The Sideward Pair-MERGE of the Relative Clause and Its Antecedent*

Takahiro Tozawa

Abstract

This paper argues that Sideward Movement is possible as long as it is adjunction (Pair-MERGE), although Chomsky (2019a, b, c) and Chomsky, Gallego, and Ott (2019) exclude Sideward Movement as illegitimate. We show that our argument makes Nunes' (2001, 2004) cyclic analysis for certain cases of relative clause adjunction available under the theory of MERGE and offers a third-factor-based account for the reconstruction asymmetry between complement clauses and relative clauses.

***Keywords*:** MERGE, late merge, sideward movement, sideward pair-MERGE, relative clause

1. Introduction

Lebeaux (1988, 1991) observes the reconstruction asymmetry between the complement clause in (1a) and the relative clause in (1b).

(1) a. *Whose claim that $John_i$ likes Mary did he_i deny *t*?

b. Which claim that $John_i$ made did he_i later deny *t*?

(Lebeaux (1991: 211))

Explorations in English Linguistics, Number 34, November 30, 2020
Department of English Linguistics, Tohoku University
71-97

While *he* in (1b) can be coreferential with *John*, *he* in (1a) cannot. These facts show that the complement clause in (1a) is reconstructed to its base position unlike the relative clause in (1b). The complement clause is reconstructed to its base position, as a result of which *John* is bound by *he*, triggering the Condition C violation. By contrast, the relative clause is not reconstructed, avoiding the Condition C violation. Lebeaux accounts for this contrast by proposing that adjuncts can be merged countercyclically. The derivations of the sentences in (1a, b) are shown in (2) and (3), respectively.

(2) $[_{CP}$ $[_{DP}$ whose claim that John$_i$ likes Mary$]_j$ did he$_i$ deny $t_j]$

(3) a. $[_{CP}$ $[_{DP}$ which claim$]_j$ did he$_i$ later deny $t_j]$

b. $[_{CP}$ $[_{DP}$ $[_{DP}$ which claim$]_j$ [that John$_i$ made]] did he$_i$ later deny $t_j]$

In (2), the complement clause is base-generated in the object position of the verb *deny* and then undergoes *wh*-movement to the sentence-initial position.[1] *John* in the complement clause is bound by *he*, which violates Condition C. Therefore, *he* and *John* in (1a) cannot be coreferential. In (3a), *which claim* occurs as the complement of *deny* and then moves to the sentence-initial position. After *wh*-movement, the relative clause is countercyclically adjoined to the *wh*-phrase as in (3b).[2] Let us call this countercyclic operation late Merge. Late Merge allows *John* to be generated outside the c-command domain of *he*. Therefore, *he* and *John* can be coreferential in (1b).

Although this approach is convincing, late Merge itself has some problems. One of the problems is that according to Epstein, Kitahara, and Seely (2012), a form of replacement is needed to get the structure formed by late Merge. Let us examine this problem in detail using (3) as an example. Consider (4).

(4) a. [$_{CP1}$ [$_{DP1}$ which claim]$_j$ [$_{C'}$ did he$_i$ later deny *t*$_j$]]

b. [$_{CP2}$ [$_{DP2}$ [$_{DP1}$ which claim]$_j$ [that John$_i$ made]] [$_{C'}$ did he$_i$ later deny *t*$_j$]]

In (4a), DP *which claim* undergoes *wh*-movement, forming CP. The subscript notation "1" is added to DP and CP for expository purposes. Next, the relative clause is late-merged with the *wh*-phrase, creating new DP. We use the subscript notation "2" to indicate this new DP. The sister of C′ has changed from DP_1 to DP_2. DP_2 replaces DP_1 as the sister of C′. This is replacement. In other words, replacement removes DP_1 and combines DP_2 with C′. Such replacement also creates new CP, which is indicated as CP_2 in (4b). Replacement changes CP_1 in (4a) (which is represented as {DP_1, C′} in set notation) into CP_2 in (4b) (which is represented as {DP_2, C′} in set notation). The general Merge operation cannot replace DP_1 with DP_2 or CP_1 with CP_2. Therefore, we need to posit some operations for replacement in addition to Merge, which goes against the spirit of the Minimalist Program, which attempts to reduce the UG apparatus to the minimum. Considering that late Merge consists of Merge and extra operations responsible for replacement, it is desirable that we capture the contrast between (1a) and (1b) in a cyclic fashion.[3]

We argue for Nunes' (2001, 2004) cyclic analysis for relative clause adjunction in (1b) based on Sideward Movement. Although Chomsky (2019a, b, c) and Chomsky, Gallego, and Ott (2019) consider Sideward Movement illegitimate, we argue that Sideward Movement is possible as long as it is adjunction. Based on this argument, we show that the contrast between (1a) and (1b) follows from Resource Restriction, which is a third factor principle.

