Merge or Move

Sau-In Yoon
(Kyungwon College)

Yoon, Sau-In. 1998. **Merge or Move.** *Linguistics*, 6-1, 329-352. The paper briefly reviews the structure-building mechanism and the structure-modifying mechanism in generative grammar. In the Minimalist Program developed by Chomsky (1993, 1994, 1995), the former is called Merge and the latter Move/Attract. The present research argues against the claim that Merge is always selected over Move (Chomsky 1995, Yang 1997, Yang, et al. 1998) when the two operations are equally applicable, and proposes the "local" principle called "as many features as possible" (AMFAP) for controlling the application of Merge and Move. I argue that AMFAP is superior to Collins' (1997) Chain-Formation Principle, and show that it provides a systematic explanation of English expletive and superraising constructions. **(Kyungwon College).**

1. The Computational System

Since Chomsky (1965), a generative grammar of language has been regarded as consisting of a lexicon and a computational system. The computational system comprises two separate components for building and modifying syntactic structures. One of the components, which is traditionally called a phrase structure (rule) system, is responsible for constructing phrase-markers, whereas the other, called a transformational (rule) system, is responsible for modifying phrase-markers constructed by the phrase structure component. This tradition has continued well into the Minimalist Program. In this paper, we will briefly review the development of the computational system, and critically analyze interactions between the two syntactic operations

Merge and *Move* in the Minimalist Syntax. In particular, we will argue against the claim that economy considerations force to choose Merge over Move, because the former is "cheaper" than the latter.

Since in a generative grammar the two syntactic operations perform different roles in generating well-formed syntactic structures for linguistic expressions, it has never been the case that we have to decide whether a phrase structure rule is selected over a transformational rule for a grammatical derivation or vice versa. Instead, it has always been the case that there is a clear boundary between the point where all structure-building rules cease to apply and the point where all structure-modifying rules begin to apply. It is especially clear in the *Aspects* (Chomsky 1965) model and the so-called Government-Binding (Chomsky 1981 and 1986) model of grammar that the phrase structure component first generates a D(eep)-structure for an linguistic expression, and then the transformational component modifies the structure to derive an S(urface)-structure.[1]

Chomsky's (1993) "A Minimalist Program for Linguistic Theory" (henceforth, MPLT) does not differ much from Chomsky's previous models of grammar in that a phrase marker is generated and transformed through a series of *generalized* and *singularly* transformations. The generalized transformation (henceforth, GT), which performs roughly the same function as traditional structure-building phrase structure rules, targets K, adds Φ, and substitutes Φ with K^1, forming K*, which must satisfy X-bar theory.[2] Besides the binary

1. I am not claiming that the two models of grammar have exactly the same phrase structure and transformational components (see Chomsky 1965, 1981, and 1986). In particular, the notion and role of S-structure in the GB-model are very different from those of surface structure in the *Aspects* model. Unlike surface structure, S-structure, which contains information of both deep and surface structures, acts as an input to both PF (i. e., phonetic form) for phonetic interpretation and LF (i. e., logical form) for semantic interpretation.

2. In addition to GT, MPLT assumes a set of so-called "projection rules" as in (i) to create a well-formed phrase marker that satisfies the X-bar theory:

operation GT, which maps a pair of syntactic objects (K, K^1) to K*, MPLT also assumes the singularly operation *Move-α*, which maps a single syntactic object K to K*. Move-α works just as GT does, except that α (i. e., K^1) that substitutes ϕ is a phrase marker within the targeted phrase marker K.[3]

However, MPLT differs from the previous models of generative grammar in several respects. Among others, it abandoned Emond's (1976) highly influential Structure Preserving Hypothesis (SPH) for Move-α.[4] The SPH restrains Move-α from building structures, but without the concept of D-structure, it is impossible to maintain the SPH in MPLT, because we can not formulate the structure which must be preserved throughout the derivation. In consequence, like GT, Move-α is also allowed to build or "extend" structures.[5] For the first time in the history of generative grammar, the structure building system (in this

---

(i) a. X
b. $[_{X'}\ X]$
c. $[_{XP}\ [_{X'}\ X]]$

The projection rules in (i) project a lexical item α with the category feature X selected from the lexicon to one of the forms in (ii):

(ii) a. X | α   b. X' | X | α   c. XP | X' | X | α

3. Perhaps, another difference between the binary operation and the singularly operation is that in the former the targeted phrase marker is always the root, while in the latter it may not be.

