

Distinguishing Copies and Repetitions

Chris Collins and Erich Groat

January 2018

Abstract: In this squib we consider a number of proposals for distinguishing copies and repetitions in the minimalist syntax literature. We show that each of these proposals faces difficulties. We conclude that there is no known way of distinguishing copies and repetitions in minimalist syntax.

Keywords: copies, repetitions, chains, occurrences, multi-dominance, phases

1. Copies and Repetitions

In minimalist syntax, the structure building operation is Merge, defined as in (1) (see Collins 2002, Seely 2006 and Chomsky 2013):

$$(1) \quad \text{Merge}(X,Y) = \{X,Y\}$$

If X and Y are independent (neither one dominates the other), then (1) is an instance of external Merge. If X dominates Y (or vice versa) then (1) is an instance of internal Merge. Internal and external Merge are two separate cases of the Merge operation, not distinct operations (see Chomsky 2004).

Internal Merge generates structures such as the following:

$$(2) \quad \text{John was seen } \langle \text{John} \rangle$$

In the passive sentence in (2), *John* has been externally merged as the complement of the passive verb *seen*. Subsequently, *John* is internally merged as the subject of the clause. In some minimalist literature, these two occurrences are called copies since they are related by internal Merge. The angled brackets around the rightmost occurrence of *John* indicate non-pronunciation (at Transfer/Spell-Out/Externalization). For ease of reference, we will henceforth refer to the leftmost occurrence of *John* as the first occurrence, and the rightmost occurrence as the second occurrence.

The structure in (2) needs to be distinguished from the structure in (3):

$$(3) \quad \text{John saw John.}$$

In (3), *John* has been externally merged as the complement of the verb. Subsequently *John* is externally merged as the subject of the clause. In some minimalist literature, these two occurrences are called repetitions, since they are not related by internal Merge. Note that in this case, both occurrences are pronounced.

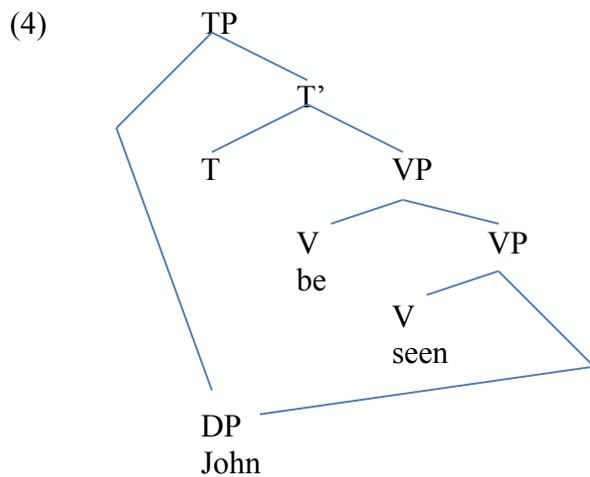
The question is how to distinguish examples like those in (2) and (3). That is, how does one distinguish the notion of copy from the notion of repetition? Concretely, how does Transfer/Spell-Out/Externalization know that the two occurrences in (2) are related by internal Merge, but the two occurrences in (3) are not?

In this paper, we will consider a number of proposals that have been given in the literature, these include: multi-dominance, chains, phase level memory, Spell-Out as part of

internal Merge, and (phase-)contextual. We will show that in each case, the proposal faces difficulties. In particular, any solution to the issue of distinguishing copies and repetitions consistent with minimalist aims must meet the following criteria: (a) no operations other than Merge should be used to build structure, (b) nothing beyond lexical items and the structures built from them by Merge should be interpreted by the interfaces, and (c) the definition of Merge should not be made more complex than the definition in (1). No current proposal satisfies these criteria. We conclude that no adequate proposal exists in minimalist syntax for distinguishing copies and repetitions.

2. Multidominance

On a multidominance theory (see Citko 2011a, 2011b for overview and extensive references), the sentence in (2) has the following structure (assuming labels, and putting aside irrelevant details such as the nature of the implicit argument and the v/vP):



In (4), *John* is immediately dominated twice, once by the TP and once by the VP.