The rest of this paper is organized as follows. Section 2 provides an outline of the framework of MERGE of Chomsky (2019a, b, c) and Chomsky, Gallego, and Ott (2019). Section 3 proposes that Sideward Movement is permitted as far as it is Pair-MERGE. Section 4 presents a cyclic analysis for relative clause adjunction in (1b) in terms of Sideward Pair-MERGE, accounting for the contrast between (1a) and

(1b) in a principled manner. Section 5 discusses the consequence of our proposal. Section 6 concludes the paper.

2. The Theory of MERGE

In this section, we review the theory of MERGE proposed by Chomsky (2019a, b, c) and Chomsky, Gallego, and Ott (2019).

2.1. Redefinition of Merge

The Minimalist Program is a research program that seeks to build a linguistic theory based on what is minimally necessary. Under the minimalist guidelines, Chomsky (2000) assumes the Strong Minimalist Thesis (SMT) in (5).

(5) The Strong Minimalist Thesis (SMT)
Language is an optimal solution to legibility conditions.
(Chomsky (2000: 96))

According to SMT, language is a perfect system in the sense that it satisfies interface conditions in a way that observes third factor principles (such as no-tampering condition and minimal search). Under SMT, UG is assumed to be a simple and elegant theory. It follows that the structure-building operation Merge, which is the only operation of UG, should be simple. In an earlier stage of the Minimalist Program, Merge is defined as in (6).

(6) The simplest such operation takes a pair of syntactic objects (SO_i, SO_j) and replaces them by a new combined syntactic object SO_{ij}. Call this operation *Merge*. (Chomsky (1995: 226))

Merge is a set-forming operation that applies to two syntactic objects, creating a new syntactic object.

However, Chomsky (2019a, b, c) mentions the problem with Merge: exocentric structures such as the subject-predicate structure. When the subject-predicate structure {NP, VP} is formed, the subject NP and the predicate VP are independently constructed before they are combined. This means that there is a place to form NP and VP in parallel and to put them together. Chomsky calls the place a workspace (WS), arguing that the operation Merge should be revised to the operation on WS, not on two syntactic objects. That is, Merge is an operation that changes WS into another WS. This version of Merge is called MERGE. The Merge operation is redefined as in (7).

(7) MERGE maps WS onto WS′.

(Chomsky (2019a, b, c), Chomsky, Gallego, and Ott (2019))

According to Chomsky (2019a) and Chomsky, Gallego, and Ott (2019), MERGE has the property of Recursion, which is shown in (8).

(8) Recursion

All syntactic objects in the lexicon and in the workspace WS are *accessible* to MERGE; (...) The basic property of recursive generation requires that any object already generated be accessible to further operations. (Chomsky, Gallego, and Ott (2019: 245))

Recursion is a property of the faculty of language, demanding that any syntactic object in WS is accessible to MERGE. The accessible elements are restricted to the minimum by Resource Restriction (RR), which is a third factor principle (Chomsky 2019b, c; Chomsky, Gallego, and Ott 2019). This is indicated in (9).

(9) Resource Restriction (RR)
The number of elements accessible to computation should be as small as possible. (Chomsky (2019b, c))

RR restricts accessible resources to the minimum. Chomsky (2019b) argues that the examples of RR are the Phase Impenetrability Condition (PIC) and minimal search. PIC prevents elements inside a phase from being accessible to operations. Minimal search requires the closest element to be accessible to operations. For example, in successive cyclic *wh*-movement, the higher copy of the *wh*-phrase rather than the lower one undergoes movement to the next landing site. This means that the lower copy of the *wh*-phrase is not accessible to the movement operation. This way, PIC and minimal search restrict accessibility. Chomsky (2019b, c) also argues that RR requires that accessible elements should appear only once in WS. That is, RR prohibits two accessible copies of a syntactic object from appearing in WS.

2.2. External Merge

We have seen that WS, Recursion, and RR are involved in MERGE. Here, using an example, we discuss how the MERGE-based derivation proceeds. Let us start with External Merge (EM). Consider the derivation of a transitive *v**P phase.

(10) a. WS1 = [Subj, *v**, V, Obj]
b. WS2 = [Subj, *v**, {V, Obj}]
c. WS3 = [Subj, {*v**, {V, Obj}}]
d. WS4 = [{Subj, {*v**, {V, Obj}}}]

Each WS in (10a-d) contains syntactic objects. WS is a set of syntactic objects including lexical items and phrasal elements (constructed structures). For convenience, the square bracket is used to represent the set.[4] WS1 in (10a) contains

Subj, *v**, V, and Obj, and these syntactic objects are accessible to MERGE owing to Recursion in (8). Then, MERGE applies to WS1, yielding WS2 in (10b), where the set {V, Obj} is formed. Given Recursion, the accessible elements in WS2 are five syntactic objects: Subj, *v**, V, Obj, and {V, Obj}. Here, when WS1 is changed to WS2, V and Obj are eliminated. That is, WS2 is not the one that includes V and Obj as in (11).