4. Chomsky (1993) argues that one of the reasons that we abandon the SPH for Move-α is the redundancy in that "the target of movement is somehow "there" before the operation takes place."

5. This is called the extension condition for Move-α and GT, which means that the two operations extend the targeted phrase K to K*, which includes K as its proper part.

case, GT) and the structure transforming system (in this case, Move-α) interact with each other to construct phrase markers.

Consider, for example, how MPLT derives the sentence in (1):

(1) It seems that John is certain to be here.

In the course of derivation, (1) will have (2) as its intermediate structure (cf. Chomsky 1993: 23):

(2) $[_{I'}$ seems $[_{I'}$ is certain [John to be here]]]

We derive sentence (1) by raising *John* to the embedded Spec and inserting the expletive *it* to the matrix Spec from the lexicon, as in (3):

(3) $[_{IP}$ *it* $[_{I'}$ seems $[_{IP}$ *John* $[_{I'}$ is certain [t to be here]]]]]

Notice that both embedded I′ and matrix I′ are extended to IP in (3), which means that both GT and Move-α *extend* phrase markers.

MPLT, however, does not make it clear whether or not *John* may be raised to the embedded Spec, forming (4), before we combine the verb *seems* and the embedded I′ through GT.

(4) $[_{IP}$ John $[_{I'}$ is certain [t to be here]]]

Then, GT targets (4), adds Φ, and substitutes Φ with *seems*, forming (5):

(5) $[_{I'}$ seems $[_{IP}$ John $[_{I'}$ is certain [t to be here]]]

Furthermore, it is not clear whether it is possible to insert *it* to the matrix Spec, forming (6), before we raise *John* to the embedded Spec:

(6) $[_{IP}$ it $[_{I'}$ seems $[_{I'}$ is certain [John to be here]]]]

In summary, it seems that MPLT allows several equally "expensive" convergent derivations for a single linguistic expression. If it is true, MPLT fails to eliminate the derivational redundancy from the grammar.

1.1. Merge

Just as Chomsky's previous models of linguistic theory, Chomsky's (1994) "Bare Phrase Structure" (henceforth, BPS) also assumes that a grammar of language consists of a lexicon and a computational system. The computational system, which includes Merge and Move, derives a pair of structural descriptions (π, λ) for each linguistic expression. A derivation *converges* if it yields a pair (π, λ) that receives an interpretation both at the articulatory-perceptual (A-P) interface and at the conceptual-intensional (C-I) interface; otherwise, it *crashes*. The pair (π, λ) receives an interpretation at the interface levels, if each of the pair consists of *legitimate* PF and LF objects, respectively. Whether or not the pair (π, λ) is legitimate representations of a linguistic expression is determined by the conditions imposed at the interface, which Chomsky (1994, 1995: 221) calls "bare output conditions". However, a pair (π, λ) yielded by a convergent derivation does not necessarily constitute a well-formed linguistic expression; it must meet an additional condition called the the *optimality* condition. The optimality condition specifies that among the competitive convergent derivations, the most optimal one is selected as an *admissible* derivation by economy principles of UG, and less economical ones are discarded even if they converge.

Since a comparison of derivational complexity can be made among convergent derivations that involve the same set of lexical items, the minimalist theory assumes that there is an array of lexical items for a particular linguistic expression, called *Numeration*. Thus, the derivations of sentences with the same meaning cannot be compared in their derivational complexities, if they contain different sets of lexical items:

(7) a. There is a man in the room
b. A man is in the room.

The two sentences, even though interpreted in the same way at the C-I interface, cannot be candidates for comparison of derivational complexities, because the former contains an expletive *there*, which is not in the latter. The computation of the pair (π, λ) begins after we formulate a numeration for a particular linguistic expression that we want to derive. A numeration contains a set of lexical items (LIs) with their indices. The operation Merge, then, forms a new syntactic object by selecting and combining (two) elements from the numeration and/or from syntactic objects already formed.

(8) *Merge*
Merge two syntactic objects α and β to form a new syntactic object K.

