Economy Conditions in the Minimalist Theory: Local or Global

Hong Bae Lee
(Sogang University)

Lee, Hong-Bae. 1997. **Economy Conditions in the Minimalist Theory: Local or Global.** *Linguistics*, 5-1, 255-279. The paper discusses the problems related to the "global" nature of economy conditions in the Minimalist Program developed by Chomsky (1993 & 1995), and the advantages of reformulating "global" economy conditions into "local" ones. I suggest the Maximality Condition that requires that the operation with maximal effects be chosen to apply, to supplement Collins' (1997) Minimality Condition. **(Seokang University)**

1. A General Outline of the Minimalist Theory

A grammar of a particular language L is a steady state L_s of UG, developed from the initial state L_i of UG by specifying the values of parameters through linguistic experiences. Thus, L is an instantiation of the language faculty with parametric options specified. One component of the human language faculty includes a generative procedure that produces pairs of representations (π, λ) for linguistic expressions. π is a PF representation that is interpreted at the articulatory-perceptual (A-P) interface, and λ is an LF representation that is interpreted at the conceptual-intentional (C-I) interface. In that sense, the pair (π, λ) is regarded as "instructions" to the performance systems, telling how meaning and sound of the linguistic expression under question are to be interpreted.

A grammar consists of a lexicon and a computational system C_{HL}. The lexicon contains a set of lexical items (LIs) specified with their idiosyncratic features. The computational system provides a set of structural descriptions (SDs) for each linguistic expression in accordance with the computational principles of UG (e.g., Select, Merge, and Move) and the economy principles of UG (e.g., Last Resort, Shortest Derivation Condition, etc.). The Minimalist Theory assumes that "the language faculty is nonredundant, in that particular phenomena are not "overdetermined by principles of language" (Chomsky 1993, 2), from which the name "minimalism" came. Therefore, the descriptions of linguistic phenomena include only "conceptually necessary" elements, and no more. Given this, the concepts of D- and S-structures have no place in the Minimalist Theory, since they are not conceptually necessary levels of the grammar, but levels assumed grammar-internally.

A derivation converges if it yields a pair (π, λ) that receives an interpretation at each relevant interface level; otherwise, it crashes. A pair (π, λ) receives an interpretation at the interface levels, if each of the pair consists of "legitimate" PF and LF objects, respectively. Whether 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 whether it is derived in an optimal way. Among the competitive derivations, the most optimal one is determined by economy principles of UG, and less economical derivations are discarded even if they converge.

2. The Computational Systems

2.1 Merge

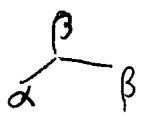
As we have said, a grammar consists of a lexicon and the computational system. The operation Select selects lexical items (LIs) from the lexicon to form a Numeration for a particular linguistic expression. A Numeration contains a set of LIs with their indices, from which we form a particular linguistic expression under the principle of derivational optimality. The operation Merge, then, form a new syntactic object by combining (two) elements from the Numeration or from syntactic objects already formed.

(1) *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. Chomsky (1994, 1995) expresses 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 formed by the operations is expressed as {γ, {α, β}}. Here, 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}} as in (2), or K = {H(β), {α, β}}.

(2)
```
      α
     / \
    α   β
```



K consists of only features of LIs, meaning that C_{HL} does nothing but arrange features of LIs. Chomsky (1995, 225) calls it a condition of *inclusiveness*. According to the inclusiveness condition, no bar levels of categories or indices for coreference can be introduced into syntactic objects. Thus, when Merge applies to two objects *the* and *book*, it produces a new object {the, {the, book}}, assuming that the determiner *the* is the head of the newly formed syntactic object. The result can be represented as in (3).

(3)
```
       the
      /   \
    the    book
```

Compare (2) with the awkward phrase marker (4) assumed in the early minimalist theory (see Chomsky 1993).