(11) WS2′= [Subj, *v**, V, Obj, {V, Obj}]

WS2′ violates RR because there are two accessible copies of V (V and V of {V, Obj}) and two accessible copies of Obj (Obj and Obj of {V, Obj}) in WS2'. RR prevents two accessible copies from appearing in WS. Therefore, V and Obj are eliminated from WS2.[5] Let us return to the derivation of the *v**P phase. MERGE applies to WS2, updating it to WS3 in (10c), where the structure {*v**, {V, Obj}} is constructed and *v** and {V, Obj} are removed. Finally, MERGE updates WS3 to WS4, where the structure {Subj, {*v**, {V, Obj}}} is built and Subj and {*v**, {V, Obj}} are eliminated.

2.3. Internal Merge

Next, we consider Internal Merge (IM). Suppose that the derivation in (10) moves on to the CP phase. Consider (12).

(12) a. WS5 = [C, T, {Subj, {*v**, {V, Obj}}}]
b. WS6 = [C, {T, {Subj, {*v**, {V, Obj}}}}]
c. WS7 = [C, {Subj {T, {Subj, {*v**, {V, Obj}}}}}]
d. WS8 = [{C, {Subj {T, {Subj, {*v**, {V, Obj}}}}}}]

MERGE applies to WS5, yielding WS6, where T is combined with {Subj, {*v**, {V, Obj}}}. Next, MERGE changes WS6 to WS7, where Subj is internally merged with

{T, {Subj, {*v**, {V, Obj}}}}. Recursion allows all syntactic elements in WS7 to be accessible to MERGE. Then, the accessible elements are C, Subj, T, Subj, *v**, V, Obj, {V, Obj}, {*v**, {V, Obj}}, {Subj, {*v**, {V, Obj}}}, {T, {Subj, {*v**, {V, Obj}}}}, and {Subj {T, {Subj, {*v**, {V, Obj}}}}}. Although there are two copies of Subj, only the higher copy is accessible to operations, owing to minimal search. Therefore, IM satisfies RR. The derivation goes on and MERGE updates WS7 to WS8, as a result of which we get the CP structure.

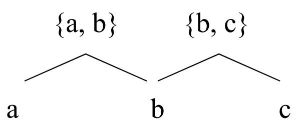
2.4. Extensions of Merge

Chomsky (2019a, b, c) and Chomsky, Gallego, and Ott (2019) argue that MERGE prohibits extensions of Merge: Parallel Merge, Sideward Movement, and late Merge.[6] In this section, we present an overview of the illegitimacy of these extensions of Merge.

2.4.1. Parallel Merge

First, we discuss Parallel Merge as proposed by Citko (2005). This is illustrated in (13).

(13)



Suppose that after *a* is externally merged with *b*, *b* undergoes Merge with *c*. Merge of *b* and *c* is called Parallel Merge. Such Merge has the property of EM in that two independent syntactic objects are combined but it also has the property of IM in that a subpart of the constructed structure (the set {*a*, *b*}) is taken as one of two syntactic objects.

However, Parallel Merge is problematic under the framework of MERGE because it creates two accessible copies of a syntactic element, which violates RR. Consider the derivation in (14).

(14) a. WS1 = [a, b, c]
b. WS2 = [{a, b}, c]
c. WS3 = [{a, b}, {b, c}] (Parallel Merge)

MERGE applies to WS1 in (14a), yielding WS2 in (14b), where *a* undergoes EM with *b*. Next, MERGE updates WS2 to WS3 in (14c), where the set {*b*, *c*} is formed by Parallel Merge. Recursion allows all syntactic objects in WS3 to be accessible to operations. The accessible syntactic objects in WS3 are *a*, *b*, *b*, *c*, {*a*, *b*}, and {*b*, *c*}. It is important to note that two copies of *b* are accessible, which violates RR. Therefore, Parallel Merge is illegitimate.[7]

2.4.2. Sideward Movement

Second, we consider Sideward Movement. Nunes (2001, 2004) proposes Sideward Movement, which allows a copy of a syntactic object in a structure K to merge with the syntactic object L constructed independently of the structure K. This is illustrated in (15).

(15) a. $[_K \ldots \alpha_i \ldots]$ α_i ⟷ $[_L \ldots]$
Copy Merge
b. $[_K \ldots \alpha_i \ldots]$ $[_M\ \alpha_i\ [_L \ldots]]$

(cf. Nunes (2001: 305))

A series of operations in (15a, b) show Sideward Movement of α. In (15a), α in the syntactic object K is copied and merged with the syntactic object L, which is constructed in parallel with K. As a result, the syntactic object M is formed as in (15b).