A derivation is completed only when no LI is left in the numeration, and it ends up with a single syntactic object. Chomsky (1995) represents the syntactic object formed by Merge in terms of the set notion. Thus, when two elements α and β are merged by Merge, the new syntactic object K is represented as {γ, {α, β}}, where γ is called the *label* of K, which is the head of either α or β (that is, either H(α) or H(β)). Thus, K = {H(α), {α, β}} can be represented as in (9).

(9)
```
      α
     / \
    α   β
```

The Minimalist Theory assumes that K consists of only features of LIs, which means that the computational system does nothing but rearrange features of LIs. Chomsky (1995: 225) calls this requirement a condition of *inclusiveness*. According to the inclusiveness condition, no bar levels of categories or indices for coreference can be added to syntactic objects. Thus, when Merge applies to two objects *the* and *man*, it produces a new syntactic object {the, {the, man}}, assuming that the determiner *the* is the head of the newly formed syntactic object:

```
(10)        the
           /   \
        the     man
```

Unlike a phrase-marker in any previous versions of generative grammar, (10) does not contain nodes specified with lexical or phrasal categories. But Chomsky (1995) argues that each lexical item in (10) is equipped with its unique categorial feature, together with other phonetic, semantic, and formal features. For example, we know that *the* and *man* are a D(eterminer) and a N(oun), respectively, because *the* is specified with categorial feature D and *man* with categorial feature N in the lexicon. From these categorial features, we can determine categories of syntactic objects. Chomsky (1995) calls a phrase-marker like (10) a *bare* phrase structure. Furthermore, the phrasal status of syntactic objects is determined by the following algorithm (Chomsky 1995: 242):

(11) a. a category that is not a project at all is a minimal projection
b. a category that does not project any further is a maximal projection
c. a category that is neither minimal nor maximal is an intermediate projection

According (11), in (10) the lower *the* is a minimal projection (D or D^0), and the upper *the* is a maximal projection (DP); *man* is both a minimal (N) and a maximal projection (NP). For the expositional purpose, however, the structure of (10) would be informally represented as in (12).

(12)
```
        DP
       /  \
    the    book
```

One of the most important characteristics of Merge[6] is that it combines only two (no more and no less) elements, consequently generating only a binary branching structure. Therefore, it never allows a structure like (13).

(13)
```
        K
      / | \
     α  γ  β
```

There are two leading arguments for why Merge combines only two syntactic objects. First, it is of conceptual necessity that at least two elements are required to form structure; of course, we can construct structure by combining three elements as in (13), but it is against the spirit of "minimalism": if we can build structure with less number of elements, why do we care to use more elements? Furthermore, non-branching structures like (14) are not allowed in the minimalist framework.

(14)
```
     D
     |
    the
```

---

6. Furthermore, Merge always applies at the root. For example, we cannot merge DP with TP within CP to construct (ii) from (i).

(i)
```
        CP
       /  \
      C    TP
          /  \
         T    vP
```

(ii)
```
        CP
       /  \
      C    TP
          /  \
        DP    T'
             /  \
            T    vP
```

Structures like (14) "redundantly" express one of the formal properties of the lexical item *the* (i. e., the categorial feature).

Another argument for binary branching is from Kayne's (1994) Linear Correspondence Axiom (LCA). According to Kayne's LCA, one of the conceptually necessary properties, the linearity of constituents of a linguistic expression, reflects their hierarchical structure. In other words, if α precedes β in a linguistic expression, α must asymmetrically c-commands β. But in a structure like (13), one of the elements cannot establish an asymmetric c-command relation with another element.[7]

In summary, the operation Merge generates a new syntactic object K = $\{\gamma, \{\alpha, \beta\}\}$ by combining two syntactic objects α and β, where γ is the label of K (which is head of K, H(K)), and α and β are lexical items or syntactic objects already formed.

1.2. Move

Another computational operation that plays a crucial role in the Minimalist Syntax is Move.

(15) *Move*

Suppose we have the category Σ with K and α. Move forms a new category Σ' by raising α to target K.

Unlike Merge, which always applies at the root (see fn. 6), Move applies either at the root in the case of maximal projection movement or at the non-root in the case of head and feature movement.

For illustration, let us reconsider the derivation of sentence (1), repeated here as (16):

(16) It seems that John is certain to be here.

