To appear in the Proceedings of *Interphases*, Kleanthes K. Grohmann (ed.)

Surviving Reconstruction[♣]

Thomas S. Stroik
University of Missouri-Kansas City
stroikt@umkc.edu

Michael T. Putnam
University of Michigan
mtputnam@umich.edu

1 INTRODUCTION

Constructions with Condition C asymmetries (see (1) and (2)) pose the problem of explaining why some, but not all, the DPs contained in the bracketed wh-phrases can be co-referential with the pronoun **he**.

(1) [which picture of Bill that John likes] did he buy? *Bill…he/Ok John…he

(2) a he likes [everything that John writes] *John…he

b [everything that John writes] he likes Ok John…he

To account for asymmetries such as those in (1) and (2), Lebeaux (1991, 1995), Chomsky (1993), Epstein et al (1997), Rubin (2003), and Fox (2003, 2004) propose two different applications of the Merge operation—one that applies cyclically and one that applies non-cyclically (i.e., after the application of Move or Remerge operation). Besides positing two applicational platforms of Merge, these theorists also stipulate that all arguments must be merged cyclically, while adjuncts can be merged cyclically or non-cyclically. These assumptions, together with Chomsky's (1993) copy theory of movement, will generate the following derivation for (1).

(3) a he did buy [which picture of Bill] – wh-movement →

b [which picture of Bill] did he buy [which picture of Bill] – adjunct Merge→

c [which picture of Bill [that John likes]] did he buy [which picture of Bill]

Importantly, this derivation allows us to explain the co-referential relations in (1): the DP **Bill** in (3c) cannot be co-referential with the pronoun **he** because the pronoun c-commands the most embedded DP copy *Bill*, whereas the DP **John** can be co-referential with the pronoun because the late merger of the relative clause prevents the pronoun from ever c-commanding the DP. This sort of analysis can account for the data in (1) and (2),

[♣] Portions of this paper have been presented at LASSO 2005, DEAL 2005 and the Syntax Support Group Colloquy Series at the University of Michigan. In connection with these presentations, this paper has benefited from comments and discussions with Sam Epstein, Kleanthes Grohmann, Dennis Ott, David Pesetsky and Peter Sells. All remaining shortcomings and errors remain our own.

but it comes at a cost. In particular, having non-cyclical operations dramatically complicates processing since such operations force derivations to return to, and re-compute, structures already built (cf. Frampton and Gutmann 2002). In an attempt to simplify processing, Chomsky (2001) eliminates all (expensive) non-cyclical Merge operations. However, purging non-cyclical Merge leaves Chomsky with the problem of how to account for the reconstruction asymmetries in (1). To solve this problem, Chomsky proposes that the narrow syntax must have two separate, cyclical Merge operations: argument-Merge (Set Merge) and adjunct-Merge (Pair Merge). Although these two Merge operations are both cyclical, they differ in their structural properties—the output of the adjunct-Merge operation is not structurally visible (or "simple") until it is converted to a Set Merge output via an operation called Simpl. Hence, for Chomsky, the relative clause in (1) is merged into the derivation, as in (4a), but it remains invisible until the derivation reaches (4b). (In (4) the italicized clause is syntactically invisible.)

(4) a he did buy [which picture of Bill *that John likes*] – wh-movement →

b [which picture of Bill *that John likes*] did he buy [which picture of Bill *that John likes*] – Simple-operation →

c [which picture of Bill that John likes] did he buy [which picture of Bill *that John likes*]

It is the structural invisibility of the most embedded relative clause in (4a) and (4b) that keeps the DP *John* from being c-commanded by the pronoun. Though Chomsky explains the coreferential relations in (1), we should note the fact that to account for these coreferential data, without appealing to non-cyclical Merge operations, Chomsky must complicate his system of operations, in a rather bizarre way, by adding a non-simple adjunct operation (Pair Merge) and a Simp operation that later undoes (makes simple) the output of the adjunct operation.

In this paper, we offer a re-analysis of the reconstruction data in (1) and (2) that avoids both non-cyclical operations and Chomsky's undoing operations. We propose, in particular, that the Condition C asymmetries in (1) and (2) can be explained within Stroik's (1999, forthcoming) version of minimalism (called *Survive*). According to Stroik, an optimal minimalist syntax will have only strictly local Merge operations that map elements from the Numeration N onto the Derivation D. To maximally simplify processing, these operations will not have look-back or look-forward properties (this criterion rules out Attract and Agree operations, economy conditions, and Internal Merge operations). For Stroik, there are only two such types of syntactic operations: Merge, which concatenates, in D, two elements from N (what actually gets merged are copies of elements of N—the originals of which continue to be contained in N); and Remerge, which remerges into D syntactic objects SOs in N that have already been merged but still have concatenative features that must be checked (these features have "survived" previous applications of *Merge*). An abbreviated Survive-derivation can be seen in (5) (note that lower copies of constituents are marked in boldface type).