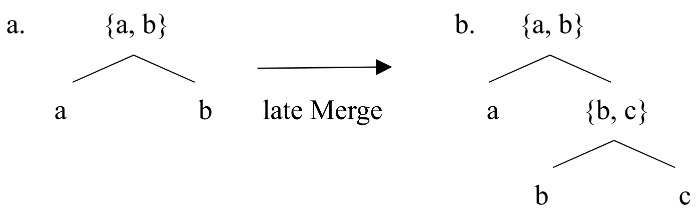
Sideward Movement is impossible under the MERGE theory because it results in a violation of RR like Parallel Merge. The derivation of Sideward Movement proceeds in the same way as that of Parallel Merge. The derivation is shown in (16).

(16) a. WS1 = [a, b, c]
b. WS2 = [{a, b}, c]
c. WS3 = [{a, b}, {b, c}] (Sideward Movement)

Suppose that the derivation proceeds to WS2 in (16b). Here, MERGE applies to WS2, yielding WS3, where *b* undergoes Sideward Movement to *c*, which is a syntactic object independent of the set {*a*, *b*}. Consequently, the set {*b*, *c*} is created. All syntactic objects in WS3 are accessible because of Recursion. The accessible elements are *a*, *b*, *b*, *c*, {*a*, *b*}, and {*b*, *c*}. There are two accessible copies of *b*, which violates RR. Therefore, Sideward Movement is excluded just like Parallel Merge.

2.4.3. Late Merge

Third, we turn to late Merge, which is a countercyclic operation which applies to an element inside the structure that has already been constructed.[8] Consider (17).

(17)


In (17a), Merge of *a* and *b* takes place, creating the set {*a*, *b*}. Then, late Merge applies to *b* inside the structure, as a result of which {*b*, *c*} is formed as in (17b). The

MERGE theory bans late Merge because late Merge leads to a violation of RR just like Parallel Merge and Sideward Movement. The MERGE-based derivation of late Merge is the same as that of Parallel Merge and Sideward Movement. Consider (18)

(18) a. WS1 = [a, b, c]
b. WS2 = [{a, b}, c]
c. WS3 = [{a, b}, {b, c}] (late Merge)

After MERGE updates WS1 to WS2, it changes WS2 to WS3, where *c* is late-merged with *b* and the set {*b*, *c*} is newly created. As we have already seen, the two accessible copies of *b* induce the violation of RR.[9] Thus, late Merge is no longer permitted.

We have seen that Parallel Merge, Sideward Movement, and late Merge are illegitimate operations under the framework of MERGE. Therefore, these extensions of Merge are eliminated from the theory of grammar.

2.5. Pair-MERGE

Chomsky (2019b, c) and Chomsky, Gallego, and Ott (2019) argue that structure-building mechanisms include an asymmetrical operation Pair-MERGE in addition to a symmetrical operation MERGE.[10] Pair-MERGE yields an adjunction structure, creating an ordered pair. For example, consider (19).

(19) [... [$_{NP}$ [$_{AP}$ young] [$_{NP}$ man]]]

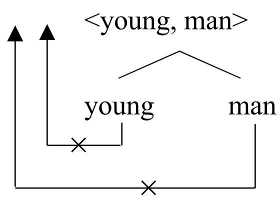
The adjunction structure results from Pair-MERGE of the adjunct AP *young* and NP *man*. Pair-MERGE creates the asymmetrical structure where the adjunct AP *young* is attached to NP *man* and the category of the entire structure is NP. This structure is asymmetrical in that NP rather than AP is the label of the entire structure.[11]

Pair-MERGE forms an ordered pair <α, β>, which means that α adjoins to β.[12] Pair-MERGE makes the elements of the ordered pair inaccessible. This is indicated in (20).

(20) The elements of the ordered pair formed by Pair-MERGE are inaccessible.

As schematically shown in (21), Pair-MERGE yields the ordered pair <*young*, *man*>.

(21)



Neither *young* nor *man* can be extracted out of the ordered pair, whereas the ordered pair <*young*, *man*> itself can undergo movement. This way, Pair-MERGE makes the elements of the ordered pair inaccessible.

3. A Proposal

As we have already seen in (20), α and β of the ordered pair <α, β> are inaccessible. Considering this property of the ordered pair, we agree with Kitahara (2019) that the elements of the ordered pair are not accessible terms for applications of MERGE. This enables us to pursue the possibility that Sideward Movement is allowed in certain cases. Let us discuss this possibility in detail. Consider (22).

