# In terms of Merge: Copy and Head Movement*

*Jonathan David Bobaljik, Harvard / McGill*

The primary aim of this paper is to derive certain theory-internal assumptions from more basic, though no less theory-internal assumptions about the nature of the derivation. In doing so, certain implications of the view advocated here for other aspects of the syntax are considered. It may turn out that the end result is solely an exercise in formalism. Then again, it may not.

The framework assumed throughout is a particular instantiation of more general ideas presented in Chomsky (1992, 1995). Within this framework, the syntax is characterized by two structure-building operations: *Merge* and *Move*. There is an oft-noted redundancy between the two procedures, but attempts to eliminate this redundancy have succeeded only in as much as the second operation, Move, may be defined as a two step procedure of which the first operation, Merge, is a one step. In this, the syntax then consists of the simple operation, Merge, and the more complex operation, Copy + Merge. In this short paper, I argue that the (sub-)operation Copy is straightforwardly deriveable from more basic assumptions. That is, I claim that there is only one syntactic operation—Merge—and that the apparent additional "step" of Copy is epiphenomenal, in a large part an artifact of phrase-structure notation. I will also argue that stipulated differences in the character of Move as it applies to XP-movement (substitution) and head-movement (adjunction) are superfluous, and arise only when the derivation is expressed in terms of phrase markers.

The paper is organized as follows: In section 1, I discuss the standard definitions of Merge and Move internal to this framework. I focus in particular on the apparent necessity of the additional step, Copy, in the latter operation. Next, I turn to the proposal at hand. Following Chomsky (1995), I claim that phrase markers are convenient representations of individual terms in a derivation, but not of individual stages of a derivation. I suggest that each stage in the derivation is characterized rather by an (unordered) set of the terms

* This paper reflects a number of ideas developed in parallel with my dissertation, though most of the material reported here is not in that work (with the exception of section 3). The work reported here is still very much in progress and comments are appreciated. While the ideas here have been discussed with a great number of people, I would like to thank especially Andrew Carnie, Noam Chomsky, Chris Collins, and Howard Lasnik for extensive comments on a much earlier draft of this paper. Errors of fact and interpretation are of course my own. This research was funded in part by a Mellon Dissertation grant (1994-1995).

defined to that point in the derivation. Phrase markers add additional, and in certain instances, unwarranted information and restrictions to the computation. I argue in section 2 that syntactic operations are defined as procedures relating successive stages of the derivation, and not, as standardy assumed, procedures mapping one phrase marker to another. Paring away the additional structure created by the notation of phrase-structure / X'- schemata, I demonstrate that there is no necessity for a Copy operation. The effects of such an operation fall out from the standard characterization of Merge without any additional machinery. In this section I argue that the exceptional nature of head-movement with respect to the Extension Requirement (Chosmky, 1992) is also reducible to the added constraints of phrase marker notation, and is thus not a property of the syntactic computation. Finally, in section 3, I discuss some implications of this characterization of the syntactic computation for the definition of other theory-internal concepts, including a definition of the output representation.

Prior to the discussion proper, it is worth clarifying a point of terminology. In what follows, I will refer to the elements of the syntactic computation as *terms*, following Chomsky. Terms are the pieces which are ultimately arranged to form the output representation(s). These correspond to the nodes of the familiar phrase markers. It will simplify the exposition somewhat to distinguish two types of terms: initial and complex. Assuming that the derivation begins as a list of the pieces which are to be concatenated, i.e. Chomsky's (1992) Array or (1995) Numeration, we may call these pieces the *initial terms*. These correspond to the terminal nodes of X-bar theory, though we now approach the derivation from the bottom up: these are the nodes which begin the derivation. I assume that the function of the derivation is to define a specific arrangement or concatenation of these terms, i.e., an output representation which meets certain demands imposed by the interfaces with other aspects of the computational system. The derivation proceeds incrementally, and thus at each step of the derivation, a new term is defined with other terms as constituents. We may call the new terms defined in the course of the derivation *complex terms*. Once again, this distinction will play a role only for the purposes of exposition; the underlying assumption is that the syntactic concatenative operation is blind to the internal make-up of terms. The operation Merge, as I will argue below, has no inherent constraints on the nature (initial versus complex) of its input terms; the operation simply Merges any two terms.[1]

## 1. Merge and Move

Chomsky (1992 et seq) characterizes the syntactic component of the computational system of language as a sequence of structure-building operations. The elements of a defined input set (our initial terms) are manipulated and concatenated to build a complex structure, namely the phrase-

---

[1] I assume without discussion that the operation is binary, though this is not necessary in what follows. Groat (1995) and Collins (this volume) suggest that the binarity of Merge is derivable from considerations of economy.

marker(s) which serve(s) as the input to other components of the language faculty, especially the (morpho-)phonological component(s)—PF—and the logico-semantic component—LF. In part resurrecting ideas from early generative work, Chomsky (1995) proposes that there are exactly two such structure-building operations: Merge and Move:[2]

(1) Applied to two objects $\alpha$ and $\beta$, *Merge* forms the new object $\gamma$. (p.396)

(2) Applied to a phrase marker $\beta$, with constituents $\alpha$ and $\delta$, *Move* copies $\alpha$ and Merges $\alpha$ with $\beta$ to form the new object $\gamma$. (cf. p.399)[3]

For ease of reference, let us call the terms $\alpha$ and $\beta$ in these definitions the *input terms*. Obviously, there is a not-insignificant overlap between the two operations. In particular the first operation, Merge, is a proper sub-part of the second operation, Move. In addition, the nature of the output category $\gamma$ is common to both operations. Thus, for both operations, given input terms $\alpha,\beta$:

(3) (i) $\gamma$ is the set $\{\alpha,\beta\}$ [$\alpha,\beta$ are said to be the *constituents* of $\gamma$]
and
(ii) $\gamma$ inherits the relevant properties of either $\alpha$ or $\beta$
["one or the other *projects* and is the *head* of $\gamma$." (p.397)]

We lay aside (3ii) – i.e. feature / checking theory – for a while, as it is not immediately relevant. We return to this topic in section 3.

Again, comparing (1) and (2) we find that, on the characterization offered by Chomsky (1995), the operation Move (2) includes as a component "step" the operation Merge (2). The former operation is characterized by the additional step, Copy (i.e., "leave a trace" (p. 399). Moreover, there is no part of the operation Merge which is not a part of the operation Move. Move is Copy + Merge.