(5) a Merge {hired, Sam} → hired Sam

b Survive {Sam}

b Merge {was, {hired, **Sam**}} → was hired **Sam**

c Remerge {Sam, {was, {hired, **Sam**}}} → Sam was hired **Sam**

The DP **Sam**, which is merged with the verb **hired** in (5a), must remerge, from N, later in the derivation (see (5d)); this is necessary to ensure that its Case and agreement features, which have survived the verb-merge, are appropriately checked. It is worth emphasizing here that (5d) has been derived without any non-cyclical operations and without any Internal Merge operations.

As we will show in this paper, we can give derivations for (2a) and (2b), similar to the one in (5), if we adopt a version of Fox's (2004) late adjunct Merge—i.e., by allowing adjuncts to merge with an SO prior to Remerge (such a merger should be permissible because the SO is in the Numeration, not in the Derivation; hence the merger will not be non-cyclical). Under this version of late Merge, (2b) could be derived as follows:

(6) a Merge {likes, everything} → likes everything

b Survive {everything}

c Merge {he, {likes, **everything**} → he likes **everything**

d Merge {everything, {John writes}} → everything John writes

e Remerge {everything John writes, {he, {likes, everything}}} → everything

John writes he likes **everything**

Given that the pronoun in (6a-e) never c-commands the DP **John**, it is possible for the DP and the pronoun to be coreferential. In this paper, we will provide equivalent derivations, and explanations, for the coreferential relations in (1) and in (2a).

2 ON SYNTACTIC OPERATIONS

The ultimate goal of this paper is to investigate the sorts of syntactic operations that are conceptually necessary to account for the reconstructions effects in constructions such as those in (1) and (2). According to Chomsky (1993, 1995, 2001), the computational system of Human Language (HL) requires two structure-building operations—External Merge and Internal Merge (a variant of the Move operation). External Merge (EM) is a mapping from the Numeration (N)—a lexical array of items taken from the lexicon—to the syntactic Derivation (D), as stated in (7).

(7) EM: N $\rightarrow$ D

This operation builds syntactic structure from lexical input: it takes a lexical item and concatentates it in D with another lexical item or a syntactic object (as already formed in the derivation D). That is, for lexical item $\alpha \in$ N and for $\beta \in$ N or D, EM syntactically connects α and β (see (8)).

(8) EM $\{\alpha,\beta\} \rightarrow \alpha\beta$

It is the EM operation that will derive the phrase the syntactic object **that child** from the lexical items **that** and **child**: EM {that, child} $\rightarrow$ that child.

Some version of the EM operation is conceptually necessary if we are to account for Frege's (1884) observations about the compositional (concatenative) nature of semantics. However, the simple version of the EM operation formulated in (8) is too strong for HL. As given in (8), EM would not place any constraints on the ability of the computational system to introduce lexical items into any derivation. In other words, this EM would allow any, and every, concatenation. The EM operation could, for example, concatenate a Determiner with a Determiner Phrase (DP), as in (9), producing an ungrammatical syntactic object.

(9) EM {the, {this, child}} $\rightarrow$ the this child

That the EM operation in (8) overgenerates permissible syntactic objects in a Language suggests that this operation cannot be freely concatenative; rather, it must impose constraints on licit concatenations.[1]

Adger (2003) and many others have reduced the expressive power of EM (his Merge) by positing this operation not as a concatenating operation, but as a linking operation (Putnam (2006b) formalizes this as *Link!*). Noting that lexical items, and the

[1] Collins (1997) also proposes a similar constraint on the primitive operation *Merge* for similar reasons.

syntactic objects formed from them, consist of sets of phonological, morphological, syntactic, and semantic features F, Adger proposes that lexical item α will merge with syntactic object β if and only if α,β share, and check, features—this is stated in (10).

(10) EM {α<F>, β<F>} → αβ

Under this analysis of Merge, the basic structure-building operation in HL compiles syntactic structure by linking together lexical items with matching syntactic features. We can see this in (11a) and (11b). In (11a), the verb **give** has object features for both a nominal <N> object and a prepositional <P> object; hence the verb can merge with both the nominal constituent **that book** and the propositional constituent **to Pat**. On the other hand, in (11b), the verb **admire** has an object <N> feature, but not a <P> feature; therefore this verb can merge with **that book**, but not with **to Pat**.

(11) a Chris gave that book to Pat

b Chris admired that book (*to Pat)

Relatedly, the determiner **that** has a singular Number feature <SING>, so it will not be able to merge with any nouns that have an un-matching plural Number feature <PL>, as in the determiner phrase in (12).