(4)
```
        DP
        |
        D'
       /  \
      D    NP
      |    |
     the   N'
           |
           N
           |
          book
```

Returning to (3), let us consider how categories and phrasal levels of constituents are determined. As we will see section 3.1, every lexical item is specified with its unique categorial feature in the lexicon. From these categorial features, we can determine categories of syntactic objects. For example, we know that *the* and *book* are a D(eterminer) and a N(oun), respectively, because *the* is specified with categorial feature D and *book* with categorial feature N in the lexicon. Thus, phrase marker (4) redundantly specifies the fact that *the* and *book* are a determiner and a noun, respectively. Furthermore, structures like (4) stipulate phrasal levels of constituents of a linguistic expression by employing the bar or prime notion. In the bare phrase structure theory, however, the phrasal status of constituents is determiend by the following algorithm (Chomsky 1995, 242):

(5) a. a category that does not 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.

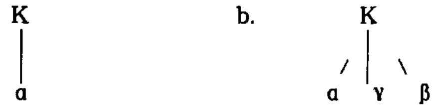
According (5), the lower *the* is minimal (D or D^0), and the upper *the* is maximal (DP); the constituent *book* is both minimal (N) and maximal (NP). Thus, the "informal" phrase marker for (3) would be as in (6).

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

Merge can generate two different types of categories: a one-segment category and a two-segment category. To distinguish them, different labels are used: a one-segment category projected from K by merging α and K is represented as {H(K), {α, K}}, and a two-segment category as {<H(K), H(K)>, {α, K}}.

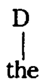
Another characteristic of Merge is that it combines only two (no more and no less) elements, generating only a binary branching structure. Therefore, there will be no non-branching structure and no structure like (7).

(7) a. b.



There are two leading arguments for why Merge allows only binary branching. 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 (7), but it is against the spirit of "minimalism." That is, if we can accomplish something with less number of elements, why do we care to use more elements? Furthermore, nonbranching structures like (8) are not allowed in the minimalist framework.

(8)



Structures like (8) "redundantly" express one of the formal properties of the lexical item *the* (i.e., the categorial feature).

Another argument for binary branching is based on Kayne's (1994) Linear Correspondence Axiom (LCA) (see Collins (1997). According to Kayne's LCA, linearity of constituents in a linguistic expression reflects their hierarchical structure. In other words, if α precedes β, α must asymmetrically c-commands β. But in a structure like (7), one of the elements cannot establish an asymmetric c-command relation with another element (See Kayne (1994) for the discussion).

As we have noticed, when α and β are merged to form K, K must be the projection of either α or β. The question is why not γ, distinct from both α and β. The answer should be sought from the fact that every linguistic expression has an "endocentric" structure. In other words, every expression in a natural language consists of a head with or without surrounding elements such as specifier, complement, modifier, etc.

Furthermore, Merge always applies at the root only. For example, we cannot merge DP with TP within CP to construct (9b) from (9a).

(9) a.

```
          CP
         /  \
        C    TP
            /  \
           T    vP
```

b.

```
          CP
         /  \
        C    TP
            /  \
          DP    T'
               /  \
              T    vP
```

In short, the operation Merge always 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 either lexical items or syntactic objects already formed. In the structure $\{\gamma, \{\alpha, \beta\}\}$ formed by Merge, the elements that function in the computation are called terms of K Chomsky (1995, 247).

(10) a. K is a term of K.
b. If L is a term of K, then the members of the members of L are terms of K.

Thus, according to (10) the terms of {the, {the, book}} are {the, {the, book}}, *the* and *book*, that latter two of which are the members of the members {the, book} of {the, {the, book}}.

2.2 Move

The second computational operation that plays a crucial role in this grammar is Move. Unlike Merge, which always applies at the root, Move applies either at the root or at the nonroot.