This redundancy between the two operations has been noted before, in particular by Kitahara (1995) and Epstein (1995). Both authors make suggestions towards eliminating or reducing the redundancy in the system. Kitahara (1995) suggests that there is a single strucutre-building operation , Target-$\alpha$, which is for all intents and purposes Merge as defined in (1). Epstein (1995) also takes the operation in (1) (which he calls Concatenate) to be "common to or shared by both Merge and Move alike" (p. 11). While both of these works take steps towards reducing the redundancy between Move and Merge, neither diverges significantly from Chomsky's position. All of these

---

2 Chomsky proposes one further operation, namely *delete* (p.400). Discussion of this operation is orthogonal to present concerns, though.

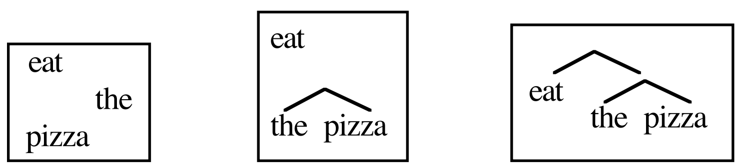
3 I have reworded Chomsky's definition to make it more consistent with the definition of *Merge*, above. Chomsky's definition of *Move* is:

(i) Given the phrase marker $\Sigma$ with terms K and $\alpha$, *Move* targets K, raises $\alpha$, and merges $\alpha$ with K to form the new category $\gamma$ with the constituents $\alpha$,K. (p.399) approaches posit a basic concatenative operation, Merge (1), plus an aditional copying operation which is implicated in Move.[4]

Common to Chomsky (1995), Kitahara (1995), Epstein (1995), and work deriving from these is a view of syntax whereby the basic concatenative operations manipulate the input terms. By this, I mean that the terms to which the structure-building operation(s) apply are themselves affected by the operations. Using a common metaphor, the syntactic computation can be viewed as taking place in a *workspace*. The initial terms of the derivation (Chomsky's Numeration) are entered into this workspace and the operations concatenate these terms. To take a hypothetical example, consider a partial derivation which begins with the terms *eat, the,* and *pizza*, which will lead to the VP consisting of the verb and its object. Graphically, we may depict this by means of the following schema:

(4) A partial derivation

a. step 0 (Numeration) b. step 1 Merge (the,pizza) c. step 2 Merge (eat, {the,pizza})




With each application of *Merge*, two terms in the workspace—either initial or complex—are joined together to form a single complex term. In the phrase-structure representations in (4), we see that the operation Merge thus has the effect of reducing the number of independent phrase-markers in the workspace by one. In (4a), there are three (trivial) phrase-markers, corresponding to the initial terms. After one application of Merge (4b), there are only two independent phrase-markers, i.e. two root nodes, in the workspace. This number is subsequently reduced to one by yet another application of Merge (4c). Though it is rarely stated explicitly, it is worth keeping in mind that the bottom-up theories of phrase-structure-building require that phrase-markers be built in parallel, and eventually joined. Consider, for example, a slightly expanded example, i.e. the derivation of (4) plus a subject such as [a monster]. To the workspace in (4) we add two terms: {a} and {monster}. It is clear that the operation of Merge which joins these two terms operates without (yet) reference to the phrase-marker corresponding to the verb + object. It is only after the construction of the complex term corresponding to the subject DP that an

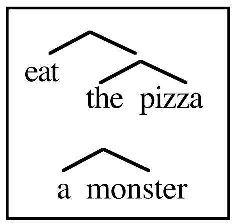
---

4 Chomsky (fall lectures 1995) considers a different tack. Here, it is suggested that the basic operation is *Move* (being *Merge + Copy*), and that the operations characterized as *Merge* involve an additional step of *Deletion*, i.e., of the "initial copy". This position is somewhat closer to the approach I will take below. It is, however, not significantly different from the theories just discussed for present purposes; it distinguishes two operations with one characterized as a suboperation of the other.

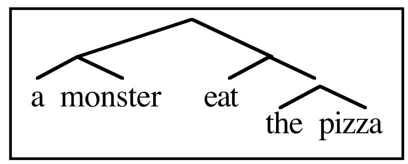
operation of Merge can apply to join two complex terms: the subject (DP) and predicate (VP):

(5) Continuing from (4)...

a. Merge (a,monster)




b. Merge (DP,VP)

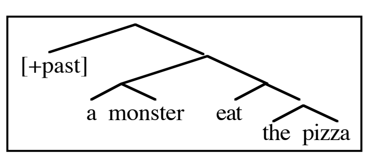



Even though not all applications of Merge overlap in the content of their input terms, they all have the effect of reducing the number of independent phrase-markers in the workspace by one. Thus, (5a) contains two phrase-markers, and an application of Merge (5b) joins these into one.

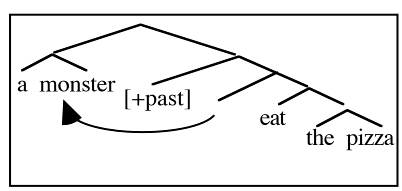
The operation Move does not, under the standard characterization, alter the number of phrase-markers in the derivation. In terms of the procedural aspects of the operation, it is in this that Move differs from Merge. To illustrate, extend the derivation in (4) and (5) yet further. Assume that an operation of Merge has joined the initial term [$_{\text{Infl}}$ +past] with the complex term depicted in (5b). We assume, of course, that the Infl term was "present" in the workspace from the beginning of the derivation and omitted from (4) and (5) for expository convenience. The step which we would like to consider now is the "raising" of the subject from (what is essentially) Spec,VP to Spec,IP. That is, consider the operation which takes (6a) and maps the structure to (6b):

(6)

a. Merge (Infl,VP)



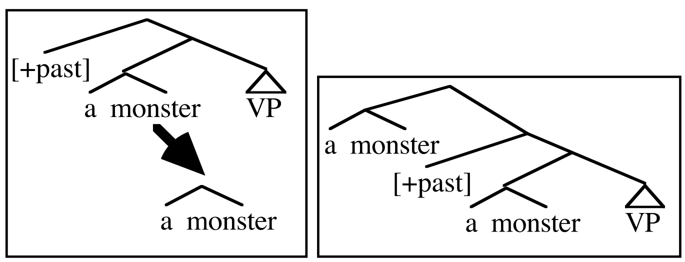

b. Move




While Merge is characterized as an operation which joins two independent phrase markers (i.e. root nodes), reducing the number of such nodes in the workspace, Move does not alter the number of such nodes; it operates internal to a single phrase marker. For this reason, an additional mechanism is necessary to express Move in terms of Merge in the manner discussed above. This extra mechanism is Copy. Informally, the operation Copy creates a new, independent phrase marker by copying a sub-term of some root node. This copy is then available as a new term to be Merged with the root node, as usual reducing the number of phrase markers by one. In other words, the approach taken in Kitahara (1995) and Epstein (1995) to eliminating the redundancy between Move and Merge—i.e. expressing Move as involving Merge as a component of the complex operation, has a cost. This cost is the postulation of an extra operation, Copy. In order for Merge to have two, independent phrase-markers to conjoin, the procedure must first create a new phrase-marker where there isn't one. While Move has the net effect of neither reducing nor increasing the number of independent phrase-markers in the workspace, this is acheived proceduraly by first increasing and then decreasing the number of independent phrase-markers. Graphically, then:

(7) Move is Copy + Merge

a. Copy (+1 phrase-marker) b. Merge (-1 phrase-marker)




2 root nodes 1 root node

I assume without argument that the evidence for traces (i.e. a copy—or partial copy—in the position prior to movement) is compelling. This precludes a version of this operation without Copy, whereby the element to be moved is actually removed from the lower position, and then Merged with the root node. This would have the same effect, increasing and then reducing the number of root nodes in the workspace, however, the final output of the derivation would contain no history of the derivation; I will therefore not consider such an approach here.

A complete reduction of either Move to Merge or vice versa would appear to be blocked by the need to retain a copy of the moved element in the base position. On the assumption that the syntactic operations manipulate the formal objects involved in the derivation, the operation Move must be distinguished from Merge by the extra step of Copying; Move is Copy + Merge, essentially as proposed in Chomsky (1995), Kitahara (1995), and Epstein (1995).[5]

---

[5] Or *Merge* could involve an extra step, not involved in *Move*, as suggested by Chomsky (fall lectures, 1995) (see the previous footnote). Either way, the point is that neither is wholly reducible to the other on these assumptions.

Recent work, most significantly Brody (1995), Chomsky (1995), Epstein (1995), and Kayne (1995), has challenged the notion that the X'-theoretic representations or phrase markers are axiomatic components of the theory of grammar. In different ways, all these works propose that the familiar aspects of these representations are deriveable from more basic assumptions, and significantly, that the representations add additional, arguably unmotivated, information (see especially Epstein, 1995, and the discussion in Chomsky, 1995, p.397).

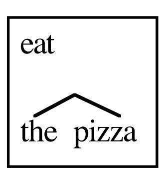
At this point, I would like to approach the problem of collapsing Move and Merge from this perspective. In particular, I will show that the need for Copy arises because of the properties of the phrase marker notation. In the next section, I argue that, if, following Chomsky (1995), we take this notation to be merely an expository device, used to represent select terms in the derivation, then the need for a Copy operation disappears. The apparent effects of such an operation, when comparing two phrase-structure representations, are derived from the fact that the syntactic operations act on terms, whereas the phrase-markers are simply translations of the more basic terms into a familiar notation. Hopefully, this will become clearer as we proceed.

## 2. Syntax without phrase markers

### 2.1. Deriving Copy

Consider again the derivation of the VP [eat the pizza] in (4). At the initial stage the workspace consists of three terms: {eat}, {the}, and {pizza}. Each of these corresponds to a phrase-marker in the representation in (4a). As discussed above, the first operation of Merge as characterized in (1) takes two terms as its input, namely {the} and {pizza}. This operation then defines or *derives* the new term $\gamma$, i.e., the set consisting of the two input terms: {{the},{pizza}} (see (3)). In terms of phrase-structure notation, this operation reduced the number of root nodes (i.e., independent phrase markers) in the workspace, as in (4b) repeated here as (8):

(8) =(4b)




Consider now the terms in the workspace in (8). In particular, ignoring the notation of phrase markers and looking instead at the actual terms involved, the workspace in (8) contains *four* terms: the three initial terms plus the complex term defined by Merge (the order of terms in the list is irrelevant):[6]

(9) The terms of (8).

{eat}
{the}
{pizza}
{{the},{pizza}}

At the same time as the operation Merge decreases the number of root nodes, the operation *increases* the number of terms in the workspace (by exactly one). Compare the schematized workspace (8) after the application of Merge with the list of terms contained in that workspace (9). The subsequent application of Merge schematized in (4b) will take as its input the two terms {eat} and {{the},{pizza}}, and create exactly one new term: {{eat},{{the},{pizza}}}. In terms of phrase-markers, recall that this reduced the number of root nodes by one (4c), but again, it increases the membership in the list of terms in the derivation by one. The relevant derivation is now characterized by the following tems (compare (9)):

(10) The terms in (4c).

{eat}
{the}
{pizza}
{{the},{pizza}}
{{eat},{{the},{pizza}}}

Following Chomsky (1995), I propose that the derivation is not carried out in terms of manipulation of phrase markers of the type used in (4)-(7). I suggest rather that the syntactic computation is process of definition of (complex) terms.[7] That is, I claim that the basic operation, Merge, as characterized in (1)

---

[6] I am proposing nothing new here, of course. This part of the discussion makes explicit certain aspects of the theory as presented in Chomsky (1995) which are less salient in other work. For instance, Chomsky's definition of terms has exactly the result I am giving here: the four terms in ($15) are involved in the derivation after an application of Merge has created one new term, though for Chomsky the set of terms in the derivation is apparently defined relative to the structures and not derivationally, i.e. in terms of Merge.

(i) For any structure K:
K is a term of K
if L is a term of K, then the members of the members of L are terms of K. (p. 399)

[7] See Watanabe (this volume) for a discussion of the derivation in terms of definition and redefinition of terms. Watanabe derives the Strict Cycle, both for overt and covert movement, from a principle of Economy which requires minimal redefinition of terms.

operates not on phrase-structure representations, but rather on (unordered) lists which may be represented similarly to those in (9) and (10).

Consider, then, the operation of "raising the subject" in these terms. In particular, (11a) is the list of terms of the derivation corresponding to the phrase marker (6a) above—i.e., the point immediately prior to "movement."

(11) The terms of (6a)

{Infl}
{a}
{monster}
{eat}
{Fido}
{{eat},{Fido}}
{{a},{monster}}
{{{a},{monster}},{{eat},{Fido}}}
{{Infl},{{{a},{monster}},{{eat},{Fido}}}}

From this list, we select two terms, say the seventh and ninth in (11), though this order is simply an artifact of the exposition. We then apply the operation Merge to these two terms, that is, these terms from (11) should be the input terms (i.e., $\alpha$,$\beta$) to the operation (1). The output of this operation is the complex term (12a). This expands the list of terms in the derivation by exactly one term, and the resulting list is given in (12b). For convenience, I add the new term at the bottom of the list, though I stress again that the order of the list is arbitrary for presentation and irrelevant for the computation.