(12) *that books

By placing a feature-matching constraint on concatenation, EM (10) radically reduces the generative capacity of the merge operation and it makes syntactic structure a linked structure and not merely a concatenated structure.

Chomsky (1965, 1980) has long argued that syntactic structures are not only built—by phrase structure rules or by the Merge operation—but they are also modified (or transformed). Motivation for "transformed" structures comes from the data in (13) and (14), where the bracketed DP object of the verb **like** in (13a) can appear in a modified position in (13b) and where the wh-constituent in (14), which is merged as the object of the verb **elected**, must be transformed into the subject of the embedded sentence and must also be placed into the fronted interrogative position in the matrix sentence.

(13) a I like [that book]

b [that book] I like

(14) Which politicians does Chris think will be elected.

To explain how a constituent can appear in a syntactic position distal from its merged position, Chomsky (2001) claims there is a second syntactic operation, called Internal Merge (IM), which can re-locate constituents within a Derivation. This operation differs from EM in one essential way—it builds structure not by linking elements from N into D, but by re-positioning elements already in D somewhere else in D. That is, IM is a D-to-D mapping, as stated in (15).

(15) IM: D → D

What IM does, in particular, is take an element a ε D and remerge it in D. So, if D consists of [d [ba]], then IM can select the element a and remerge it (see (16)).

(16) IM {a, [d[ba]]} → [a[d [b **a**]]] where **a** is a phonetically inert copy of a.

As with our first formulation of EM (8), IM (16) is much too powerful. It would allow any element in D to be remerged, thereby permitting the verb **have** in (17a) to remerge as in (17b).

(17) a Pat shouldn't have left → IM

b *Have Pat shouldn't **have** left.

The fact that (17b) is ungrammatical demonstrates that IM (16) overgenerates syntactic structure and that IM must reduce its derivational capacity. Chomsky (2001) lessens the computational power of IM in two ways. First, rather than letting any element remerge at any point in the derivation, Chomsky limits remerge to cases involving feature-match. For remerge to occur, a head (say d) must have a concatenative feature <F> that must be linked to a paired feature <F> on another syntactic constituent. The head d will "probe" the derivation D searching for a constituent with an active <F> feature, i.e., a feature not previously checked for feature-match by another head. If constituent a has such an <F> feature, then the constituent will be able to be remerged, in accordance with IM (18).

(18) IM {a<F>, [d<F> [b a<F>]]} → [a [d [b**a**]]]

Notice that both EM and IM build syntactic structure by locally linking constituents with matching features.

The second way Chomsky constrains IM is by delimiting the search that d<F> can undertake to find a constituent with a counterpart <F> feature. Chomsky seems keenly aware that having unbounded searches (searches through an entire derivation D) would greatly increase the processing burden on the computational system. So, he proposes that these searches be top-down searches that are domain-restricted within D by phase boundaries (where a phase is a semantically complete phrase—a CP or a *v*P). A head d<F> can look down through the derivation D until it encounters a phase boundary; at this point, it can still look at the Spec of this phase. However, it can't look any deeper in D. Should d<F> not find any constituent with an active <F> feature in its search-restricted domain, it must abort its search. Simplified even further, the search will also terminate once its goal—a constituent with an active <F> feature--is found. The search will not continue to look for subsequent goals.

To see how these operations are implemented, consider the derivation in (19). We will pick up the derivation once it has merged its C<Q, WH> head, which has both a <Q> feature and a <WH>.

(19) a [C<Q, WH> [Pat will<Q> read [what<WH> to whom<WH>]]] → IM

b [will<Q> [C<Q, WH> [Pat **will** read [what<WH> to whom<WH>]]]] → IM

c [what<WH> [will<Q> [C<Q, WH> [Pat **will** read [**what** to whom<WH>]]]]]

'What will Pat read to whom?'

In (19a), the C<Q> head will look for a constituent with a <Q> feature within its search domain. Once it locates the constituent will<Q>, the search will terminate and IM will apply, remerging will<Q> in the matrix CP. Since the C head also has a <WH> feature, the head must begin another search—one that looks for a <WH> counterpart feature. The search will terminate when the first active <WH> is found on the constituent what<WH>. IM will apply again, this time remerging what<WH> in the matrix CP.