---

7. See Kayne (1994) for the detailed discussion.

At some point in the derivation, we will have (17) as an intermediate structure of (16):

(17) $[_{I'}$ is certain $[_{IP}$ John to be here]]

Move applies to the matrix I′ in (17), converting it to (18):

(18) $[_{IP}$ John $[_{I'}$ is certain $[_{IP}$ t to be here]]]

Notice that BPS never allows a structure like (2), repeated here as (19):

(19) $[_{I'}$ seems $[_{I'}$ is certain [John to be here]]]

In (19), Merge has applied, merging the verb *seems* to the embedded I′, before we move *John* to the embedded Spec. (19) violates the following Strong Feature Condition (SFC):

(20) *Strong Feature Condition*
The derivation D is cancelled if α with a strong feature is in a category not headed by α.

To continue the derivation, the strong D feature in the embedded I (in this case, the verb *is*) must be checked off or erased by raising *John* to its Spec position. (19) violates (20), because the strong EPP feature of the embedded I remains in the category headed by the matrix I. Furthermore, it goes without saying that BPS does not allow structures like (6), repeated here as (21):

(21) $[_{IP}$ it $[_{I'}$ seems $[_{I'}$ is certain [John to be here]]]]

As in (19), the strong D feature of the embedded I remains in the matrix IP, violating the Strong Feature Condition.

Move may also apply at the non-root (i. e., K is contained in Σ in (15)). We find the instance of this kind of movement in head (X^0-) adjunction movement as in (22):

```
(22) a.        TP                         b.          TP
              /  \                                   /  \
            DP    T'                               DP    T'
                 /  \                                   /  \
                T    VP                           T0max     VP
                    /  \                          /  \     /  \
                   V    YP                       V    T   tv    YP
```

Chomsky (1995) assumes that X^0-adjunction of α to β (in this case, V-adjunction to T) always takes place within β^{0max} headed by β (in this case, T^{0max} headed by T).

2. Forming Phrase Markers

We have discussed two syntactic operations, Merge and Move. As we have seen, these two operations cooperate with each other in generating "legitimate" syntactic objects that can be interpreted at the interface levels. In the Minimalist Syntax, Merge is a conceptually necessary operation without which it is impossible to construct a linguistic expression. Since (virtually) all linguistic expressions consist of smaller expressions, language must have some form of mechanism for combining or merging smaller expressions to form larger ones. We call this mechanism Merge.[8]

8. As we have said, Merge can combine only two (no more and no less) elements. It seems that one of the elements acts as Selector, and the other as Selectee, as the verb *hit* may select the DP *the ball* to form the VP *hit the ball*. We call the Selector head of newly formed syntactic object. This means that Selector rather than Selectee always projects. Selectee becomes either a complement or specifier depending on the status of the Selector. If Selector is a minimal projection, the Selectee is a complement; otherwise, it is a specifier.

Movement of a syntactic object (i. e, maximal projection) takes place, always targeting the root and extending the targeted root, as Merge does. This is illustrated by wh-movement to [Spec, CP] in English, converting (23a) to (23b):

```
(23) a.        CP                    b.      CP
              /  \                          /  \
            C     TP                     wh-    C'
                 /  \                          /  \
                wh-                           C    TP
                                                  /  \
                                                  t_wh
```

Since, the operation Move is claimed to be "costly", it has to have some motivation to apply. Chomsky (1993, 1995) argues that the motivation is the morphological feature checking: Move applies to check features of the raised element and the target. This condition is called Last Resort:

(24) *Last Resort*

Move raises α to target K only if some feature F of α enters into a checking relation with some feature F′ of the target K.

Since F = F′, Move raises α to K only if some features of α and $H(K)^{0max}$ (including those features adjoined to H(K)) matches. If there is no such feature checking taking place between them, Last Resort prohibits Move from α raising to K.

As is well known, movement is also constrained according to how far it can move an element:

(25) *[$_{TP}$[$_{α}$ *John*] seems [that [$_{β}$ it] is certain [t_{John} [to be here]]]

(25) seems to satisfy the Last Resort Condition, both *John* and *it* entering into relevant checking relations with the matrix T and the

embedded T, respectively. The Minimal Link Condition (MLC) is suggested to prevent the application of Move in this way (Chomsky 1995: 311):

(26) *Minimal Link Condition*

H(K) attracts α only if there is no β, β closer[9] to H(K) than α, such that H(K) attracts β.