(12) a. $\gamma$ = {{{a},{monster}},{{Infl},{{{a},{monster}},{{eat},{Fido}}}}}

b. {Infl}
{a}
{monster}
{eat}
{Fido}
{{eat},{Fido}}
{{a},{monster}}
{{{a},{monster}},{{eat},{Fido}}}
{{Infl},{{{a},{monster}},{{eat},{Fido}}}}
{{{a},{monster}},{{Infl},{{{a},{monster}},{{eat},{Fido}}}}}

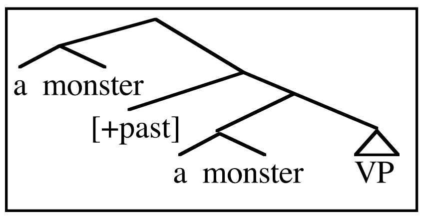
A phrase marker representation is equivalent to an individual term, i.e., a phrase marker is a representation of a specific term. A phrase marker is *not* a

---

The procedures which I entertain in this paper involve no redefinition of terms. Watanabe's conclusions concerning the overt syntax for XP-movement carry over directly to the present paper. For LF-movement and head-movement, though, I take a different approach in Bobaljik (1995,ch6) and section 3, below.

representation of stage of the derivation. I have argued above that any stage in the derivation is characterized by a(n unordered) set of terms, and is thus not accurately represented by a single phrase marker in all cases. The phrase marker corresponding to (12a) is (7b), repeated here. Indeed, this term/phrase marker contains two copies each of the terms {a}, {monster}, and {{a},{monster}}. However, this term is derived not from any one other term (or phrase marker) but rather from the set of terms defined to that point in the derivation, i.e., the list in (11). This is the result of simple application of the rule Merge as expressed in (1).

(7b), repeated




When we considered this step of the derivation originally in terms of manipulation of phrase-markers, we were forced to posit an independent operation or sub-operation, Copy, which created the copy of the element to be moved, in order for that copy to serve as the input to the subsequent (sub-)operation Merge. Recall that this was the sole significant difference between Move and Merge, and was thus the stumbling block to collapsing the two significantly redundant operations into a single, unified syntactic operation. Now, abandoning the phrase marker notation and looking only at the (list of) terms involved in the derivation, we find that straightforward mechanical application of the procedure (1) characterises both putative operations, Move and Merge. The copy "appears" only as an artifact of the notation, in the translation of a given term into a phrase-marker.

The operation Merge is creative—each application defines exactly one new term in the course of a derivation. In the discussion surrounding (4)-(7), there was the appearance of a reductive effect; the operation appeared to be reducing the number of root nodes in the derivation's workspace. However, as argued by Chomsky (1995), the "nodes" of phrase-structure representation are added for expository clarity, and should therefore play no role in the explanation. What is being "reduced by one" with each application of Merge is a reduction in notation and notation only. The true effect of the operation is in fact an increase in the number of terms involved in the computation.

To restate the central point: I propose that the rule in (1)—i.e, Merge—does not relate two phrase markers (e.g., (4b) to (4c)). Rather, the rule relates two subsequent stages of the derivation, e.g., (11) to (12b).

The effect of "copying" falls out of the unconstrained mechanical application of (1), Merge, to the terms of the derivation. Thus far, we assume that the input terms, $\alpha$ and $\beta$, may be in principle any two terms defined in the course of the derivation, initial or complex. For all terms, the output is defined as the set consisting of the input terms and nothing more (feature theory (3ii) aside). If some term happens to be a sub-term of both $\alpha$ and $\beta$, then the output term will contain "copies" when translated into a phrase-structure representation. However, no distinct process of copying (or deletion) is assumed.

I have demonstrated that (2)—the operation Move— is therefore superfluous as a distinct syntactic operation. Abandoning phrase-structure representations for matters beyond convenient exposition, we complete the reduction of the two syntactic operations to a single one, Merge. Obviously, this is a desirable consequence only if the simplification in this aspect of the theory does not lead to equal or greater complication elsewhere. A full discussion of this is well beyond the scope of this paper. In the next section, though, I scratch the surface of this problem, and note certain implications for the view of syntactic operations presented above for other definitons and concepts within the broad domains of the framework assumed. The section will be quite brief at this time. Further relevant discussion is to be found in Bobaljik (1995: 297-323). Highly relevant also are parts of Kitahara (1995), Epstein (1995), Lasnik (1995), Groat (1995), Watanabe (this volume), Collins (this volume), and others.

## 2.2. Head movement and the Extension Requirement[8]

Above, I noted that the view of the derivation which I am entertaining is creative. That is, a defining property of the operation Merge as characterized above is that it derives exactly one new term with every application. That is, the procedure as I have proposed it in (1) (Chomsky's Merge) never alters the definition (constituency) of either of its input terms; there is no operation which takes as its input two terms $\alpha$ and $\beta$, and does not create from them a new term $\gamma$. Watanabe (this volume) demonstrates that a ban on redefiniton of terms will derive a strong form of the Extension Requirement of Chomsky (1993), and thus derives a version of strict cyclicity.[9]

(13) *The Extension Requirement*

a. Syntactic operations must extend the phrase-marker.

---

[8] As this goes to press, Sam Epstein has drawn my attention to the work of Samuel Brown (Harvard) who has independently observed the fact that trans-arboreal movement is not excluded in the current framework, and that head-movement is thus not a problem for the Extension requirement in the manner discussed here.

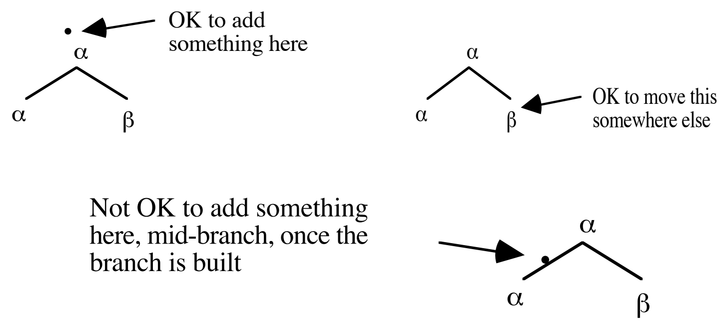
[9] Watanabe's version of strict cyclicity is stricter than Chomsky's, with implications for LF-movement and head-movement as well. See Watanabe for more detailed discussion, and Kitahara (1995) for a different view on the derivation of the Extension Requirement.

or, equivalently:

b. The input terms to syntactic operations must be root nodes.