Despite the seeming ease with which (19c) is derived, this derivation is actually extremely complex. If we look closely at how the derivation proceeds from (19b) to (19c), we will see some of this complexity. Of note here is the fact that IM can't apply directly to (19b). Rather, before IM can apply to (19b), the C<WH> head must initiate a search for a constituent with a <WH> feature—that is, a FIND<WH> operation must apply to D. The FIND operation itself is a complex (composite) operation that must LOOK-AT-FEATURES of every constituent K in the Derivation, that must DISREAGARD all <F> features of K that are not a <WH> feature, and that must GO-TO-NEXT constituent should a <WH> feature not be found on a given K; and of course, all these operation will have to apply to each successive K that is searched. Furthermore, should the search successfully FIND<WH> in D (the what<WH> constituent), then the search ends, but another set of operations must now apply. These operations include a MAKE-COPY operation, an operation that, according to Chomsky (2001), will create a copy without phonetic features[2], and the REMOVE-ORIGINAL and INSERT-COPY operations, which will remove the constituent located by FIND and replace it with the

2 This is a rather problematic operation in that it is unclear where this operation could take place.

copy formed by the non-phonetically realized MAKE-COPY operation. The removed constituent can now undergo IM with D, deriving (17c) from (17b) after the application of all the intermediate operations identified above.[3]

Needless to say, using IM to build syntactic structure is extremely costly, in processing terms, because of all the ancillary operations required to license the IM operation.[4] For derivations that must apply IM several times (as would be the case for (14)), the expensive processing needs of IM are all the more daunting.

(14) Which politicians does Chris think will be elected.

The computational complexity surrounding the IM operation raises significant questions – similar to those raised by Brody (1998, 2002) and Stroik (forthcoming) – about the conceptual necessity of this operation.

3 INTERNAL MERGE AND RECONSTRUCTION EFFECTS

In this section, we look all the more closely at the IM operation by considering how an IM analysis can account for the Reconstructions Effects in (20).

(20) a [which picture of Bill that John likes] did he buy? *Bill…he/OK John…he

b He bought [a picture of Bill that John likes]. *he…Bill/*he…John

What needs to be explained in (20) is why the coreferentiality relationship between the pronoun **he** and the DP **Bill** is not affected by whether the bracketed DP undergoes displacement (as in (20a)) or not (as in (20b)), while the coreferentiality relationship between the pronoun and the DP **John** is affected by the displacement of the bracketed DP. We need to determine what role, if any, IM plays in these differing coreferentiality relationships.

The data in (20) led Chomsky (1993) to conclude that arguments contained within displaced constituents, such as the DP **Bill** in (20a), behave, in terms of Principle C relations, as if they are in their pre-displacement positions (see (20b)). In other words, the constituent **[which picture of Bill]** in (20a) acts as if it shows up twice in a syntactic derivation—once in its displaced (internally merged) position and once, in reconstructed copy form, in its (externally) merged position, as in (21).

---

3 Another problem with current minimalism is that in the derivation illustrated in (19) the C-head possessing the [Q]-feature that will peer down into its c-command domain to find a relevant goal is previously blind to the wh-items that have existed in the derivation for quite some time. That these wh-items are not found and recognized until the very end of the derivation is problematic for a true minimalist analysis of syntax.

4 For example, Internal Merge requires ancillary operations such as FIND PHASE EDGE that burden the processing load of the language faculty.

(21) [which picture of Bill that John likes] did he buy **[which picture of Bill]**

For Chomsky (1993), Lebeaux (1995) Epstein et al (1997), Fox (2003), and many others, the data in (20) provide important support for some version of a copy theory of movement (variants of Internal Merge).

This above analysis, however, does not hold for adjuncts contained in displaced constituents. As we can see in (20a), the adjunct (the relative clause) contained within the displaced constituent does not appear to show up twice syntactically. If it did show up in the same places that the argument DP **Bill** does, then we would expect that (20a) would have (22) as its syntactic representation at some point in the derivation.

(22) [which picture of Bill that John likes] did he buy [**which picture of Bill that John likes]**

But (22) is not a possible representation for (20a) because in (22) the pronoun **he** c-commands (and should not be co-referential with) the adjunct contained DP **John**. The fact that the pronoun and the DP can be co-referential argues against (22). What this means is that although the adjunct can show up once the bracketed constituent is displaced in (20a), it cannot also show up syntactically in the pre-displaced bracketed constituent.

Within IM-style analyses, there are only two ways of explaining why the adjunct in (20a) shows up syntactically after the wh-constituent has been displaced. Either the adjunct is merged into the derivation after the wh-displacement (the Late Merge Hypothesis) or the adjunct is merged into the derivation prior to the displacement, but becomes syntactically visible only after the displacement (the make Simple Hypothesis). Late Merge analyses of data such as (20a) have been advanced by Lebeaux (1991), Chomsky (1993), Fox (2003), and others. Under the Late Merge analysis, (20a) is derived as follows:

(23) a he did buy [which picture of Bill] -- wh-movement →

b [which picture of Bill] did he buy **[which picture of Bill]** – adjunct Merge →

c [which picture of Bill [that John likes]] did he buy **[which picture of Bill]**