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

If Move applies at the root (i.e., K = Σ), Move functions like Merge, forming Σ′ by combining α and Σ (= K). This is the case of overt XP movement like *wh*-movement to [Spec, CP] in English as in (12) and universal EPP movement as in (13).

```
(12) a.        CP                      b.        CP
              /  \                              /  \
             C    TP                          wh-   C'
                 /  \                              /  \
                wh-                               C    TP
                                                      /  \
                                                      t_wh
(13) a.        TP                      b.        TP
              /  \                              /  \
             T    υP                          DP    T'
                 /  \                              /  \
               DP    υ'                           T    υP
                                                      /  \
                                                   t_DP    υ'
```

If Move applies at the nonroot (i.e., K is contained in Σ), it replaces K with L = {γ, {α, K}}. We find the instance of this kind of movement in head (X^0-) adjunction.[1]

```
(14) a.        υP                      b.        υP
              /  \                              /  \
            DP    υ'                          DP    υ'
                 /  \                              /  \
                υ    VP                      υ^0max   VP
                    /  \                      /  \   /  \
                   V    YP                   V   υ  t_V   YP
```

1. υ in (13) stands for a light verb. Chomsky (1995) assumes that a sentence with a transitive verb has the argument structure:

```
        υP
       /  \
   Subj    υ'
          /  \
         υ    VP
             /  \
            V    Obj
```

Chomsky (1995) assumes that X^0-adjunction of α to β (in this case, V-adjunction to υ) always takes place within β^{0max} headed by β (in this case, υ^{0max} headed by υ).[2]

2.3 Projection of Target

You have indicated that phrase markers generated under the Bare Phrase Structure Theory differ in a number of respects from those assumed in the early minimalist theory: no bar level, no indices, no categorial projections etc. Another important difference between the early generative grammars and the Bare Phrase Structure Theory can be found in the nature of phrase markers to which the operation Move applies. Ever since Emonds (1976), it has been one of the inviolable principles in the generative grammar that transformations cannot build structures, which is called the Structure Preserving Hypothesis. In other words, until Chomsky (1993) a phrase marker to which the operation Move is to apply must provide the "landing site" for the moved constituent, so that Move does not build any structure. As we have seen in the case of overt XP movement exemplified in (12) and (13), overt substitution movement always builds a new structure.[3] In (12a) Move picks out a *wh*-phrase within the target phrase CP (in Chomsky's (1981) GB framework and Chomsky (1993), it is C'), and merges it with CP, projecting the target category CP, the result of which is (12b). The same type of computation also takes place in (13): Move copies the subject DP and merges it with TP (in the GB framework and the early minimalist framework, it is T'), projecting the target. The result of the computation is (13b).

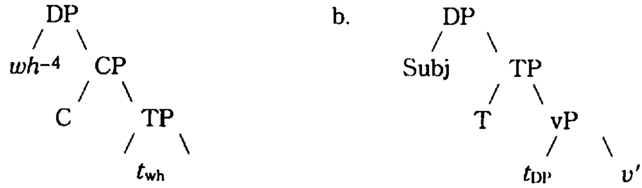
We may ask what happens if, rather than the target, the phrase that is raised projects in (12) and (13). It is obvious that it yields undesirable phrase markers:

---

2. In (14b), L (= v^{0max}) = {v, {V, v}} has replaced K (= v) in (14a).

3. We use the term "substitution" for the movement that creates a one-segment category to distinguish it from the adjunction movement that creates a two-segment category; even in the present framework, adjunction is taken to be structure-preserving.

(15) a. b.



In (15) because the moved *wh*-phrase and the subject DP are internally complex (that is, they are not a minimal X^0 category), they both become the intermediate projection D′ according to the algorithm in (5). This means that in (15) CP and TP are interpreted as specifiers of D′, which are certainly unwanted options. Thus, we can conclude that at least in XP movement the target must always project.

Suppose that an "internally simplex" category (for example, *who* and *it*[5] moves and merges with CP and TP, projecting themselves. Then, they become the head of the new projection, with CP and TP as their complements, respectively, which is also an unwanted option.[6] Therefore, we may say that for the case of (substitution) movement the target must always project.

2.3 Computational Economy Principles

We have discussed two operations, Merge and Move[7]. According to the minimalism assumptions, a "perfect" language would be one that is derived without the application of any computation but Merge, which is assumed to be "costless". In other words, the language derived by applying only the operation Merge is the "best" language from the economic point of view. Then, why does the computational system C_{HL} in human language employ the additional operation Move? The answer to the question can be found from the fact that linguistic items commonly appear "displaced" from the position at which they receive

4. We assume that a *wh*-phrase is also a DP.

5. In fact, Chomsky (1994, 1995) assumes that they are not internally simplex minimal categories.

6. As we will see shortly, movement takes place to check morphological features, but no checking is possible in head-complement relations.