(22) a. WS1 = [a, b, c]
b. WS2 = [{a, b}, c]
c. WS3 = [{a, b}, <b, c>] (Pair-MERGE)

Let us assume that the derivation has reached (22b). Pair-MERGE applies to WS2, yielding WS3 in (22c), where *b* undergoes Pair-MERGE with *c*. Since *c* is a syntactic object independent of {*a*, *b*}, we call this type of Pair-MERGE Sideward Pair-MERGE.[13] Sideward Pair-MERGE does not violate RR because *b* of the ordered pair <*b*, *c*> is inaccessible and only *b* of {*a*, *b*} is accessible. Therefore, we propose (23).

(23) Sideward Movement is permitted as far as it is Pair-MERGE.

In the next section, we show that under this proposal, Nunes' (2001, 2004) cyclic analysis for relative clause adjunction in certain instances is tenable.

4. Analysis

We are in favor of Nunes' (2001, 2004) cyclic approach to relative clause adjunction in a case where the relative clause is not reconstructed. If we adopt Nunes' cyclic analysis, we can give a principled account for the contrast between (1a) and (1b), repeated here as (24a) and (24b), respectively.

(24) a. *Whose claim that John$_i$ likes Mary did he$_i$ deny *t*?
b. Which claim that John$_i$ made did he$_i$ later deny *t*?
(Lebeaux (1991: 211))

Let us first consider (24b). The derivation under the MERGE-based cyclic analysis is indicated in (25).

(25) a. WS1 = [{did he later deny which claim}, {that John made}]
b. WS2 = [{did he later deny which claim}, <{that John made} {which claim}>]
c. WS3 = [{<{that John made} {which claim}> {did he later deny which claim}}]

Suppose that the derivation has reached the stage of WS1 in (25a), where the matrix and relative clauses are constructed in parallel. Here, Pair-MERGE applies to WS1, yielding WS2, where *which claim* undergoes Sideward Pair-MERGE with the relative clause.[14, 15] Note that Sideward Pair-MERGE does not violate RR because WS2 does not include two accessible copies of *which claim*: *which claim* of the ordered pair <*that John made*, *which claim*> is inaccessible and only the object *which claim* of the verb *deny* is accessible.[16] Next, MERGE updates WS2 to WS3, where the *wh*-phrase is merged with {*did he later deny which claim*}. Then, *he* does not c-command *John* at any stage of the derivation. Therefore, *he* and *John* in (24b) can be coreferential.[17, 18]

Next, we turn to the derivation of (24a). This is shown in (26).

(26) a. WS1 = [{did he deny whose claim}, {that John likes Mary}]
b. WS2 = [{did he deny whose claim}, {claim {that John likes Mary}}]
c. WS3 = [{did he deny whose claim}, {whose {claim {that John likes Mary}}}]
d. WS4 = [{{whose {claim {that John likes Mary}}}{did he deny whose claim}}]

We assume that the derivation proceeds to the stage in (26a). Here, MERGE changes WS1 to WS2, where *claim* is sideward-merged with the complement clause. After that, MERGE applies to WS2, yielding WS3, where *whose* undergoes Sideward Movement to NP *claim that John likes Mary*. Finally, MERGE updates WS3 to WS4, where the *wh*-phrase is merged with CP *did he deny whose claim*. Although *he* does not c-command *John* at any stage of the derivation and therefore, there is no Condition C violation, this derivation violates RR. Concretely, when WS1 is mapped to WS2, *claim* undergoes Sideward Movement to the complement clause. But this type of Sideward Movement is Set-MERGE rather than Pair-MERGE. Therefore, two accessible copies of *claim* are created in WS2. Similarly, two accessible copies of *whose* are created in WS3 by Sideward Set-MERGE. Two accessible copies of *claim* in WS2 and two accessible copies of *whose* in WS3 lead to the RR violation.[19] Therefore, (24a) is ungrammatical. Note that if Sideward Movement of *claim* and *whose* does not take place, a Condition C violation occurs. Consider the structure in (27).

(27) WS = [{{whose {claim {that John likes Mary}}} {did he deny {whose {claim {that John likes Mary}}}}}]

Suppose that the derivation proceeds to the final stage, where *claim* is merged with the complement clause, *whose* is merged with NP *claim that John likes Mary*, and the *wh*-phrase undergoes *wh*-movement to CP. In this structure, *he* c-commands *John*, which violates Condition C. Therefore, (24a) is ungrammatical. This way, (24a) is ruled out by RR or Condition C.

5. A Consequence

We have seen that Sideward Movement is possible insofar as it is Pair-MERGE. If our argument is on the right track, the same should be true for Parallel Merge: Parallel Merge is permitted as long as it is Pair-MERGE. Consider (28).