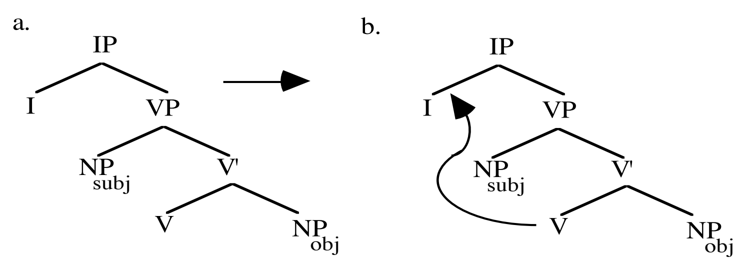
To illustrate, given a phrase marker as in (14), only operations (Move and Merge) "joining" terms at the root of this phrase marker are permitted by (13):

(14) The effects of the Extension Requirement, in terms of trees:




Chomsky (1992) noted that there were two apparent classes of exceptions to this requirement. The first exception was that the requirement could not hold in the LF component, since operations such as "covert object shift" (the movement of an object from a VP-internal position to a functional specifier position below the root node) would involve exactly the operation banned in (14). That is, the target of the operation is a sub-term of a larger phrase-marker; in Watanabe's and our terms, LF movement of this sort would require redefining terms created earlier in the derivation. The second exception is head-movement via adjunction. This operation also does not extend the entire phrase marker. For example, if Infl is Merged with a VP, projecting IP, then subsequent adjunction of the verb to Infl does not extend the phrase-marker IP. The adjunction operation would have to redefine IP as illustrated in (15):

(15) Merge Infl & VP, then move V = ER violation




$IP = \{\{I\},\{\{NP\},\{\{V\},\{NP\}\}\}\}$

$\rightarrow$ $IP = \{\{\{I\},\mathbf{\{V\}\}},\{\{NP\},\{\{V\},\{NP\}\}\}\}$

While Chomsky (1993) stipulated that these two classes of operations were exceptions to the Extension Requirement,[10] such a tack is neither appealing nor available on the view presented above. That is, since I wish to characterize the syntactic computation solely in terms of definition of (i.e. creation of) terms, there is no room in this system for procedures which redefine existing terms.

I will not discuss the problem of LF movement here. In Bobaljik (1995:chapter 6), I propose that the copy theory of movement allows a straightforward resolution of this apparent problem. There, I suggest that all movement is "overt" in the sense that the syntax proceeds in a single cycle, deriving a single ouput representation which is fed to both the semantic-interpretive (LF) and morpho-phonological (PF) components of the grammar. The effects of "covert" versus "overt" movement obtain in the phonological component: if the higher copy of a given term is pronounced (lower copies being deleted) then we have the effect of overt movement; if the lower copy is pronounced (higher copies being deleted) then we have the effect of covert movement. Approahces of varying degrees of similarity are proposed in Groat & O'Neil (1994), Brody (1995), and Pesetsky (in prep), among others.

However, the apparent problem of head-movement (by adjunction) remains. One possibility is that all head-movement is substitution. I will not consider the approach here.[11] Instead, I maintain that this problem, too, is only apparent—again an artifact of the representational constraints of phrase marker notation.

The characterization of Merge which I gave in (1) places no inherent restrictions on what two terms may count as input terms; any two terms previously defined in the course of the derivation (either initial or complex) may serve as $\alpha$ and $\beta$ in the definition in (1). Moreover, there is no restriction prohibiting any given term from serving as the input to more than one operation of Merge. Indeed, in the discussion of the effects formerly attributed to Move, we saw that such a restriction could not hold. Thus, in the (partial) derivation leading to the list in (12), there have been two operations which have the term {{a},{monster}} as one of their input terms. One step in the derivation Merges this term with the term {{eat},{{the},{pizza}}} to form the eighth term in the list in (12b), and another operation of Merge takes {{a},{monster}} again as one input ($\alpha$), with the other input term ($\beta$) being {{Infl},{{{a},{monster}},{{eat},{{the},{pizza}}}}. this forms the complex

---

[10] Kitahara (1995) derives these exceptions from other assumptions, though that paper maintains the crucial approach of structure building which prevents a true collapse of *Move* and *Merge* as discussed in section 1, above. Watanabe (this volume) also offers a version of the extension requirement which differs from Chomsky's and derives the effects of these stipulations from an economy condition.

[11] This approach is generally discarded out of hand in the literature. See Kitagawa 1986 for one well articulated approach to verb raising in English and Japanese which at least partially appeals to this mechanism.

term in (12a), i.e. the final term in the list in (12b). These two steps are repeated as (16):

(16)

a. Merge $\alpha$ = {{a},{monster}}
$\beta$ = {{eat},{{the},{pizza}}}

b. Merge $\alpha$ = {{a},{monster}}
$\beta$ = {{Infl},{{{a},{monster}},{{eat},{{the},{pizza}}}}

In the case of the latter operation (i.e. (16b)), the terms {a}, {monster}, and {{a},{monster}} occur as subterms of both input terms. Recall that this situation, when translated into phrase structure notation, resembles (XP) movement, specifically substitution. Now consider the following partial derivation:

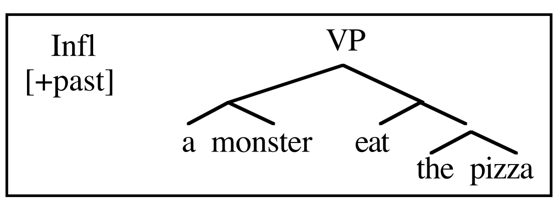
Begin with the point at which the following list characterizes the derivation thus far:

(17) a. The terms of the derivation, stage *n:*

{Infl}
{a}
{monster}
{eat}
{Fido}
{{eat},{Fido}}
{{a},{monster}}
{{{a},{monster}},{{eat},{Fido}}}

The independent root nodes of phrase-structure based accounts would be:

b.