Even though this derivation expresses all the structural constraints on co-referential relations between the pronoun **he** and both the DP **Bill** and the DP **John**, it is an untenable analysis. Behind the apparent simplicity of (23) lurks a significant problem. As Chomsky (2001) notes, no Merge operation should "tuck" elements into a derivation D because of the processing complications that arise when structure is re-modeled from within (the structure would have to be, in essence, dismantled and then rebuilt). Merge operations, according to Chomsky, must be cyclic operations that build structure at the edges of D, not within D. If we look at derivation (23), we will see that the Late Merge of the adjunct is a case of a non-cyclic Merging of the relative clause into the wh-constituent; therefore this Merge should be disallowed. It might be possible, however, to circumvent the "tucking in" problem by allowing the IM operation to re-situate the wh-constituent in (23a) in some work-space outside the derivation (where it could Merge the adjunct) prior to remerging the constituent back into D; unfortunately, doing this would add several complexities to the IM operation beyond those we discussed in the previous section. These complexities would include changing IM from a D-to-D mapping to a D-to-Workspace mapping and then requiring a subsequent operation that could remerge elements from the Workspace to D, among other complexities. At this point, the IM operation loses any semblance of simplicity and, in fact, loses any claim to being an operation that involves the internal merge of constituents.

It would seem, then, that if we are to have the IM operation in our Narrow Syntax, we must follow Chomsky in assuming that the adjunct in (20a) is not merged late into the derivation; rather, it is merged in its externally Merged position shown in (20b), though it is Merged differently than an argument would be. The adjunct is Pair Merged into D, which means, essentially, that it is merged, but it is not syntactically visible; and it will remain syntactically invisible and inert until the Simpl operation makes the adjunct syntactically present. Under these assumptions, (20a) will have partial derivation (24). (Recall that the italicized constituent is syntactically invisible.)

(24) a he did buy [which picture of Bill *[that John likes]*] –wh-movement →

b [which picture of Bill *[that John likes]*] did he buy **[which picture of Bill** *[that John likes]***]** – Simpl →

c [which picture of bill [that John likes]] did he buy **[which picture of Bill** *[that John likes]***]**

Attractive as this analysis may be for the data in (20), it does not hold up if we look at some other adjunct constructions. Consider the data in (25).

(25) a [after Pat wakes up] I want her to leave

b I want her to leave *[after Pat wakes up]*

In (25a), the bracketed constituent is a temporal adjunct that modifies the embedded verb **leave** and, under Chomsky's analysis of adjuncts, this adjunct will have to be externally

Merged in its verb-modifying position, as in (25b). Since the adjunct is Paired Merged in the embedded VP, it will be syntactically invisible there. But now we face a quandary. We need to move the adjunct from its embedded position in (25b) into its displaced position in the matrix sentence (as in (25a)). However, this adjunct can't get a free ride the way the adjunct in (20a) does. That is, the adjunct in (20a) is contained within a wh-constituent that gets displaced, so the adjunct gets displaced as a by-product of the wh-movement. Since the adjunct in (25b) can't get a free ride, it will have to move on its own accord. This requires, however, that the adjunct be syntactically visible for the IM operation. In other words, before the adjunct can be moved, it will have to undergo Simpl. The derivation for (25a) will have to proceed as in (26).

(26) a I want her to leave *[after Pat wakes up]* – Simpl →

b I want her to leave [after Pat wakes up] -- IM →

c [after Pat wakes up] I want her to leave **[after Pat wakes up]**

This derivation leads us to a problematic conclusion; that is, given derivation (26), it should be impossible to have co-referential relations between the pronoun **her** and the DP **Pat** because the pronoun c-commands the DP in structures where the adjunct is syntactically visible--(26b, c). The fact that the pronoun and the DP can be co-referential suggests that the Simpl-based derivation (26) is not a viable derivation for (25a).

Of the two IM analyses we've considered for adjunct-displacement constructions, neither of them is feasible. Since it seems at present that there are only two possible IM analyses for these constructions, we are left as were at the end of the last section with the nagging sense that IM (or any movement-related operation) lacks both conceptual and empirical motivation.

4 SURVIVING RECONSTRUCTION

After exposing the conceptual faults that we encountered with the two IM-analyses in accounting for adjunct-displacement constructions and the potential added burden that they add to the computational system with regards to language processing, in this section we will present an alternative analysis along the lines of Stroik (1999, forthcoming) and Stroik and Putnam (in progress). Under this view, we interpret the displacement of syntactic objects from their based position not necessitated by *Attract* or *Move*, but enacted by means of survival. Stroik defines this grammatical primitive as the *Survive Principle*:

(27) *The Revised Survive Principle* (based on Stroik 1999:286)
If Y is a syntactic object (SO) in an XP headed by X, and Y has an unchecked feature [+F] which is incompatible with the feature X, Y remains active in the Numeration.

To provide an illustration of the *Survive Principle* in action, consider the following sentence in (28) with its derivational history following in (29) (data taken from Stroik (forthcoming): 79-80).