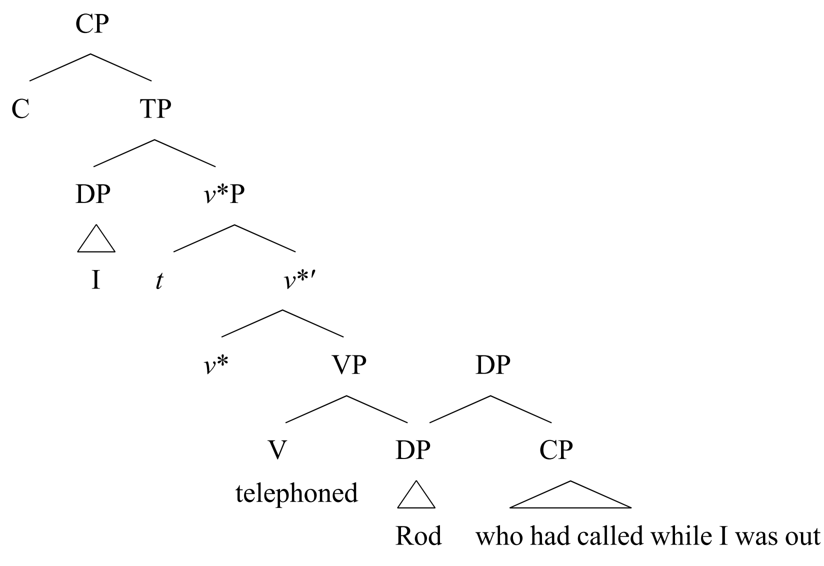
(28) a. WS1 = [a, b, c]
b. WS2 = [{a, b}, c]
c. WS3 = [{a, b}, <b, c>] (Parallel Pair-MERGE)

After the derivation proceeds to WS2, Pair-MERGE applies to WS2, yielding WS3, where *b* of {*a*, *b*} undergoes Parallel Pair-MERGE with *c*. In WS3, *b* of the ordered pair <*b*, *c*> is inaccessible and therefore, only *b* of {*a*, *b*} is accessible. There is no violation of RR. Thus, Parallel Pair-MERGE should be a legitimate operation. Note that WS3 shows that if we assume that {*a*, *b*} in WS3 is the main clause, *c* of <*b*, *c*> is generated outside the main clause. Is there any syntactic object that is generated outside the main clause? We suggest that non-restrictive relative clauses such as (29) occur outside the main clause.

(29) I telephoned Rod, who had called while I was out.

The sentence in (29) has the structure in (30).

(30)



The non-restrictive relative clause CP undergoes Parallel Pair-MERGE with DP *Rod*. The structure in (30) shows that the non-restrictive relative clause CP appears outside the CP-TP-*v**P-VP structure. Then, we predict that the element in the main clause cannot bind the element in the non-restrictive relative clause. This prediction is borne out.

(31) a. Everyone$_i$ spoke about museum that he$_i$ had visited.
b. *Everyone$_i$ spoke about the Millennium Dome, which he$_i$ had visit.
(cf. Vries (2006: 256))

Everyone can license the bound pronoun *he* in the restrictive relative clause in (31a), while it cannot license *he* in the non-restrictive relative clause in (31b). The structure of the sentence in (31b) is shown in (32).

(32) WS =[{everyone$_i$ spoke about the Millennium Dome}, <{which he$_i$ had visit} {the Millennium Dome}>]

Parallel Pair-MERGE creates the ordered pair formed independently of the main clause. In this structure, the bound pronoun *he* is outside the c-command domain of *everyone*. Then, *he* cannot be licensed by *everyone*, resulting in the ungrammaticality of (31b).[20]

One potential problem with the present analysis is that we cannot account for why the main-clause subject DP can be the antecedent of a non-restrictive relative clause. Consider (33).

(33) a. His bag, which was not so large, contained ten books.
b. *Everyone$_i$ thinks that Mary, who he$_i$ suspected, is likely to defect.

(33a) shows that the main-clause subject can serve as the antecedent of the non-restrictive relative clause, just like the main-clause object in (29). Furthermore, (33b) shows that *everyone* cannot license the bound pronoun *he* in the non-restrictive relative clause headed by the main-clause subject DP, which supports the prediction of the present analysis, as discussed above. The question then arises of how the non-restrictive relative clause, which is outside the main clause, undergoes movement to TP together with the main-clause subject. We suggest that actually, the non-restrictive relative clause in (33a) undergoes Parallel Pair-MERGE with the main-clause subject after the subject moves to TP. The derivation is shown in (34).