Now, assume that the language in question is English-prime: a language like English but which requires overt raising of the verb to Infl. Taking for granted that all movement is driven by feature-checking (see the next section), assume further that the verb and Infl must come to be in a checking relation, and that this relation is to be established via Merge. Thus, the next step of the derivation is:

(18)

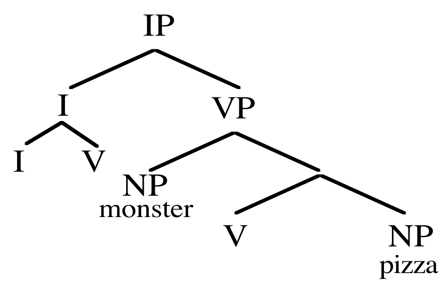
Merge $\alpha$ = {Infl}
$\beta$ = {eat}
result: $\gamma$ = {{Infl},{eat}}

As always, the output of the operation, $\gamma$, is added to the list of terms in the derivation. Presupposing some suggestions of the next section, we assume that the featural properties of $\gamma$ are those of one of its components, i.e. its *head*. In the case of (18), assume that $\gamma$ has the relevant syntactic properties of {Infl}. In (11) and (12) we assumed that {Infl} was Merged with (what corresponds to) the VP. Instead, following on from (18), we Merge the complex term {{Infl},{eat}} with the VP. Since this complex term by hypothesis is a projection of {Infl}, it has the relevant features of Infl and for the purposes of feature checking, the result of merging {{Infl},{eat}} with the VP should be the same as merging {Infl} with the VP. Hence:

(19)
Merge $\alpha$ = {{Infl},{eat}}
$\beta$ = {{{a},{monster}},{{eat},{{the},{pizza}}}}
result $\gamma$ = {{{Infl},{eat}},{{{a},{monster}},{{eat},{{the},{pizza}}}}}

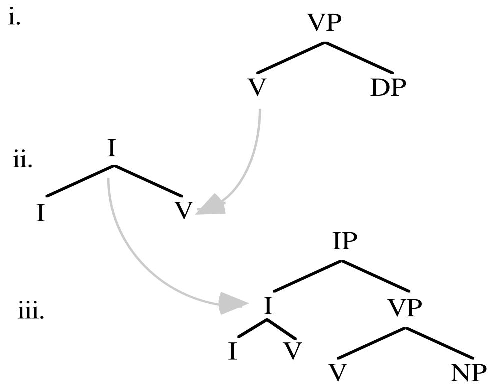
The next step, raising of the subject, proceeds as in (11)-(12). Translating the latest term created by (19) into a phrase marker, we have:

(20) $\gamma$ = {{{Infl},{eat}},{{{a},{monster}},{{eat},{{the},{pizza}}}}}




This, of course, is the desired result. The operations used to create this phrase marker from the workspace list in (17)—namely (18) and (19)—clearly meet the structural description of (1). Where is the trick? There is none. Like the case of XP-substitution considered in the previous section, the apparent "trick" comes from viewing the derivation as a single rule, (1), operating on set-theoretic objects—terms—and not on the more complex phrase markers. Expressed as operations on the phrase markers in (17b), the operations in (18) and (19) would amount to movement across phrase markers, expressed as in (21):

(21) From tree unto tree...




Once again, the notation of phrase markers adds unnatural and unwarranted restrictions to the basic syntactic operation (1). Phrase markers are convenient representations of individual terms, but terms are not themselves individual stages of the derivation. The stages are sets of terms. Merge must be defined not on these convenient representations, but rather on the sets of terms which constitute the true objects of the syntactic derivation. In the previous section, we saw that Copy, the operation distinguishing Move from Merge, was superfluous; the necessity of such a distinction was warranted only by artificial constraints imposed by the notation of phrase markers. While there is indeed a copying effect to the derivation, this effect is acheived through eliminating these constraints and not through the postulation of additional mechanisms. In a similar spirit, we have seen here that the differences between head-movement (adjunction) and XP-substitution, i.e. the two core cases of Movement, are similarly superfluous. The Extension Requirement is an axiomatic part of the definition of Merge: the operation simply and solely derives new terms. Assuming that the question of LF movement can be resolved,[12] the otherwise stipulated exception of head-movement is also unnecessary. Straightforward, mechanical application of the single rule (1) will serve to derive all these operations, without further assumptions, if the rule operates on the most basic formal objects of syntax: the sets of terms characterizing stages of the derivation, and in particular if the rule is taken simply to recursively define new terms from the set of previously defined terms.

## 3. Constraining the system: Inclusiveness, Last Resort, and Shortest/Closest

In the preceding sections, I have defined a powerful rule, Merge, and claimed that this rule forms the basis of the syntactic derivation. I have shown that the appearance of other operations (Move/Copy) and certain conditions on their application (the exceptional nature of head-movement) are artifacts of the

---

[12] See references above.

notation we use to express the relations among steps of the derivation. In particular, the null hypothesis would be that there are no restrictions on the characterization of the input to (1); any two terms may be Merged and the output is always defined, by (3). This state of affairs obviously requires some constraints. I will not devote any time in this paper to an investigation of the constraints on the input to Merge. Rather, I will state briefly how well-motivated constraints from the general framework assumed here may be stated under this view, where Merge operates on terms and not on phrase markers.

Above, I have essentially followed Chomsky (1992 et seq.) in my characterization of the beginning of the derivation. Thus, in (4) I proposed that the derivation begins by entering an unordered list of terms, which I called the initial terms, into a hypothetical workspace. The derivation proceeds by repeatedly deriving new terms via Merge (1). With each application of Merge, the operation adds exactly one new term to the list of terms involved in the derivation. But where does it end? For Chomsky, the answer has two parts, which we may translate reasonably straightforwardly into the framework at hand.

In the first place, the ouput must *include* all the pieces of the input. Chomsky (1992) states this in terms of phrase markers: "At any point, we may apply the operation SPELL-OUT, which switches to the PF component. If $\Sigma$ [the set of phrase markers in the derivation] is not a single phrase marker, the derivation crashes at PF, since PF rules cannot apply to a set of phrase markers and no legitimate PF representation $\pi$ is generated." (p. 30). This makes sense if the operation (1) is characterized as successively reducing the number of phrase markers, eventually down to one, as in section 1 of this paper. However, I have abandoned that assumption, claiming that it is an artifact of the notation. Instead, $\Sigma$ on the view proposed here is the lists, i.e., sets, of terms involved in the derivation. Of course, this increases rather than decreases in size with every application of Merge. Recall that every term may be translated trivially into a phrase-marker. We may then characterize SPELL-OUT as an operation which applies not to $\Sigma$, but rather to any term in the derivation. Any term is selected as the *output* of the derivation and fed to the PF (and perhaps LF) components. The requirement of inclusiveness holds not of the entire derivation (indeed, it would be hard to state it non-trivially for a list of terms) but of the output term, along the following lines

(22) *Inclusiveness*

> The output term which is fed to the interfaces must *include* all the terms in the list,
>
> where a term $\tau$ *includes* a term $\rho$ if $\rho$ is a term or sub-term of $\tau$.

For any list, i.e. for any stage in a derivation, there is maximally one term which will satisfy (22). At many points in the derivation there will be no such term. For instance, the lists in (17) or (9) have no term which meets the requirement.

There will, however, never be more than one term at any given stage (i.e. list) in a derivation. Thus the last terms in (10) or (12b) satisfy (22) but as no other term in either list includes the last term, the terms satisfying (22) are unique for any given list.

