruled out as D_{John} cannot participate in further E-Merge and thus it will eventually reach the C-I interface without having its uninterpretable Case feature valued. (36c), where D_{Mary} moves by breaking the link between *v** and V^1 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 (36c). Furthermore, if we adopt the idea of what projects is always the Selector, it is not clear how the projection would work in (36c). If we assume φ-feature-inheritance by V, however, all the problems found in the three derivations above disappear:

(37)



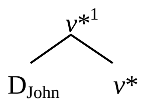
In (37), φ-features of *v** are inherited by V. What this means is that D_{Mary} must be connected with V since the requirement for D_{Mary} now resides in V. Once D_{Mary} moves to Spec-V via our eliminative I-Merge, V projects to become the label of the outcome as V is the Selector. In this application 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 in the structure (i.e. eliminative I-Merge) and thus violating NTC: language is indeed an optimal solution to interface conditions.

I conclude this section with a modified definition of Merge:

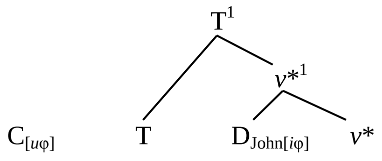
(38) Merge
Merge takes two syntactic objects (SOs), α and β, to form a set {α, β}. In doing so, Merge can modify an existing structure if the modification is required by interfaces.

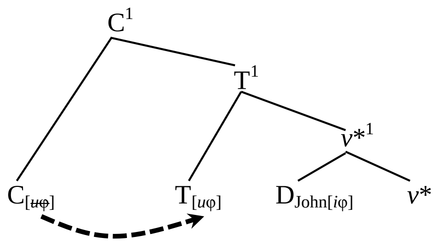
5.3 Feature-Inheritance and Transfer in the CP-domain

Let us now consider the implications of our analysis for the C-T domain. First, suppose that VP has been transferred to each of the interfaces, leaving the following structure in our workspace:

(39)


SELECT now chooses another phase head, C, from the lexicon[39] and puts it into the workspace. Then, C summons T to the workspace (the Summoning Condition (7i)). Although both C and T are introduced into the workspace, they cannot undergo E-Merge with each other before T E-Merges with $v*_1$ due to the Repulsion Effect in (13). (40a) and (40b) below show the structures before and after E-Merge of C with T^1, respectively.

(40) a.


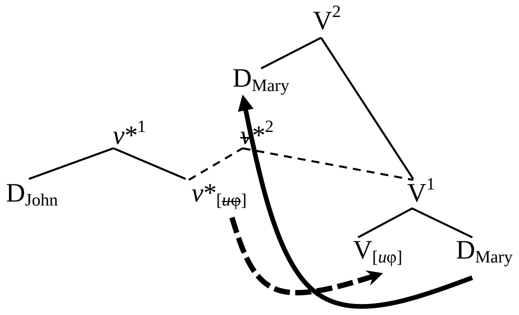
b.


Once C undergoes E-Merge with T^1, uninterpretable φ-features on C are inherited by T as shown in (40b). Subsequently, uninterpretable φ-features on T locate valued counterpart on D_{John} and AGREE takes place between the

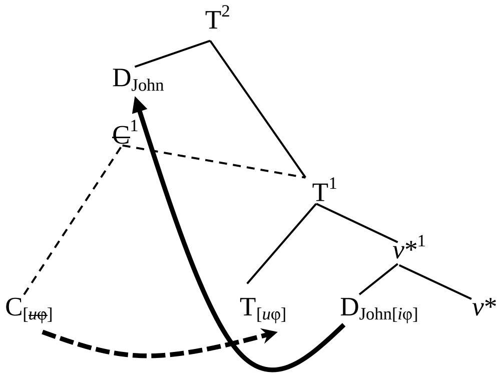
39. One may wonder what mechanism prevents choosing another *v** from the lexicon. In fact, we do not need such mechanism since the derivation with two *v**'s will not have a single dominator, so that it cannot be interpreted at the C-I interface (see Condition (15)). We do not need any order between the two phase heads, C and *v**, in terms of SELECT since in either case, we have the same two structures as in (40).

two. Finally, (φ-features on) T attacts D_{John}. The same question arises at this point that we asked in the previous section with regard to the landing site of D_{Mary}, that is, where does D_{John} move to? For comparison, the movement of D_{Mary} in (37) is repeated here as (41a) and the movement of D_{John} is shown in (41b):

(41) a. Movement of IA (= D_{Mary})

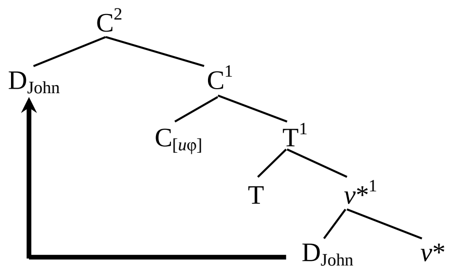


b. Movement of EA (= D_{John})



We argued in the previous section that D_{Mary} in (41a) moves to Spec-V by breaking the existing link between *v** and V^1. The same logic applies to the movement of D_{John} in (41b), where it moves to Spec-T by disconnecting the link between C and T^1. As a result, T^2 is transferred to each of the interfaces. Notice, however, that the structure in (41b) can be generated only if we assume the mechanism Feature-Inheritance. Consider the following structure to see what problems arise if we do not assume Feature-Inheritance:

(42)



D_{John} in (42) moves to Spec-C because the requirement for the movement resides in C, i.e. φ-features on C. Furthermore, I-Merge of D_{John} is not eliminative. Therefore, no structure can be transferred at this point and this invokes a problem pointed out by Richards (2007), i.e. uninterpretable φ-features on C are no longer distinguishable from their interpretable counterparts (on D_{John}). In addition, as Epstein (p.c.) points out, we cannot account for the occurrence of φ-features on T (i.e. *John love-**s** Mary*) if we do not assume Feature-Inheritance. If we assume Feature-Inheritance as in (41b) above, however, all these problems disappear.

6. Conclusion

I have proposed that along with other operations in NS, E-Merge can also be initiated only by phase heads and I have shown that this type of E-Merge (by phase) 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. E-Merge by phase vs. anomalous structure in the *v**P-domain created by phase-triggered E-Merge, I have proposed that I-Merge can eliminate a member from a set to satisfy interface conditions and that this eliminative I-Merge gets the operation Transfer activated. Finally, I have explored the implications of E-Merge by phase and eliminative I-Merge for Feature-Inheritance.

References

Chomsky, N. 1957. Syntactic Structures. The Hague: Mouton.

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. Ms.

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

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., 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.

Hornstein, N. 1999. Movement and Control. *Linguistic Inquiry*, 30: 69-96.

Ouali, H. 2008. On C-to-T Phi-feature Transfer: the Nature of Agreement and Anti-Agreement in Berber. In Roberta D'Alessandro, G. Hrafnbjargarson, and S. Fischer, eds., *Agreement Restrictions*. 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.