(34) a. WS1 = [C, {T {{his bag} contained ten books}}, {which was not so large}]

b. WS2 = [C, {{his bag} {T {{his bag} contained ten books}}}, {which was not so large}]

c. WS3 = [<{which was not so large} {his bag}>, C, {{his bag} {T {{his bag} contained ten books}}}]

d. WS4 = [<{which was not so large} {his bag}>, {C {{his bag} {T {{his bag} contained ten books}}}}]

Suppose that the derivation has reached WS1, where the main-clause TP and the non-restrictive relative clause are built in parallel. MERGE applies to WS1, yielding WS2, where the subject *his bag* moves to TP. Next, Pair-MERGE updates WS2 to WS3, where the non-restrictive relative clause undergoes Parallel Pair-MERGE with the subject in TP. Finally, MERGE changes WS3 to WS4, where C is merged with TP. In this derivation, the non-restrictive relative clause does not move to TP, which makes the Parallel Pair-MERGE analysis maintainable.[21]

6. Conclusion

We have proposed that Sideward Movement is a legitimate operation as long as it is Pair-MERGE. We have shown that this proposal makes Nunes' (2001, 2004) cyclic analysis of relative clause adjunction available under the framework of

MERGE, offering a principled account for the argument-adjunct asymmetry in reconstruction for Condition C. To the extent that our proposal is on the right track, there are three implications. First, we should rethink the view of Chomsky (2019a, b, c) and Chomsky, Gallego, and Ott (2019) that Parallel Merge and Sideward Movement should be eliminated from the theory of grammar. Second, our proposal supports Nunes' (2001, 2004) cyclic analysis for relative clause adjunction in certain cases. Third, our proposal provides support for RR, which is a third factor principle, in that we can give an RR-based account for the argument-adjunct asymmetry in reconstruction.

*This paper is a revised version of the paper presented at the 12th Hokkaido Theoretical Linguistics Society held at Asahikawa Medical University on February 15, 2020. I would like to express my appreciation to Yoshiaki Kaneko and Etsuro Shima for their invaluable comments and suggestions. I am also grateful to Yoshihisa Goto, Toru Ishii, Satoru Kanno, Nobuhiro Miyoshi, Kenji Sugimoto, and Masashi Totsuka for stimulating discussions. All errors are mine. This study is supported by a Grant-in-Aid for Scientific Research (C), No. 18K00635.

Notes

1) The complement clause must occur in the object position owing to Chomsky's (1981) Projection Principle, which requires that selectional properties of lexical items must be satisfied at all levels of syntactic representation: D-structure, S-structure, and LF. The *that*-clause is selected as the complement by *claim* and therefore, the Projection Principle requires it to appear in the complement position of *claim* in D-structure.

2) The relative clause is an adjunct, which is not restricted by the Projection Principle. Therefore, it can be merged with the relative clause head after *wh*-movement.

3) Chomsky (1995, 2019a, b, c) and Chomsky, Gallego, and Ott (2019) also point out the complexity of late Merge.

4) The set formed by MERGE is indicated by the curly bracket, which we are familiar with.

5) In other words, V and Obj in WS1 are replaced by {V, Obj} in WS2. According to Chomsky (2019a, b, c), in this sense, we go back to Chomsky's (1995) definition of Merge in (6), which is repeated here as (i).

(i) The simplest such operation takes a pair of syntactic objects (SO_i, SO_j) and replaces them by a new combined syntactic object SO_{ij}. Call this operation *Merge*. (Chomsky (1995: 226))

(i) states that the operation Merge replaces SO_i and SO_j with SO_{ij}. The definition assumes that Merge eliminates SO_i and SO_j. This way, MERGE incorporates the property of Merge.

6) Chomsky (2013: 40, fn. 20) notes that only EM and IM are legitimate if Merge is a binary operation and the notions such as multidominance and late Merge postulate an extension of Merge.

7) Minimal search does not make one of the two copies inaccessible because neither of them is higher than the other. Therefore, both are accessible.

8) There are three types of late Merge: late Merge of an adjunct, a complement, and a lexical head. See Lebeaux (1988, 1991) for late Merge of an adjunct, Takahashi and Hulsey (2009) for late Merge of a complement, and Safir (2019) for late Merge of a lexical head.

9) Late Merge has the problem concerning accessibility and the one concerning complexity, which was discussed in Section 1.

10) Also see Chomsky (2000, 2004).

11) In the case of symmetrical MERGE, the label of a constituent is determined by minimal search: when a head and XP are merged, the head is the label of the constituent. However, this does not work for asymmetrical MERGE. In the case of asymmetrical MERGE, the host of the adjunct is always the label of the entire structure, whether it is complex or not. In this sense, Pair-MERGE is taken to be an asymmetrical operation.

12) If β adjoins to α, the ordered pair is represented by <β, α>. See Chomsky (2000).

13) We can also consider the operation as Parallel Pair-MERGE. In this paper, we advance the discussion under the assumption that the operation is Sideward Movement.

14) Sideward Pair-MERGE of the type discussed here is the one by which the Sideward Movement target (the relative clause CP) adjoins to the sideward-moved element (the relative clause antecedent DP) rather than the one by which the sideward-moved element adjoins to the target. When we use the term “Sideward Pair-MERGE,” it refers to this type of Sideward Pair-MERGE.