7. Adjunction operation will not be discussed here; see Chomsky (1994, 1995).

interpretation at LF. We do not know much about why human language displaces elements from their base-generated positions.[8] But we do know something about the displacement mechanisms in human language, called Move. In this section, we will discuss some economy conditions that constrain the movement operation.

Since, unlike Merge,[9] the operation Move is 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:

(16) *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:

(17) $*[_{TP}[_{\alpha}$ *John*] seems [that $[_{\beta}$ it] was told t_{John} [that he passed the exam]]]

(17) seems to satisfy the Last Resort Condition, *John* entering into a checking relation with the matrix T. The Minimal Link Condition is suggested to prevent this application (Chomsky 1995, 311):

(18) *Minimal Link Condition*

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

---

8. Chomsky (1995, 317) briefly mentions that displacement is due to "... facilitation of parsing on certain assumptions, the separation of theme-rheme structures from base-determined semantic (θ) relations, and so on."

9. Collins (1997) claims that Merge is also a costly operation.

To understand the notion "closeness", we have to know the notion "equidistance" first, as formulated in Chomsky (1993):

(19) *Equidistance*

If α, β are in the same minimal domain,[10] they are equidistant from γ.

Consider the following abstract structure:

(20)
```
              XP
             /  \
        Spec2    X'
                /  \
           [Y-X]    YP
                   /  \
              Spec1    Y'
                      /  \
                    tY    ZP
```

$Spec_1$ and $Spec_2$ are both in the minimal domain of the chain CH = (Y, *t*). and are equidistant from α = ZP or within ZP. Move can therefore raise α to target either $Spec_1$ or $Spec_2$. If we reformulate the notion of equidistance in terms of Attract, we get (21):

(21) *Equidistance*

β does not prevent H(K) from attracting α if β is in the same minimal domain of α.

In (20) $Spec_1$, being in the same minimal domain as $Spec_2$, does not prevent the category X' (= {X, {X, YP}}) from attracting α = ZP or within ZP to $Spec_2$.

The notion "closeness" is now defined as in (22).

(22) *Closeness*

β is closer to H(K) than α iff β c-commands α, and β is not in the same minimal domain of α.

Given (22), let us consider how the MLC (18) prevents the derivation of

10. We will discuss the notion of minimal domain in the following section.

(17). Before the DP *John* (= α) raises to the matrix Spec, *it* (= β) c-commands *John*, and *it* is not in the same minimal domain of *John*. Hence, *it* is closer to the matrix Spec than *John*, preventing *John* from raising to the matrix Spec position.

When Pollock (1989) formulates the Split INFL Hypothsis, he argues that in French-type languages a verb (overtly) moves to I, while in English-type languages it does not. Chomsky (1993) explains this fact in terms of the notion of Procrastinate:

(23) *Procrastinate*

Reduce the number of overt operations unless required for convergence.

In English, overt verb movement to I is not required for convergence; therefore, Procrastinate forces verb movement to be covert.[11] On the other hand, in French Procrastinate allows overt verb movement, because it is required for convergence.

Another economy condition for derivation is the Shortest Derivation Condition:

(24) *Shortest Derivation Condition*

Minimize the number of operations necessary for convergence.

(24) is required to choose an optimal derivation among convergent derivations.

3 Checking Features

3.1 Types of Formal Features

Chomsky (1995) classifies formal features[12] into Interpretable and Uninterpretable features. The [+Interpretable] features are those that are required for semantic interpretation, and include categorial features and ф-features of nominals, while [-Interpretable] features are those that are

11. In the minimalist framework, covert operation is less costly that overt one.

12. A lexical item is equipped with phonological features, semantic features and formal features; only the last ones are relevant to syntax.

not required for semantic interpretation, hence being deleted, and include Case features of nominals, Case-assigning features of T and V, agreement features of verb and adjective, strength of a feature, and affixal features.