Let us consider how the MLC (26) prevents the derivation of (25). Before the DP *John* (= α) raises to the matrix Spec, *it* (= β) c-commands *John* in t_{John}, and *it* is not in the same minimal domain[10] of *John*. Hence, *it* is closer to the matrix Spec than *John*, preventing *John* from raising to the matrix Spec position.

3. Merge or Move

Chomsky's (1993, 1994, 1995) minimalist framework contains a number of "global" economy principles. To take a typical global economy principle, consider the Shortest Derivation Condition (SDR) in (27) (Kitahara 1997):

(27) *Shortest Derivation Condition*

Minimize the number of operations necessary for convergence.

The SDR in (27) is global in two respects. First, the grammar must count the number of operations applied in the set of alternative convergent derivations to choose the most optimal convergent derivation among them. Second, the grammar must look ahead whether application

9. The notion "closeness" is defined as follows:
β is closer to H(K) than α iff β c-commands α, and β is not in the same minimal domain of α.

10. See Chomsky (1993) for the definition of minimal domain.

of a certain operation will lead to a convergent derivation or not.[11]

For illustration, consider the examples in (28).

(28) a. There seems [$_{TP}$ *t* to be a man in the room]
b. *There seems [$_{TP}$ a man to be t in the room]

To derive (28a), *there* is merged in the Spec position of the embedded T to meet the EPP of the T. Then, the expletive *there* is raised to the Spec position of the matrix T to check off the strong D-feature of the matrix T. On the other hand, to derive (28b) *a man* is raised to the Spec of the embedded T to check the strong D-feature of the T; then, the expletive *there* is inserted to the Spec position of the matrix T to check off the D-feature of the matrix T.[12]

Chomsky (1995, 346) argues that Procrastinate forces us to choose the derivation in (28a) over (28b).

(29) *Procrastinate*
Reduce the number of overt movement operations unless required for convergence.

He argues that at some point of derivation, both derivations in (28) would have the structure in (30):

(30) [$_{T'}$ to be a man in the room]

There are two possible ways to check off the strong D-feature of the infinitival T: we can either merge the expletive *there* or raise *a man* as the specifier of the T. According to Chomsky (1995), Procrastinate forces us to choose the first option, yielding (31).[13]

11. We may say that any definition is global if it refers to "convergence".

12. The Case feature and Φ-features of the matrix T in both cases are checked by raising the relevant features of the associate nominal *a man* at LF.

13. Yang, et al. (1998: 232) also claims, ". . . if Merge and Move compete with

(31) [$_{TP}$ there [$_{T'}$ to be a man in the room]

Next, we merge the verb *seems* with (31), deriving (32).

(32) [$_{T'}$ seems [$_{TP}$ there to be a man in the room]]

Given the MLC in (26), only (28a) is derivable from (31), raising the expletive *there* to the specifier position of the matrix T in (32). At LF, the Case and Φ-features of the associate nominal *a man* are raised to the matrix T to check off the nominative Case assigning feature and Φ-features of the T.

Yang (1997), however, argues that the Case and Φ-features of the associate nominal must be coupled with *there* as soon as the expletive is introduced in the derivation.[14] He claims that we cannot account for the agreement of (33) in overt syntax, if we assume the LF movement of Case and Φ-features of the associate nominal.

(33) a. There is a man in the room.
b. *There are a man in the room.

Under the assumption of LF feature movement, the grammar has to look ahead at LF whether appropriate feature checking will take place there, so he argues that we have to assume overt movement of relevant features, if we want "local" explanation of the expletive constructions.

Except that feature checking must take place as soon as possible, as the Strong Feature Condition in (20), which claims that strong formal features must be checked as soon as they are introduced in a phrase marker, Yang's (1997) analysis of *there*-construction is similar to

---

each, the former always wins," [translated by SIY] meaning that only (31) is derivable from (30), given the numeration that contains the expletive *there*.

14. He argues that the formal features of the associate are overtly raised, following the principle which he calls the "As Soon As Possible" principle: check features as soon as possible, don't delay.

Chomsky's (1995) in that Merge is chosen over Move when we have an option to choose one over another. Thus, he argues that since the operation Merge is "costless" and overt feature movement is "less costly" than overt category movement, economy considerations force us to choose the derivation of (28a) rather than that of (28b)[15] as the optimal convergent derivation.

However, there is a classic example that contradicts the interpretation of Procrastinate that overt movement applies as late as possible, only allowing for Merge to apply first over Move. Consider the derivation of superraising construction in (34).