One issue that comes up right away is that (4) is a graph theoretic object. In minimalism, Merge forms sets $\{X,Y\}$, so the correct representation of (2) would be:

(5) $\{\text{John}, \{\text{T}, \{\text{be}, \{\text{seen}, \text{John}\}\}\}\}$

Compare (5) to the structure of (3) which is given in (6):

(6) $\{\text{John}, \{\text{T}, \{\text{see}, \text{John}\}\}\}$

In both (5) and (6), *John* has two occurrences. On the multi-dominance account, there is only one syntactic object in (5) with two occurrences, but in (6) there are two syntactic objects each with its own occurrence. However, nothing in the structures in (5) and (6) allows one to conclude that there is one syntactic object in (5), but two syntactic objects in (6).

In order to implement this intuition, something additional needs to be added to the representations. One possibility is a diacritic, like an index (see Chomsky 1995, Collins and Stabler 2016). Given this modification, the structures are as follows:

(7) {John₁, {T, {be, {seen, John₁}}}}

(8) {John₁, {T, {see, John₂}}}

Now *John*₁ has two occurrences in (7), whereas in (8) *John*₁ and *John*₂ have one occurrence each. (7) can be taken as the minimalist Merge based implementation of the multi-dominance analysis.

The problem with the structures in (7) and (8) is that they require a diacritic to work. And diacritics of this nature are clear violations of the inclusiveness condition, stated in (9):

(9) Inclusiveness

“...bars introduction of new elements (features) in the course of computation: indices, traces, syntactic categories or bar levels, and so on.” (Chomsky 2001:2-3)

As noted in Collins and Stabler 2016, Inclusiveness is actually a theorem of a system that involves Merge. For this reason, multi-dominance approaches to movement must be rejected as being inconsistent with a fundamental principle of UG.

3. Chains

In a chain based theory, (2) is distinguished by (3) in terms of the chains formed. A chain is defined as a sequence of occurrences. The first occurrence in the sequence is the head of the chain. The last in the sequence is the tail. An occurrence is defined as a location in a tree. See Collins and Stabler 2016 for formal definitions of these notions. The chains in (2) and (3) are illustrated in (10):

- (10) a. One chain: <first occurrence of *John*, second occurrence of *John*>
b. Two chains: first occurrence of *John*, second occurrence of *John*

In (10a), *John* has two occurrences and these occurrences are linked by a chain. In (10b), *John* once again has two occurrences, but they are not linked by a chain, rather there are two trivial (one member) chains. Note on this theory that there is no need for diacritics (unlike in the multi-dominance approach). The work of the diacritics is being done by the chains.

The problem with chain based accounts is the massive machinery that they require to get off the ground. A chain is a sequence of occurrences, so minimally one needs definitions of chain and occurrence. Furthermore, one needs to assure that when internal Merge takes place, a chain is formed. But this formation of a chain goes way beyond the simple definition of Merge in (1), which does not form chains. In effect, one is introducing a new operation (Form Chain, see Nunes 2004) into the theory. Lastly, one needs to define Transfer/Spell-Out/Externalization to make reference to chains rather than simply to syntactic objects created by Merge. All of these steps are non-trivial.

4. Phase Level Memory

Chomsky has suggested in several places that the issue of distinguishing copies and repetitions can be addressed by introducing the notion of phase level memory. Some quotes are given in (11-12):

- (11) Chomsky 2008
 “There must be some way to identify internally merged α with its copy, but not with other items that have the same feature composition: to distinguish, say, ‘John killed John’ or ‘John sold John to John’ (with syntactically unrelated occurrences of John), from ‘John was killed John’ (with two copies of the same LI John). That is straightforward, satisfying the inclusiveness condition, if within a phase each selection of an LI from the lexicon is a distinct item, so that all relevant identical items are copies. Nothing more than phase-level memory is required to identify these properties at the semantic interface C-I, where the information is required.”
- (12) Chomsky, Gallego and Ott 2017
 “At TRANSFER, phase-level memory suffices to determine whether a given pair of identical terms Y, Y’ was formed by IM. If it was, then Y and Y’ are copies; if it was not (i.e., it was formed by EM), Y and Y’ are independent repetitions.”

The two quotes both invoke phase level memory, but in different ways. The quote in (11) talks about selection of lexical items, and the quote in (12) talks about whether IM or EM takes place.