(25) a. Interpretable Features
  (i) categorial features
  (ii) Φ-features of nominals
 b. Uninterpretable Features
  (i) Case features of nominals, verbs and Tense
  (ii) Φ-features of verbs, adjectives
  (iii) strength feature and affixal features

Now, consider the following structure in which feature F is attracted by a sublabel F′ of $H(K)^{0max}$:

(26)
```
          K
         / \
        F   K
           / \
     H(K)^0max  . . .
```
(the sublabel F′ = a feature of $H(K)^{zero}$)

The attracted feature F may be [±Interpretable]; however, F′ is always [-Interpretable]. The target K has interpretable features such as its categorial features, but they never enter into checking relations. In that sense, checking takes place to erase [-Interpretable] features in the target. The [+Interpretable] features, which do not have to be checked, have to remain even after checking took place, because they are needed for interpretation.

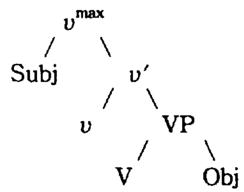
Formal features may also be classified based on the strength of whether they can incur overt movement. Thus, we say that the overt verb movement to T in French is due to a strong V-feature of T. For this Chomsky (1995, 234) proposes the so-called the Strong Feature Condition:

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

According to Chomsky (1995, 232), only features of functional categories are [±strong], and [+strong] features are checked only by categorial features of substantive categories. This means that only functional categories can overtly attract only lexical categories. In other words, no functional categories can overtly move, nor can lexical categories overtly attract other categories.

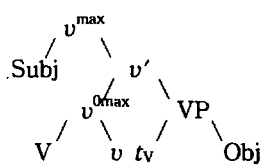
As an example, let us consider the argument structure of a transitive verb construction postulated in the recent minimalist framework:

(28)



In (28), the light υ is postulated following Larson's (1986) analysis of verbs like *give* and *put* which can take more than one internal arguments. In the Larsonian shell structure, the internal arguments occupy the positions of specifier and complement of V, and the external argument occupies the specifier of the light verb υ. This idea is extended to a transitive verb construction like (28) in which the internal argument Obj occupies the position within VP, and the external argument Subj occupies the specifier of v. In (28) V overtly adjoins to v due to the "strong" verbal affixal feature of the light verb, producing (29).[13]

(29)



13. V has to adjoin to υ before the external argument merges with υ^{max}.

If we merge (29) with the functional category T, we obtain (30).

(30)

```
            T^max
           /    \
          T      v^max
                /    \
             Subj     v'
                     /  \
                  V+v    VP
                        /  \
                      tV    Obj
```

The functional category T (perhaps, universally) contains the strong EPP feature D, but the derivation is not cancelled, because the T with the strong D feature is in the category T^{max} headed by T. This strong D feature gives rise to overt movement of the subject DP, merging it with T^{max} to check off the [+strong] D feature of T.

(31)

```
          TP
         /  \
       DP    T^max
            /    \
           T      v^max
                 /    \
             tSubj     v'
                      /  \
                   V+v    VP
                         /  \
                       tV    Obj
```

Note that in (31) not only the categorial feature D of the subject DP checks off the [-Interpretable] D feature of T but also nominative Case feature and Φ-features of DP, carried along as free riders, check off Case and Φ-features of T.[14]

3.2 Checking Domain

In the minimalist framework, all checking takes place within a

14. In fact, the verbal complex [V-υ] has to raise to T covertly in the case of English, but overtly in the case of French.

checking domain. Then, let's review the basic notions of domain and minimal domain. Suppose α is a feature or an X^0 category, and CH is the chain (α, *t*) or (the trivial chain) α. Then:

(32) a. MAX(α) is the smallest maximal projection including α.
 b. the domain δ(CH) of CH is the set of categories included in MAX(α) that are distinct from and do not contain α or *t*.
 c. the minimal domain MIN(δ(CH)) of CH is the smallest subset K of δ (CH) such that for any γ ∈ δ(CH), some β ∈ K reflexively dominates γ.[15]