(34) *[$_{TP}$ *John* seems [that it is certain [t_{John} [to be here]]]

At some point of the derivation of (34), we will have the structure in (35) as an intermediate structure:

(35) [$_{T'}$ is certain [John [to be here]]]

As in (30), we have two options: we either insert the expletive *it* or move *John* to the specifier position of the embedded T (i. e., the verb *is*). Following Chomsky (1995: 346) and Yang (1997), if we choose the first option, we will get (36):

---

15. Interestingly, no one has yet presented serious analysis of the fact that (28b) is ungrammatical in English. Chomsky (1995: 344) only says, "Principles of UG might bar (169) [= (28b)] generally, . . ." I think he claims that some principles of UG that account for multiple-subject constructions (MSCs) in languages like Icelandic, which allows the structure like (28b), may block (28b) in English, which does not generally allow MSCs. Notice that (28a) and (28b) have the same degree of derivational complexity: both examples involve one Merge and one Move. Furthermore, in (28b) we can legitimately raise the relevant features of *a man* to check Case and ϕ-features of the matrix T from the position in which the associate nominal occurs.

(36) [$_{TP}$ it is certain [John [to be here]]]

Suppose that we merge the verb *seems* with (36). We get (37):

(37) [$_{T'}$ seems [$_{TP}$ it is certain [John [to be here]]]

(37) cannot be an intermediate structure from which we can get any convergent derivation. If the expletive *it* is raised to the matrix Spec as in (38), the Case feature of the matrix T will remain unchecked, causing the derivation to crash.

(38) *[$_{TP}$ *It* seems [that t_{it} is certain [John [to be here]]]

The raising of *John*, however, will lead to the derivation of (34), which violates the MLC. The only way to save (35) is, then, to choose the second option: instead of merging the expletive *it*, overtly raising *John* to the specifier position of the embedded T. This means that Move has to be chosen over Merge, contrary to the claims of Chomsky (1995), Yang (1997), and Yang et al. (1998) that Merge is always chosen over Move, because the former is more economical than the latter.

Chomsky (1995: 295-97) offers a "global" analysis of the superraising construction. He claims that what bars superraising as in (34) in favor of (39) is not economy considerations, because economy of derivation is taken into consideration at some stage Σ of the derivation only if there is a convergent extension of Σ.

(39) It seems that John is certain [t to be here].

Since there are no convergent extensions of (36),[16] that is, (36) cannot be a stage on the way to a convergent derivation at all, we cannot say that (36) is rejected because of economy considerations. Notice that the

16. Only ungrammatical (34) and (38) are derivable from (36).

optimal derivation is selected only among convergent derivations. The conclusion we can draw from Chomsky's (1995) analysis of superraising constructions is whether we apply Merge or Move is determined not on the basis of which operation is more economical, but which operation will eventually lead to a convergent derivation.

If we strictly follow this line of reasoning, it seems that the grammar does not need any principles or conditions like Last Resort, SFR, SDC, MLC Procrastinate, etc. but the following single global condition suffices to constrain all applications of computational operations, as Lee (1997) argues:

(40) Apply an operation OP only if required for convergence.

For the expletive sentence like (28a), for example, Merge is chosen over Move to obtain (31) from (30), because only (31) yields the convergent extension, given the numeration containing the expletive *there*. On the other hand, to block the superraising construction like (34) and to derive the well-formed expression like (39), Move is chosen over Merge, because (36) in which Merge has applied instead of Move does not have a convergent extension, as Chomsky (1995) claims. In sum, we can say that convergence is the only criterion for whether we choose to apply Merge or Move to a certain structure.

3.1. Collins' (1997) Chain-formation Principle

Collins' (1997) "local" Minimality Condition does not seem to help account for the problems related to English expletive and superraising constructions:

(41) *Minimality*

An operation OP (satisfying Last Resort) may apply only if there is no smaller operation OP' (satisfying Last Resort).