(28) Who snores?

(29) a. Merge {who, snores} → who snores

b Survive {who}

c. Merge {T, {**who**, snores}} → T **who** snores

d. Remerge {who, {T, {**who**, snores}}} → who T **who** snores

e. Survive {who}

f. Merge {C, {**who**, {T, {**who**, snores}}}} → C **who** T **who** snores

g. Remerge {who, {C, {**who**, {T, {**who**, snores}}}}} → who C **who** T **who** snores

Upon the concatenation of a syntactic object with a head both bearing the matching feature δ through the operation *Merge* the syntactic object will survive and remain active in the lexicon if the syntactic object bears any additional features not present on the immediately governing head. In the derivation above the wh-item **who** will be mapped into the *v*P to check its θ-feature. At this point in the derivation a link is established signaling to the external interfaces (e.g. LF, PF) the thematic identity associated with this concatenate structure (cf. Putnam 2006b). Immediately after the concatenation of <who, snores> (29a) **who** survives from this position due to the additional features it possesses that must be properly licensed through iterative applications of *Merge* and *Remerge* in the course of the derivation. In steps (29d) and (29g) **who** remerges from the lexicon in order to properly discharge its agreement and Q-features. Perhaps the term 'discharge' is a bit of a misnomer, because the true motivation behind the sequence of *Merge-Survive-Remerge* is to generate concatenate structures that are interface interpretable. As explained early in **Section 2** of this paper, the mapping of copies of lexical items into the narrow syntax rather than the objects themselves eliminates the need for Copy Theory and "movement" *a priori* from the theory, thus providing a purely derivational account of syntactic operations rather than a view of mixed theory that is weakly representational (cf. Brody 1998, 2002).

The iterative application of *Merge-Survive-Remerge* also provides a straight-forward account to long-distance wh-movement previously unattainable in minimalism.

(30) What$_i$ do [$_{TP}$ t_i you think [$_{CP}$ t_i John likes t_i]]?

(31) *What$_i$ do [$_{TP}$ t_i you think [$_{CP}$ that John likes t_i]]?

Sentence similar to (30) and (31) serve as canonical examples in the generative tradition to illustrate the reality of cyclic movement.[5] In (30) the wh-item **what** must move to the left periphery of the embedded CP, TP and then to its final destination in the matrix CP. Example (31) shows that **what** must strictly adhere to cyclic movement or else the system will ultimately crash. Any theory of syntax employing either a *Move* or *Attract* model of constituent construal must delay the final feature-evaluation and subsequent checking or valuing process until the final C enters the derivation. Furthermore, successive cyclicity is an unsubstantiated formative in these models, i.e. it is a necessary component of the theory although we have little if any proof why it exists. Our version of XP-displacement under the *Merge-Survive-Remerge* mechanism forces the evaluation of the feature identity of all lexical items upon the merger of every head into the narrow syntax. In example (30), after concatenating with V, the wh-item **what** immediately survives (due to its remaining [Q] feature) and remains active in the lexicon for further operations. This syntactic object is an eligible candidate to remerge into the syntax at any time; however, it can only do so when a head with a matching feature appears. Upon every application of head merger an evaluation process takes place within the computation system.[6]

Returning to the focus of this paper, the remainder of this section will illustrate the conceptual advantages our approach has in properly deriving reconstruction structures in avoiding the aforementioned pitfalls of IM-analyses. First, let's return to example (20) from the previous section.

(20) a [which picture of Bill that John likes] did he buy? *Bill…he/OK John…he

b He bought [a picture of Bill that John likes]. *he…Bill/*he…John

[5] Bear in mind that the derivational histories in (30) and (31) represent a movement-based analysis akin to former instantiations of generative theory. The proposal put forward in this paper does not support the theoretical approach that constituent displacement takes place by means of *Move* but rather *Survive*.

[6] David Pesetsky (personal communication) points out that the application of evaluation processes upon every iterative head merger could also potentially be very costly from a processing standpoint. Be that as it may, it is far more economical to envision a system which immediately evaluations candidates rather than one that makes use of look-ahead and look-behind operations.

Both the application of Late Merge and the Simpl operation are untenable options in explaining these Condition C inconsistencies. Late Merge (as currently formulated) requires the "tucking in" of the adjunct [that John likes] into the wh-items which in itself is an undesirable result, while the "peek-a-boo" effects of Simpl cloak the adjunct through some of the derivation and make it visible for syntactic operations and effects later on. Although it is an attractive alternative solution for (20), it does not hold up when we consider other adjunct constructions such as (25).