Let us consider the following structures for (2) and (3). Under each are listed the kinds of phase level memory described by the above quotes:

- (13) {John, {T, {be, {seen, John}}}}
 Selection: *John* was selected from the lexicon only once, so the first and the second occurrences are copies.
 EM vs. IM: the first occurrence of *John* was formed by IM, so the first and the second occurrences are copies.
- (14) { John {v, {saw, John}}}
 Selection: *John* was selected from the lexicon twice, so the first and second occurrences are distinct repetitions.
 EM vs. IM: The first occurrence of *John* was formed by EM, so the two occurrences of John are distinct repetitions.

The first approach, (11), brings up many questions. In what way are occurrences of *John* marked as distinct in (14)? The quote suggests that this can be done by looking at the selections of lexical items from the lexicon. How are these lexical selections recorded and stored in phase level memory? Of course, one can infer the lexical selections that have taken place by looking at the sequence workspaces. If the lexical item *John* is not in W_1 but it is in the following W_2 , then *John* was selected from the lexicon. But, once one establishes the lexical selections, how is this information relayed to the interfaces? Furthermore, in the case of complex syntactic objects such as *the governor of California*, exactly how would the selections of the lexical items out of which it is built be associated with it?

The second approach, (12), brings up a similar set of issues. How is the fact that *John* was formed by IM stored? Note that in minimalist syntax, there is no separate operation IM, there is just Merge that has two separate cases. So one needs a way to identify whether or not IM has applied to form (13) or (14). Again, the only way to do this is to look at a previous

workspace, in this case the immediately preceding workspace (W_{-1}), and compare it to the current workspace (W_0); there is no way that one can tell whether the two occurrences are related by IM simply by looking at the contents of W_0 .

To see this concretely, let's make the assumption (see Chomsky, Gallego, Ott 2017), that If X and Y are in a workspace W, then Merge replaces X and Y with {X,Y} (removing X and Y from the workspace). Clearly in a derivation of (13) the structure {T, {be, {seen, John}}}} will be in W_{-1} . Now the question is whether *John* was also a member of W_{-1} . If not, then we can conclude that only IM was involved, and hence the occurrences of *John* in (13) must be copies. But suppose that *John* was a member of W_{-1} . Then the only way to know whether IM or EM was involved in forming (13) is by looking at W_0 and seeing if it contains *John*. If so, (13) must have been formed from IM (since the extra occurrence of *John* in the workspace is not used up).

In either case – be it the number of lexical selections or the distinction between IM and EM – memory of preceding workspaces in addition to the current workspace is not just required, but needs to be encoded. This can be done in various ways, by using chains or diacritics for example. But then we are back to square one, since we have already established that using chains and diacritics is not consistent with the aims discussed in (a), (b) and (c) at the end of section 1.

The question is what the nature of the memory in (13) and (14) actually is. This question can be broken up into at least three questions:

- (15) a. What is a memory representationally?
- b. Where is the memory represented?
- c. How is the memory accessed by Transfer/Spell-Out/Externalization?

If the memory (14c) is not represented via chains or diacritics, then how is it represented, such that it can be accessed by Transfer? Precisely what are the data structures interpreted by Transfer? They do not appear to be structures created by Merge. Without answers to these questions, phase level memory is not a viable solution.

5. Spell-Out as Part of Internal Merge

Another way of distinguishing copies and repetitions is to make the interface operations more tightly connected to internal Merge. Perhaps the best instance of this kind of analysis is found in Stabler 1997, where phonetic manipulations are made as part of the definition of Move itself (see also Collins and Sabel 2015). As Stabler notes: “Overt movement leaves behind just a node labeled by the empty sequence of features λ .” This property is encoded directly into the definition of Move.

Such a definition for Move does in fact distinguish copies and repetitions. By definition, the lower copy will not have any features, including phonetic features, and so it will not be pronounced. However, it is not clear how to formalize Stabler's approach in a way which leaves the simple nature of Merge as defined in (1) intact. In particular, Stabler's approach explicitly has two operations Merge (which is binary) and Move (which is unary). It is unclear how these could be unified into a single Merge operation, given that one function of Move is to delete the features of copies.

6. Contextual Distinction between Copies and Repetitions

Martin and Uriagereka (2014) suggest an approach in which lexical items, and the syntactic objects built from them, are treated within Narrow Syntax purely as non-individuated

In this paper we have reviewed various proposals to distinguish copies and repetitions in the minimalist syntax literature. We have shown that none of these are satisfactory. Multi-dominance violates Inclusiveness. Chains require a new operation Form Chain. Phase level memory does not clearly avoid the problems of chains and multi-dominance.