As we have discussed, at some point of derivation, (28a) and (39) will have (30) and (35), repeated here as (42a) and (42b), respectively, as their respective intermediate structures:

(42) a. [$_{T'}$ to be a man in the room]
b. [$_{T'}$ is certain [John [to be here]]]

Suppose we assume Merge is an operation "smaller" than Move, as in Chomsky (1993, 1994, 1995). The Minimality Condition in (41) forces us to merge the expletives *there* and *it* as Specs of TP in (42a) and (42b), deriving (31) and (36), respectivley, repeated here as (43):

(43) a. [$_{TP}$ there [to be a man in the room]]
b. [$_{TP}$ it is certain [John [to be here]]]

We can get (44) from (43) by merging the verb *seems*:

(44) a. [$_{TP}$ seems [$_{TP}$ there [to be a man in the room]]]
b. [$_{TP}$ seems [$_{TP}$ it is certain {John [to be here]]]]

As we have seen, for the former there is a convergent derivation of the grammatical sentence (39) by raising the expletive *there* from embedded Spec position to the matrix Spec position in (44a). However, there is no convergent derivation for the latter: if *John* is raised to the matrix Spec as in (34), the MLC is violated, whereas the Case feature of the matrix T will remain unchecked if the expletive *it* is raised to the matrix Spec as in (38).

(34) **John* seems [that it is certain [t_{John} [to be here]]]
(38) *It seems [that [$_{TP}$ *t* is certain [John [to be here]]]]

Therefore, Collins (1997: 122) assumes that, "If merging a constituent and moving a constituent both satisfy Last Resort and Minimality, they are equally costly." He further assumes that there is no such thing as

Numeration in grammar. Thus, Merge copies a lexical item in the lexicon and merges it with another syntactic object, whereas Move copies an element in a phrase marker and merges it with the phrase marker that contains it. Since there is no Numeration, the sentences in (7) are regarded as optional variants derivable from a structure like (45):

(7) a. There is a man in the room
b. A man is in the room.
(45) [$_{T'}$ is a man in the room]

Collins (1997) argues that either inserting *there* or raising *a man* to the Spec is possible in (45), because both satisfy Last Resort and Minimality.

Now, let's return to the set of expletive constructions in (46):

(46) a. There seems to be a man in the room.
b. *There seems a man to be in the room.
c. A man seems to be in the room.

As we have discussed, at some point in the derivation of the sentences in (46), we will have the structure in (47):

(47) [$_{T'}$ to be a man in the room]

As you have already noticed, we can either insert *there* or move *a man* to Spec of the infinitive, deriving (48):

(48) a. [$_{TP}$ there [to be a man in the room]]
b. [$_{TP}$ a man [to be t in the room]]

Merging (48) with the verb *seems* will produce (49):

(49) a. [$_{T'}$ seems [$_{TP}$ there [to be a man in the room]]]

b. [$_{T'}$ seems [$_{TP}$ a man [to be t in the room]]

We know that we have no choice but to apply Move to both cases of (49) to obtain the grammatical sentences (46a) and (46c). Following Collins' (1997) assumptions, however, we can also insert the expletive *there* in both structures in (49), deriving (50):

(50) a. [$_{TP}$ there seems [there [to be a man in the room]]][17]
b. [$_{TP}$ there seems [a man [to be t in the room]]

In (50), the expletive *there* satisfies both Last Resort and Minimality, checking the EPP feature of the matrix T in both cases.

Collins (1997) proposes the following Chain-Formation Principle to account for the impossibility of (50b) (=(46b)):

(51) Chain-Formation Principle

If there are two OP_1 and OP_2 applicable to a set of representations Σ (both satisfying Last Resort and Minimality), then choose the operation that extends an incomplete chain.

An incomplete chain for the present case is defined as follows:

(52) Incomplete Chain

Let Ch be a (nontrivial) chain of the form (α, . . . , t), where α has D feature that has entered into a checking relation with a[n] EPP feature and has an unchecked Case feature. Then Ch is an incomplete chain.

Collins (1997: 123-24) argues that the chain (*a man, t*) is an incomplete chain, because *a man* that has entered into a checking relation with an

17. We could rule out (50a) by assuming that every expletive must has an associate nominal; there are two expletives but only one nominal in (50a).

EPP feature of the embedded T has an unchecked Case feature. Therefore, *a man* has to be raised in (49a) to extend the incomplete chain.