(25) a [after Pat wakes up] I want her to leave

b I want her to leave [after Pat wakes up]

The crux of the matter is determining how and when the adjunct [that John likes] enters the syntax. Fox's (2004) current version of Late Merge faces the unwanted "tucking in" problem since it is in the Derivation rather than in the Numeration. In a *Survive*-based model of syntactic derivation, we can avoid tucking the adjunct (qua *Merge*) to the complex wh-item by arguing that the adjunct resides in the Numeration and adjoins to the wh-item [what picture of Bill] prior to its remerging into the syntax since it survives and returns back to the lexicon due to its [Q]-features which must be checked in CP. Call this operation *Late Num Merge*.[7] Two points must be clarified at this point to understand the conceptual advantages of our approach to a minimalist, derivational approach to generate syntax: First, the adjunction of [that John likes] is a syntactic object in the Numeration, therefore its concatenation with [what picture of Bill] will not be non-cyclic and therefore does not fall victim to "tucking in". Second, the cyclic application of our reformulation of Late Merge forces the adjunct [that John likes] to be visible in the syntax for all operations. This fact allows us to abandon the now unnecessary Simpl operation on the grounds of virtual conceptual necessity. Since the DP **John** was not a part of the original complex wh-item [what picture of Bill] that merged into the VP prior to its repulsion there is no point in the derivation during which the pronoun **he** could potential c-command **John**, thus explaining how **John** and **he** can be co-referential in (20a). The derivational history in (32) below highlights the pivotal steps in the composition of (20a).

(32) a Merge {buy, [which picture of Bill] } → buy which picture of Bill

b Survive [which picture of Bill] ([Q]-feature) →

c Merge {he, buy **[which pictures of Bill]** } → he buy **which picture of Bill**

[7] We take an agnostic view at this time as to whether adjunction is motivated by some sort of feature or feature-like entity active in the Numeration or Derivation (also see Putnam 2006a:Ch. 4 and Rubin 2003).

d Merge {did, {he, {T, {buy **[which picture of Bill]** }}} → did he buy **which picture of Bill**

e Merge {C , {did, {he, {T, {buy **[which picture of Bill]** }}}} → C did he T buy **which picture of Bill**

f **Late Num Merge** { [which pictures of Bill] , [that John likes] }

g Remerge { [which pictures of Bill that John likes] , {C , {did, {he, {T, { buy **[which picture of Bill]** }}}}} → which pictures of Bill that John likes C did he T buy **which picture of Bill**

'Which pictures of Bill that John likes did he buy?'

The non-cyclic application of *Late Num Merge* (32f) in the Numeration rather than in the course of the Derivation provides a straightforward explanation of Condition C asymmetries within core minimalist desiderata.

5 THEORETICAL CONSEQUENCES

The removal of Internal Merge (Move) would undoubtedly have wide-sweeping effects on the entire generative enterprise. A closer look at these changes shows that they would be a welcome adjustment to the minimalist program. First, by dismissing Internal Merge in favor of a Survive approach, the theory needn't construe and enforce economy constraints upon the language faculty anymore; the fact that local evaluation processes take place at ever step of Merge and Remerge throughout the course of the derivation mitigates the necessity of the existence of such constraints on the grammar. Economy is a natural by-product of the *Survive Principle*. Second, Chomsky's formulation of phases (*v*P and CP) and multiple Spell-Out (cf. Uriagareka 1999 and a host of others) can be removed from the system. In current minimalist models of syntax supporting either a *Move* or *Attract* stance on XP-construal, not only must some version of look-ahead or look-back feature evaluation exist, but some sort of evaluating property that recognizes the shape of phases must also be a component of the theory. In a minimalist syntax that is argued to be label-free in the spirit of Collins (2002), it is conceptually puzzling and taxing from a processing standpoint why the human language faculty should/must be responsible for both feature evaluation and the recognition of larger units such as phases. A derivational model based on the *Survive Principle* has the distinct advantage of destroying the theory's reliance on such rich ontological commitments.

Before moving to the conclusion it must be pointed out that through the *Survive Principle* we seek to remove Internal Merge *without* abandoning a derivational view of syntax. Taking into account Brody's (1998, 2002) excellent and accurate criticism of minimalism being a mixed, weakly derivational theory, we needn't surrender minimalism to a purely representational line of thought. On the contrary, we seek to remove Internal Merge which, according to Stroik (forthcoming), is the culprit of minimalism's classification as being partly-derivational and partly-representation. The elimination of Internal Merge in favor of Survive creates a pure derivational view of minimalism. Most importantly, our revision of the minimalism program does not increase the computational workload of the language faculty, but rather significantly reduces constraints and ontological internal interfaces in the narrow syntax as well as what materials appear at the interfaces. These adjustments bring us one step closer to Frampton and Gutmann's (2002) vision of a crash-proof syntax and how it should operate.