The problem can be understood as an apparent limit on the power of Merge to create objects that naturally encode the distinction. The structure-building involved in Internal Merge is identical to that in External Merge – the creation of a new set $\{X, Y\}$ out of pre-existing sets X and Y (Epstein et al 1998, Chomsky 2000, 2008) – leaving no trace of the origins of X or Y. The attractiveness of a theory that takes such a simple operation as Merge as its sole structure-building operation compels us to resist arbitrary complications, yet no approach so far has been able to make sense of the distinction between copies and repetitions without doing so.

Chomsky (2004: 110) sums up an important advance of recent years:

- (19) NS is based on the free operation Merge. SMT entails that Merge of α, β is unconstrained, therefore either external or internal. Under external Merge, α and β are separate objects; under internal Merge, one is part of the other, and Merge yields the property of “displacement,” which is ubiquitous in language and must be captured in some manner in any theory. It is hard to think of a simpler approach than allowing internal Merge (a grammatical transformation), an operation that is freely available. Accordingly, displacement is not an “imperfection” of language; its absence would be an imperfection.

There is no question that the move to a Merge-based view of structure-building has provided tremendous insight into the otherwise mysterious phenomenon of displacement. But until the matter of distinguishing copies from repetitions has been settled, our understanding is far from complete.

We do not take these issues to be a failure of minimalist syntax but rather as a challenge. We hope future work will come up with new alternatives that do not face the obstacles we have pointed out here. We believe that formalization could help to resolve these issues.

References

- Chomsky, Noam. 1995. *The Minimalist Program*. MIT Press, Cambridge.
- Chomsky, Noam. 2000. Minimalist inquiries: the framework. In *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, eds. Roger Martin, David Michaels and Juan Uriagereka, 89-155. MIT Press.
- Chomsky, Noam. 2004. Beyond Explanatory Adequacy. In *Structures and Beyond*, ed. Adriana Belletti, 104–131 (originally published as: Chomsky, Noam. 2001. Beyond Explanatory Adequacy. MIT Occasional Papers in Linguistics 20. MIT Working Papers in Linguistics.)
- Chomsky, Noam. 2008. On Phases. In *Foundational Issues in Linguistic Theory*, eds. Robert Freidin, Carlos P. Otero, and Maria Luisa Zubizarreta, pgs. 133-166. Cambridge, MA: MIT press.
- Chomsky, Noam. 2013. Problems of Projection. *Lingua* 130, pgs. 33-49

- Chomsky, Noam, Angel Gallego, and Dennis Ott. 2017. Generative Grammar and the Faculty of Language: Insights, Questions, and Challenges. (<https://ling.auf.net/lingbuzz/003507>)
- Citko, Barbara. 2011a. Multidominance. In Cedric Boeckx (ed.), *Handbook of Linguistic Minimalism*, 119-142. Oxford University Press, Oxford.
- Citko, Barbara. 2011b. *Symmetry in Syntax: Merge, Move and Labels*. Cambridge, MA: Cambridge University Press.
- Collins, Chris. 2002. Eliminating Labels. In *Derivation and Explanation in the Minimalist Program*, eds. Samuel David Epstein and T. Daniel Seely, pgs. 42-64. Oxford: Blackwell.
- Collins, Chris and Joachim Sabel. 2016. A C/I-Interface condition on In Remnant Movement. In Günther Grewendorf (ed.), *Remnant Movement (SGG 123)*, 93-131. De Gruyter.
- Collins, Chris and Edward Stabler. 2016. A Formalization of Minimalist Syntax. *Syntax* 19, 43-78.
- Epstein, Samuel, Erich Groat, Ruriko Kawashima, and Hisatsugu Kitahara. 1998. *A Derivational Approach to Syntactic Relations*. Oxford: Oxford University Press.
- Martin, Roger and Juan Uriagereka. 2014. Chains in Minimalism. In *Minimalism and Beyond: Radicalizing the interfaces*, edited by Peter Kosta, et al., John Benjamins Publishing Company. 169-194.
- Nunes, Jairo. 2004. *Linearization of Chains and Sideward Movement*. Cambridge, MA: MIT Press. Cambridge: MIT Press.
- Seely, T. Daniel. 2006. Merge, Derivational C-Command, and Subcategorization in a Label-Free Syntax. In *Minimalist Essays*, ed. Cedric Boeckx. Amsterdam: John Benjamins.