Consider the following abstract structure

(33) a.
```
        XP
       /  \
     WP    X'
          /  \
         X    αP
             /  \
           YP    α'
                /  \
               α    ZP
```
b.
```
          XP
         /  \
       WP    X'
            /  \
      X^0max    αP
       /  \    /  \
      α    X  YP   α'
                  /  \
               t(α)   ZP
```

In (33a), MAX(α) is αP; the domain δ(α) of the trivial chain α is {YP, ZP} and whatever they dominate (note that ZP, YP, and the categories they dominate are distinct from α and do not contain α. The *minimal domain* MIN(δ(α)) of the trivial chain α, is defined as {ZP, YP} in (33a), because only the maximal categories can dominate any categories in the domain δ(α). The complement domain of α in (33a) is the set ZP and whatever it dominates. The minimal complement domain of α is its *internal domain*: ZP in (33a). The *residue* of α is its domain minus its complement domain: YP and whatever it dominates. Thus, the minimal residue of α is its *checking domain*: YP in (33a).

---

15. Here, the reflexive domination is understood as follows:

Reflexive domination
γ is reflexively dominated by a category β
 if either (i) γ = β,
  or (ii) γ is dominated by every segment of β)

On the other hand, In (33b), where α raised and adjoined to X forming the chain CH (α, t), MAX(CH) is XP; the domain δ(CH) is the set {WP, YP, ZP} and whatever they dominate (note that WP, YP, ZP, and the categories they dominate are distinct from CH and do not contain CH. The *minimal domain* MIN(δ(CH)) is defined as {WP, YP, ZP}, because only the maximal categories can dominate any categories in the domain δ(CH). The complement domain of CH in (33b) is the set YP, ZP and whatever they dominate. The minimal complement domain of α is its *internal domain*: YP and ZP in (33b). The *residue* of CH is its domain minus its complement domain: YP and ZP and whatever they dominate. The minimal residue of CH is its *checking domain*: WP in (33b). The checking domain is where a feature can enter into a checking relation with CH.

We must therefore understand the notion of domain and minimal domain of α *derivationally*, not *representationally*: they are defined "once and for all" for each CH as part of the process of introducing α into the derivation. If α is a trivial (one-membered) chain, then they are defined when α is lexically inserted; if α is a nontrivial chain (β_1, . . . , β_n), they are defined when CH is formed by movement.

4. Problems and Suggestions

As you have noticed, Chomsky's (1994, 1995) minimalist framework contains a number of "global" economy principles. To take an example, consider the Shortest Derivation Condition in (24):

(24) *Shortest Derivation Condition*

Minimize the number of operations necessary for convergence.

The definition in (24) is global in two ways: it refers to the number of steps in an alternative derivation, and it refers to convergence. In other words, we have to examine the number of steps in every convergent derivation to choose the most economical derivation as the admissible derivation. And, furthermore, when we apply some operation, we have to look ahead whether the application leads to eventual convergence or crash.

Let us consider the now-famous example in (34).

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

In the derivation (34a), *there* is merged in the Spec position of the embedded T to check off the strong D-feature of the T. Then, the expletive *there* is raised to the Spec position of the matrix T to check off the D-feature of the matrix T. On the other hand, in the derivation of (34b) *someone* is raised to the Spec of the embedded T to check the strong D-feature of the T; then, the expletive is inserted to the Spec position of the matrix T to check the D-feature of the matrix T.[16]

Chomsky (1995, 346) argues that Procrastinate forces us to choose the derivation in (34a) over (34b). Let us consider his argument. At some point of derivation, both derivations would have the structure in (35) in common.

(35) [$_{TP}$ to be someone in the room]

There are two possible ways to check off the strong D-feature of the infinitival T: as the specifier of the T, we can either merge the expletive *there* or raise *someone*. Chomsky claims that Procrastinate means that overt operation applies as late as possible, which forces us to choose the first option, yielding (36).

(36) [$_{TP}$ there [$_{T'}$ to be someone in the room]

Only (34a) is derivable from (36), raising the expletive *there* to the specifier position of the matrix T of (37).