6 CONCLUSION

In this paper we sought out to address the Principle C asymmetries connected with adjunction reconstruction. We showed where both proposed solutions affiliated with Internal Merge (or anything theory of constituent displacement by means of *Move* for that matter) face significant shortcomings in the analysis presented here. Chomsky's (2001) Simpl taxes the computational language faculty with an operation that forces it to reconfigure previous 'invisible' structure for syntactic considerations, while the application of Fox's Late Merge (2004) forces the allowance of "tucking in" of structures to previously merged constituents. Based on the work of Stroik (1999, forthcoming) and Stroik and Putnam (in progress), we proposed a version of constituent displacement perpetuated by the repulsion of objects rather than attraction (*Survive*). Furthermore we adopted a version of Fox's Late Merge (2004) that applies in the Numeration rather than in the narrow syntax which we labeled *Late Num Merge*. Two conceptual advantages to our approach immediately come to the forefront: First, we do not have nor require any non-cyclic applications of Merge (or any other operation for that matter) in our system. Second, *Late Num Merge* obviates the "tucking in" issue associated with Fox's (2004) current formulation of Late Merge.

The implementation of Survive has far-reaching effects on a view of minimalist syntax. By removing Internal Merge from the system, economy constraints, the concept of phases and multiple Spell-Out are also deemed disposable due to the fact they are no longer conceptually necessary.

References

Adger, David. 2003. *Core Syntax.* Oxford: Oxford University Press.

Brody, Michael. 1998. Projection and phrase structure. *Linguistic Inquiry* 29: 367-98.

Brody, Michael. 2002. On the status of representations and derivations. In Samuel D. Epstein and T. Daniel Seely (eds.), *Derivation and Explanation in the Minimalist Program*, Oxford: Blackwell, 90-105.

Chomsky, Noam. 1965. *Aspects of the theory of syntax.* Cambridge, MA: MIT Press.

Chomsky, Noam. 1980. *Lectures on government and binding.* Dordrecht: Foris Publications.

Chomsky, Noam. 1993. A minimalist program for linguistic theory. In K. Hale and S. J. Keyser (eds), *The view from Building 20*, 1-52. Cambridge, MA: MIT Press.

Chomsky, Noam. 1995. *The minimalist program.* Cambridge, MA: MIT Press.

Chomsky, Noam. 2001. Beyond explanatory adequacy. *MIT Occasional Papers in Linguistics 20*. Cambridge, MA: MITWPL.

Collins, Chris. 1997. *Local Economy.* Cambridge, MA: MIT Press.

Collins, Chris. 2002. Eliminating labels. *MIT Occasional Papers in Linguistics 20*.
Cambridge, MA: MIT Press.

Epstein, Samuel et al. 1997. *A Derivational Approach to Syntactic Relations*. Oxford: Oxford University Press.

Fox, Danny. 2003. On logical form. In Randall Hendrick (ed), *Minimalist Syntax*, 82-123. Oxford: Blackwell Publishing.

Fox, Danny. 2004. Condition A and scope reconstruction. *Linguistic Inquiry* 35: 475-485.

Frampton, John and Sam Gutmann. 2002. Crash-proof syntax. In Samuel D. Epstein and T. Daniel Seely (eds.), *Derivation and Explanation in the Minimalist Program*, Oxford: Blackwell, 90-105.

Frege, Gottlob. 1884. *Die Grundlagen der Arithemtik. Eine logisch mathematische Untersuchung über den Begriff der Zahl.* Breslau: Koebner.

Lebeaux, David. 1991. Relative clauses, licensing, and the nature of the derivation. In S. Rothstein (ed), *Perspectives on Phrase Structure: Heads and Licensing*, 209-239. San Diego, CA: Academic Press.

Lebeaux, David. 1995. Where does the Binding Theory apply? In *University of Maryland Working Papers in Linguistics 3*, 63-88. Department of Linguistics, University of Maryland.

Putnam, Michael. 2006a. *Scrambling in West Germanic as XP-Adjunction: A Critical Analysis of Prolific Domains.* Ph.D. Dissertation, University of Kansas.

Putnam, Michael. 2006b. Eliminating Agree. Ms. University of Michigan.

Rubin, Edward. 2003. Determining Pair-Merge. *Linguistic Inquiry* 34: 660-668.

Stroik, Thomas. 1999. The survive principle. *Linguistic Analysis* 29: 278-303.

Stroik, Thomas. Forthcoming. *Locality in Minimalist Syntax*. Cambridge, MA: MIT Press.

Stroik, Thomas and Michael Putnam. In Progress. *Beyond Minimalism* [tentative title]. Ms. University of Missouri-Kansas City and University of Michigan.

Uriagereka, Juan. 1999. Multiple spell-out. In Samuel D. Epstein and Norbert Hornstein (eds.), *Working Minimalism*, Cambridge, MA: MIT Press, 251-82.