Note that (22) will also prevent instances of "movement across trees" which create terms that are not eventually re-Merged into the output term. For instance, the operation in (18) merging {Infl} and {eat} could take place after {Infl} has Merged with the entire VP. This would satisfy the hypothetical checking requirements between Infl and the verb. However, unless the resultant complex term {{Infl},{eat}} ($\gamma$ in (18)) serves as the input to Merge at some later stage, the derivation (list) will contain no term which satisfies (22). This serves to rule out many, but not all, classes of overapplication of Merge.

Thus, the Inclusiveness condition is satisfied by at most one term at any given stage of a derivation. However, this alone is not sufficient to define the unique output of a derivation. Compare (11) and (12b), repeated here as (23a-b), respectively:

(23)

a. {Infl}
{a}
{monster}
{eat}
{Fido}
{{eat},{Fido}}
{{a},{monster}}
{{{a},{monster}},{{eat},{Fido}}}
{{Infl},{{{a},{monster}},{{eat},{Fido}}}}

b. {Infl}
{a}
{monster}
{eat}
{Fido}
{{eat},{Fido}}
{{a},{monster}}
{{{a},{monster}},{{eat},{Fido}}}
{{Infl},{{{a},{monster}},{{eat},{Fido}}}}
{{{a},{monster}},{{Infl},{{{a},{monster}},{{eat},{Fido}}}}}

The list in (23b) is derived from that in (23a) by one step of the derivation, mechanical application of Merge to two terms in the list. However, (23a) and (23b) each contain a unique term which satisfies (22): the last term in each list. While (22) is thus a necessary condition for generating a legitimate output representation, it cannot be a sufficient condition.

Considering the example at hand, what is needed is some form of a feature-checking theory. Thus, the last term in (23a) is not the desired output since the subject has not raised. If the raising of the subject is required in order to satisfy the checking of some formal feature as characterized in Chomsky (1993 et seq), then we are in accord with Chomsky's characterization of the other output condition: the derivation must proceed until all relevant features are checked.

We now have a characterization of the output (i.e., SPELL-OUT, perhaps coextensive with the LF interface) which is not significantly different from Chomsky's:

(24) *The* output *of the derivation*

At any point in the derivation, select any term and feed it to the interfaces with the other computational components (LF,PF).

This term must
(i) meet (22), and
(ii) contain no unchecked features.[13]

Assuming moreover that feature checking is the sole motivation for syntactic operations, we may integrate Greed / Last Resort into the system. Here, we differ slightly from Chomsky. Since I have argued that there is no difference between Merge and Move, the requirement of Last Resort cannot be formulated as a condition on Move alone, but must be formulated on the syntactic operation Merge defined in (1). Thus:

(25) *Last Resort*[14]

Merge ($\alpha$,$\beta$) is a legitimate operation iff
Formal properties of either $\alpha$ or $\beta$ are thereby satisfied

The significant difference between this version and the version advocated by Chomsky is that the operation of Merge for me is subject to this requirement as much as is the operation Move (i.e., since the two are not distinct). Collins (this volume) considers a related approach by allowing satisfaction of (what amounts

---

[13] Alternatively, one could phrase (25) procedurally as:

(i) Once all features have been checked, feed any/the term meeting (24) to the interfaces.

[14] This definition is Lasnik's (1995) *Enlightened SeflfInterest*, or Collins's (this volume) *Greedier* in as much as it requires that either of the elements involved check features, whereas Chomsky's (1995) *Greed* requires that the element which does not project be the one with features checked. I do not intend to take a stand on this aspect of the condition here—the reader may substitute the Greed version for this one without affecting the points of this paper.

to) Inclusiveness (22) to count as a formal property for the purposes of (25). In Bobaljik (1995:chapter 5), I have argued that the difference between Merge and Move is irrelevant for the purposes of feature-checking theory, a view advocated in different, though related, ways, by Epstein (1995) and Groat (1995). I refer the reader to those works for discussion. It remains to be seen if this direction will be productive. The implication for the system developed above is clear, though. If (25) proves ultimately to hold only of Move and not of Merge operations, then the direction pursued in section 2 of this paper must be abandoned.

It remains to be demonstrated that most unwanted instances of Merge will violate some aspect of the feature-checking theory. In particular, the XP-movement analogue of (21) is likely not made use of, though it is not excluded proceduraly. There are many tacks which could be taken at this point. First, it is difficult to come up with cases of such "sideways" movement (i.e. to a non-c-commanding position) which will result in a derivation in which all appropriate features are checked.[15] Groat (1995) takes a different approach and spells out a theory of checking which would circumvent this problem, while being compatible with the present approach. I leave demonstration of this as a promisory note, but, as with the application of Last Resort to Merge, the

---

[15] Groat (1995) notes a potential candidate for such an operation: movement of an object wh-phrase to the specifier of a CP in subject position, from (i) to (ii), resulting in (iii) to mean, perhaps, (iv)):

(i) [ $_{IP}$ [ $_{CP}$ C$^{+wh}$ [ Sam ate ham ] ] [$_{VP}$ bothers **who** ] ]

(ii) [ $_{IP}$ [ $_{CP}$ **who** C$^{+wh}$ [ Sam ate ham ] ] [$_{VP}$ bothers ***trace*** ] ]

(iii) * Who Sam ate ham bothers?

(iv) Who does it bother that Sam ate ham.

This particular example is frought with other problems, in particular many hidden assumptions about the licensing of embedded +wh complementizers. A [wh] clause as the complement or subject of matrix *bother* can have only a sort of relative clause reading, and has no embedded question-like reading. Consider (v):

(v) When Sam ate ham bothers me.

This can only mean, I think, something like "I am bothered by the timing of Sam's eating ham" and pressupposes that I know the timing. Perhaps Sam ate ham only on Mondays, and this offends my sesnsibilities. What (v) cannot mean is that "I am worried about [trying to figure out] when Sam ate ham" the closest thing I can think of to an embedded question reading for the sentence.

I am not aware of other plausible candidates for sideways movement which do not also plausibly run into difficulty elsewhere, especially in the feature checking component.

implication is that if checking theory cannot account for the apparent ban on sideways movement then the approach of section 2 must be reconsidered.