(37) [$_{TP}$ seems [$_{TP}$ there to be someone in the room]]

At LF, the Case feature and Φ-features of the associate nominal *someone* are raised to the matrix T to check off the nominative Case assigning feature and Φ-features of the T.

16. The Case feature and Φ-features of the matrix T in both cases are checked by raising relevant features of the associate nominal *someone* at LF. Hence, this operation is relevant in counting derivational complexity of both derivations.

However, there is an example that contradicts the interpretation of Procrastinate that overt operation applies as late as possible, allowing for (costless) Merge to apply first. Consider the derivation of superraising construction in (17), repeated here.

(17) *[$_{TP}$[$_{\alpha}$ *John*] seems [that [$_{\beta}$ it] was told t_{John} [that he passed the exam]]]

At some point of the derivation of (17), we have the structure in (38).

(38) [$_{TP}$ was told John [that he passed the exam]]]

As has been the case in (35), we have two options: either inserts the expletive *it* or move *John* to the specifier position of the embedded T. According to Chomsky's (1995, 346) interpretation, we have to choose the first option, deriving (39), as we did in (36).

(39) [$_{TP}$ it was told John [that he passed the exam]]]

Eventually, (39) leads to the derivation of (17) that of course violates the MLC. The only way to save (38) is to choose the second option: overtly raising *John* to the specifier position of the embedded T, contrary to Chomsky's interpretation of Procrastinate.

However, since merging of the expletive *it* as a specifier of the embedded T eventually leads to the non-convergent derivation of (17), we may say that we can violate the principle of Procrastinate, overtly raising the object *John* to the spec position of the embedded T, because that is "required for convergence." But this means that Procrastinate is "global" in the sense that application of a certain operation is not determined on the basis of the information available in the structure to which it applies, but of the information whether its application will lead to an eventual convergent derivation. If we allow this kind of globality as a property of economy conditions, it seems that we do not need Procrastinate (and any other economy condition) to account for (34) and to block (17). What we need is a more general "economy" principle like (40):

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

As we have seen in (17) and (34), 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.

Suppose, following Collins (1997), that economy principles should be "local" and that there are only two "genuinely local" economy conditions:

(41) *Last Resort*

An operation OP involving α may apply only if some property of α is satisfied.

(42) *Minimality*

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

Collins' Last Resort does not differ from the original one in (16). (42), however, covering Chomsky's (1994, 1995) MLC, SDR, and perhaps Procrastinate (in the sense that LF operations are smaller), differs Chomsky's in that it is strictly local.

Collins (1997) claims that "local" (i.e., nonglobal) economy is superior to global one in a number of respects. First, it is empirically superior, as the analyses of locative inversion and quotative inversion constructions show:[17]

(43) a. John rolled down the hill.

b. Down the hill rolled John.

(44) a. "I'm so happy," Mary thought.

b. "I'm so happy," thought Mary.

Second, it allows optionality as in (43) and (44). Third, according to global economy, the grammar is able to count the number of operations in two different derivations to choose an optimal derivation. But this kind of derivation does not seem to be necessary in other parts of the grammar.

It is, however, obvious that Collins' local economy conditions are not enough to account for the problems related to (34) and (45).

17. I will not present how these constructions are analyzed; see Collins (1997) for the analyses.

(34) a. There seems [$_{TP}$ *t* to be someone in the room]
b. *There seems [$_{TP}$ someone to be *t* in the room]
(45) a. *John seems [that [$_{TP}$ it was told *t* [that he passed the exam]]]
b. It seems [that [$_{TP}$ John was told *t* [that he passed the exam]]]

As we have examined, at some point of derivation (34) and (45) will have (35) and (38) as their respective intermediate structures:

(35) [$_{TP}$ to be someone in the room]
(38) [$_{TP}$ was told John [that he passed the exam]]]

According to the Minimality Condition in (42), we have to merge the expletives *there* and *it* as Specs of TP in (35) and (38), respectively, because the operation Merge is a "costless" operation in Chomsky's (1995) system. As we have seen, in the former case we derive the grammatical sentence in (34a) by raising the expletive *there* from embedded Spec position to the matrix Spec position. On the other hand, no grammatical sentence can be derived from the latter case; if *John* is raised to the matrix Spec as in (45a), the MLC is violated, whereas Last Resort is violated if the expletive *it* is raised to the matrix Spec as in (46).[18]