3.2. As Many Features As Possible Principle

Then, let's consider whether Collins' (1997) account of expletive constructions can be extended to the superraising construction in (53):

(53) *[$_{TP}$ *John* seems [that it was told t_{John} [that he had been admitted]]]

At some stage of the derivation, (53) will have (54) as its intermediate structure:

(54) [$_{T'}$ was told John [that he had been admitted]]

It is clear that his Chain-Formation Principle in (51) does not require *John* to be raised to Spec as in (55a), instead of inserting the expletive *it* as in (55b):

(55) a. [$_{TP}$ John was told t [that he had been admitted]]
b. [$_{TP}$ it was told John [that he had been admitted]]

Notice that in (54) neither does *John* form a nontrivial chain nor enter into a checking relation with an EPP feature.

It seems that there are two possible ways to overcome these problems. First, we assume that Merge and Move are equally costly as long as they satisfy Last Resort and Minimality, following Collins (1997). It will allow the grammar to generate both structures in (55), but (55b) will be ruled out by the MLC if it is extended to the superraising construction in (53). But whether we accept this option depends on whether we allow the grammar to generate an intermediate structure for which there is no convergent derivation. Second, we may assume an additional principle like (56) to bar the derivation of structures like (55b):

(56) As Many Features As Possible (AMFAP)
If there are two OP_1 and OP_2 applicable to a set of representations Σ (both satisfying Last Resort and Minimality), then choose the operation that checks off as many features in Σ as possible.

As we have discussed, there are two operations applicable to (54): either inserting *it* as in (55b) or raising *John* as in (55a). By inserting *it* we can check off the Case feature of the T, but *John* still contains an unchecked Case feature. On the other hand, by raising *John* we can check off the Case features of both the T and *John*. Consequently, the grammar allows only (55a) to be derived, but not (55b).

The AMFAP Principle in (56) also accounts for expletive constructions in (46). According to (56), in (47) we can either insert the expletive *there* as in (48a) or raise *a man* as in (48b), because both operations check only the EPP feature of the embedded T. But in (49) only the operation Move is applicable, raising *there* and *a man* to the matrix Spec. Suppose we apply Merge, inserting the expletive *there* as in (50). (50a) is ruled out by the fact that one of the two expletives does not have its associate nominal. (50b) violates AMFAP: that is, only the EPP feature of the matrix T is checked by inserting *there*, but both the EPP feature and Case feature are checked by raising *a man*.

In conclusion, I have argued that any economy conditions that refer to the "global" notion of convergence are not appropriate, so that they must be eliminated from grammar. I have also argued, based on English superraising constructions, that it is not always the case that the "costless" operation Merge has priority over the "costly" operation Move. I have shown that, although Collins' (1997) Chain-Formation Principle may account for expletive constructions, it is not appropriate for superraising constructions. I propose a new principle called "as many features as possible" (AMFAP) to account for both expletive and superraising constructions appropriately.

References

Chomsky, N. (1965) Aspects of the Theory of Syntax, Cambridge, Mass.: MIT Press.

Chomsky, N. (1981) *Lectures on Government and Binding,* Dordrecht: Foris.

Chomsky, N. (1986) *Barriers*, Cambridge, Mass.: MIT Press.

Chomsky, N. (1993) "A minimalist program for linguistic theory", K. Hale and S. J. Keyser, eds., *The View from Building 20,* Cambridge, Mass.: MIT Press.

Chomsky, N (1994) "Bare phrase structure", *MIT Working Papers in Linguistics,* April, 1994. Also published in G. Webelhuth, ed., Government and Binding Theory and the Minimalist Program. Oxford: Blackwell 1995.

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

Collins, C. (1997) *Local Economy*, Cambridge, Mass.: MIT Press.

Emonds, J. (1976) *A Transformational Approach to Syntax,* New York: Academy Press.

Kayne, R. (1994) *The Antisymmetry of Syntax*, Cambridge, Mass.: MIT Press.

Kitahara, H. (1997) *Elementary Operations and Optimal Derivations,* Cambridge, Mass.: MIT Press.

Lee, H. B. (1997). "Economy Conditions in the Minimalist Theory: Local or Global," *Linguistics,* 5. 1., 265-87.

Rizzi, L. (1990) *Relativized Minimality,* Cambridge, Mass.: MIT Press.

Yang, Charles D. (1997). "Minimal Computation," Ms., MIT.

Yang, D. W., et al. (1998). *The Minimalist Theory: LSK Linguistics Lecture Series 4.* Seoul: Hashin Publishing Co.

Department of General Education
Kyungwon College
Sungnam City, Kyunghi Province