A final important component of the system of constraints on syntactic operations is the local nature of such operations. In recent work, a specific aspect of this locality cocdition has been expressed as Shortest Move (Chomsky 1993) or Attract Closest (Chomsky, forthcoming). Essentially, these grow out of Rizzi's (1990) Relativized Minimality, in turn an extension of the Chain Condition (Rizzi 1986):

(26) The Chain Condition

CHAIN = $\{\alpha_1,...\alpha_n\}$ s.t. ...
- i. $\alpha_n$ c-commands $\alpha_{n+1}$
- ii. $\neg\exists\, \beta$ s.t. $\alpha_n$ c-commands $\beta$, $\beta$ c-commands $\alpha_{n+1}$ where $\beta$ is of some relevant type (e.g., same features as $\alpha$) [i.e. Relativized Minimality]

The Chain Condition is a condition on representations. It can be expressed as a well-formedness condition on chains created by syntactic operations (movement) or as a condition on the output of discrete chain-formation algorithms. For instance, we could define a *chain* as a sequence of copies in the output term. If Epstein (1995) and Groat (1995) are correct, then c-command itself may be defined as the relations between terms, i.e. in terms of Merge and phrase markers need not enter the computation at all. The chain condition could then hold of this definition, and would serve to filter out unwanted concatenations, i.e. derivations.

While an output filter is perhaps the most straightforward way to proceed given the discussion above, it is not the most appealing within the present framework, since a major conceptual force behind the programme is that derivations should be evaluated locally, i.e., at each step, and not in terms of what may or may not happen at the output (see Collins, this volume, Ura, this volume, and Bobaljik, 1995 for some discussion of this approach). Hence the motivation for reformulation of (26) effectively as conditions on movement, such as Shortest Move.

These formulations are generally defined in terms of "distance" in phrase structure, and generally in terms of "depth" from the root node. If the discussion of section 2 is on the right track, then phrase markers are representations not of stages of the derivation but rather of the individual terms which make up the derivation. It makes sense, then, to define chains internal to each term. That is, as each step of the derivation defines a new term, the relationships among the sub-terms are defined (see Epstein, 1995, and Groat, 1995). In particular, the relationship of copies one to another is evaluated in terms of the chain condition. In this way, the condition can be seen as a local condition, defined on each stage of the derivation. Closest, then, has its usual meaning as an economy condition: given a stage (S) of the derivation, there are a finite number of possible sets of input terms for Merge. Which particular set will be chosen is decided by economy considerations, including an evaluation of which set involves the formation of the shortest chain. A fuller working out of the details of this definition awaits further research, though it should be clear from the preceding discussion that this is principally a matter of definition. On this point, the system developed here does not differ significantly from closely related approaches within the general framework assumed thoroughout.

## 4. Conclusion

In this paper, I have demonstrated that certain redundancies or disjunctions internal to specific versions of the theory extending from Chomsky (1993 et seq) are artifactual and result from the constraints imposed on the system by the notational conventions of phrase markers. While phrase markers are adequate representations of individual terms of a derivation, they do not represent individual stages of the derivation. The operation Merge operates on the set of terms which characterizes a given stage in the derivation, not on individual terms themselves. Merge is a recursive procedure, which derives a new term from the set of previously defined terms, in this way mapping one stage of the derivation to the next. It does not map one term (or phrase marker) to another.

By focusing on the sets of terms in the derivaiton, I have shown that there is no need to appeal to a distinct operation Copy to relate one stage of the derivation to the next, or even to define complex terms. Even though the output term (or any complex term) may contain multiple copies of other terms, the effect of copying falls out from simple mechanical application of (1). Likewise, the apparent exceptional nature of head-movement with respect to the Extension Requirement is seen to be a result of the notational conventions associated with phrase markers. Viewing the derivation in terms of sets of terms, there is no reason to suppose that movement must occur internal to a single phrase marker. Indeed, such a requirement would be highly stipulative. Applying (1) mechanically to unordered lists of terms, we find that we derive the effects of head-movement in a manner proceduraly identical to XP-movement and Merge. The differences obtain only when the procedure is expressed shorthand in the notation of phrase markers: it appears that head movement involves movement across phrase markers, while XP-substitution involves movement internal to a given tree.

This paper has only begun to scratch the surface of the issues involved. In the final section, I touched briefly on a few implications of the view of the syntax advocated here. There remain, of course, a host of unanswered questions, and many unbroached topics. Sadly, at the present time these must be swept under the rug of future work.

## References

Bobaljik, Jonathan David (1995). *Morphosyntax: The syntax of verbal inflection*. Ph.D. dissertation, MIT. Distributed by MITWPL.

Brody, Michael (1995). *Lexico-logical form*. Cambridge, MA: MIT Press.

Chomsky, Noam (1992). A minimalist program for linguistic theory. *MIT occasional papers in linguistics* #1. [Published (1993) in Kenneth Hale & S. Jay Keyser (eds.), *The view from Building 20: Essays in linguistics in honor of Sylvain Bromberger*, Cambridge, MA: MIT Press, 1-52.]

——— (1995). Bare phrase structure. In Gert Webelhuth (ed.), *Government and Binding theory and the Minimalist program*, Oxford/Cambridge, MA: Basil Blackwell, 385-439. [Previously distributed (1994) as *MIT occasional papers in linguistics* #5.]

Collins, Chris (this volume). Toward a theory of optimal derivations.

Epstein, Samuel David (1995). Un-principled syntax and the derivation of syntactic relations. Ms., Harvard.

Groat, Erich (1995). On the redundancy of syntactic representations. Ms., Harvard. [Presented at GLOW 18, Tromsø, June 1995].

——— & John O'Neil (1994). Spell-Out at the LF interface. To appear in Samuel David Epstein, Höskuldur Thráinsson, Werner Abraham & C. Jan-Wouter Zwart (eds.), *Minimal ideas*, Amsterdam/Philadelphia: John Benjamin.

Kayne, Richard (1995). *The antisymmetry of syntax*. Cambridge, MA: MIT Press.

Kitagawa, Yoshihisa (1986). *Subjects in Japanese and English*. Ph.D. dissertation, University of Massachusetts, Amherst.

Kitahara, Hisatsugu (1995). Target α: Deducing strict cyclicity from derivational economy. *Linguistic Inquiry* 26(1), 47-78.

Lasnik, Howard (1995). Last resort. To appear in Shosuke Haraguchi and Michio Funaki (eds.), *Proceedings of the first Numazu conference on formal linguistics*.

Pesetsky, David (in preparation). *Optimality principles of sentence pronunciation*. Ms., MIT.

Rizzi, Luigi (1986). On chain formation. In *Syntax and Semantics 19: The syntax of pronominal clitics*, 65-95.

——— (1990). *Relativized Minimality*, Cambridge, MA: MIT Press.

Ura, Hiroyuki (this volume). Towards a theory of a "strictly derivational" economy condition.

Watanabe, Akira (this volume). The conceptual basis of cyclicity.

Harvard University
Society of Fellows
78 Mt. Auburn St.
Cambridge MA 02138 USA

*bobaljik@mit.edu*