(46) *It seems [that [$_{TP}$ *t* was told John [that he passed the exam]]]

It seems that we have to assume that the operation Merge is also a "costly" operation like the operation Move (see Collins (1997), if we want to keep the Minimality Condition (42). In other words, if we

18. It is not obvious that (46) violates the principle of Last Resort. Raising *it* to the matrix Spec certainly violates Last Resort from *its* point of view; *it* no longer contains an -Interpretable feature (i.e., nominative Case feature). Suppose that raising the embedded subject *it* to the matrix Spec is a violation of Last Resort. Then, this raising operation (violating Last Resort) cannot be taken to be a smaller OP' (satisfying Last Resort) than raising John to the matrix Spec (satisfying Last Resort) as in (45a). If it is correct, we need a principle like MLC.

We may say that raising *it* as in (46) does not violate Last Resort, because *it* can check D- and Φ-features of the matrix T; (46) is ungrammatical, because both the matrix T and *John* contains an -Interpretable nominative Case feature at LF. If it is correct, (46) violates the Minimality Condition in (42).

assume that Merge is a "smaller" operation than Move, there is no way to derive a well-formed expression from a structure like (38), because the "smaller" operation Merge always merges the expletive *it* as Spec. Then, what makes *John* move to Spec position of the embedded T in (38)? I propose that we have the following Maximality Condition:

(47) Maximality
Maximize the effect of an operation.[19]

Suppose that, as Spec of the embedded T, we can either merge *it* or move *John* in (38). We choose the latter, since moving *John* to the embedded Spec position brings about a maximal effect on (38) in the sense that the operation eliminates *John* with an Uninterpretable Case feature. On the other hand, the result of merging *it* as the embedded Spec in (38) still leaves *John* with Case feature, which has to be erased for LF interpretation. In the case of (35), both operations are available: that is, we can either merge *there* or move *someone* as the embedded Spec., making both sentences in (48) derivable from the intermediate structure of (35).

(48) a. Someone seems [*t* to be *t* in the room]
b. There seems [*t* to be someone in the room]

Notice that the Maximality Condition blocks the derivation of (34b), because it contains an "incomplete" chain headed by *someone*. In stead of merging *there*, moving *someone* as the matrix Spec forms a complete A-chain, maximizing the effect of the operation.

In summary, I have argued that it is possible to eliminate "global" economy conditions from grammar. Furthermore, I have argued that Collins' (1997) Minimality Condition is not enough to derive structures

19. The Maximality Condition is interpreted intuitively as follows:

An operation OP (satisfying Last Resort) may apply only if there is no bigger (i.e., more effective) operation OP' (satisfying Last Resort).

The Maximality Condition may also be termed as the Complete Chain Condition in the sense that completing a chain is the most effective operation.

like (45b), and that we need a new principle called the Maximality Condition to complete the job.

References

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

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

Chomsky, N. (1991) "Some notes on economy of derivation and representation", R. Freidin, ed., *Principles and Parameters in Comparative Grammar,* 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.

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

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

Emonds, J. (1978) "The verbal complex V'-V in French," *Linguistic Inquiry,* 9: 151-175.

Hale, K., and S. J. Keyser (1993) "On argument structure and the lexical expression of syntactic relations", K. Hale and S. J. Keyser, eds., *The View from Building 20,* Cambridge, Mass.: MIT Press.

Kayne, R. (1994) *The antisymmetry of syntax*, Cambridge, Mass.: MIT Press.

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

Larson, R. K. (1988) "On the double object construction", *Linguistic Inquiry,* 19, 3, 335-91.

Pollock, J.-Y (1989) "Verb movement, UG and the structure of IP", *Linguistic Inquiry,* 20, 365-424.

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

Lee, Hong-Bae
Department of English
Seokang University
Seoul, 121-742, Korea
Fax: +82-2-705-8751