15) It is controversial whether restrictive relative clauses adjoin to NP (DP under the

DP hypothesis of Abney (1987)) or N′. For the DP/NP adjunction analysis, see Ross (1986). For the N′ adjunction analysis, see McCawley (1981, 1998). The cyclic analysis for relative clause adjunction is consistent with the DP/NP adjunction analysis. According to Kono (2012), the relative clause adjoins to DP/NP, when it is clear what the relative clause antecedent itself refers to. Given that *which claim* in (24b) is a D-linked *wh*-phrase and it is clear what the content of *which claim* is, we can say that the relative clause adjoins to DP/NP *which claim*. We will not pursue this issue any further.

16) In the ordered pair <*that John made*, *which claim*>, *which claim* is inaccessible to MERGE because elements of the ordered pair are not accessible terms. Here the question arises of why they are not accessible terms. We leave this question for future research. I thank Yoshiaki Kaneko for pointing out this issue.

17) One might wonder whether RR restricts the way Pair-MERGE applies. Consider an alternative derivation of (24b).

(i) a. WS1 = [... deny, {which claim}, {that John made}]
    b. WS2 = [... deny, <{that John made} {which claim}>] (Pair-MERGE)
    c. WS3 = [... {deny, <{that John made} {which claim}>}] (Set-MERGE)

For convenience, we limit the discussion to how three syntactic objects, *deny*, *which claim*, and *that John made* are to be combined, ignoring other syntactic objects. Pair-MERGE applies to WS1, yielding WS2, where *that John made* adjoins to *which claim*. The number of accessible elements is two: *deny* and <{*that John made*}, {*which claim*}>. Next, Set-MERGE updates WS2 to WS3, where *deny* is merged with <{*that John made*}, {*which claim*}>. There are three accessible elements, that is, *deny*; <{*that John made*}, {*which claim*}>; and {*deny*, <{*that John made*}, {*which claim*}>}. It is important to note that the number of accessible elements in WS in (i) is smaller than that in (25). Let us consider the derivation in (25) again.

(ii) a. WS1 = [... deny, {which claim}, {that John made}]

b. WS2 = [... {deny {which claim}}, {that John made}] (Set-MERGE)

c. WS3 = [... {deny {which claim}}, <{that John made} {which claim}>] (Pair-MERGE)

First, Set-MERGE applies to WS1, yielding WS2, where *deny* is combined with *which claim*. Next, Pair-MERGE changes WS2 to WS3, where *which claim* undergoes Sideward Pair-MERGE with *that John made*. Note that WS2 and WS3 each contain four accessible elements: WS2 includes *deny*; {*which claim*}; {*deny*, {*which claim*}}; and {*that John made*} and WS3 includes *deny*; {*which claim*}; {*deny*, {*which claim*}}; and <{*that John made*}, {*which claim*}>. Note that WS2 and WS3 in (ii) contain more accessible elements than those in (i). Therefore, RR selects the derivation in (i) as optimal because the number of the accessible elements in WS in (i) is smaller than that in (ii). The derivation in (i) yields the structure in (iii).

(iii) WS = [{<{that John made} {which claim}> {did he later {deny <{that John made} {which claim}>}}}]

In this structure, *he* c-commands *John*, leading to the Condition C violation. Therefore, as long as the derivation in (i) is chosen, we cannot account for the grammaticality of (24b). We suggest that the derivations in (i) and (ii) are not comparable because (i) is a non-convergent derivation, whereas Chomsky (1995: 220-221) argues that economy conditions compare only convergent derivations. Thus, RR should compare only convergent derivations. Since the derivation in (i), which violates Condition C, is non-convergent, RR cannot compare the derivations in (i) and (ii). Therefore, RR cannot choose the derivation in (i) over (ii). I thank Etsuro Shima for bringing up this issue.

18) Head movement as well as late Merge of the relative clause violates the Extension Condition, which requires that Merge always applies to the root node. Kitahara (2019) argues that the members of the ordered pair formed by Pair-MERGE are inaccessible to operations and shows that a cyclic analysis for head movement based on an interarboreal operation of Bobaljik and Brown (1997) proves to be tenable under the framework of MERGE. See Kitahara (2019) for details.

19) Nunes (2001) rules out the derivation such as (26) with recourse to the notion of chain, which is eliminated in the current theory of grammar. See Nunes (2001: Section 3.2.1) for discussion.

20) Dobashi (2018) accounts for c-command invisibility of the bound pronoun in non-restrictive relative clauses, based on the notion of termination of a derivation, which states that a derivation may, but need not, terminate whenever WS contains a single object (as in Chomsky, Gallego, and Ott (2019: 245)). According to Dobashi, the non-restrictive relative clause is a domain of termination of a derivation, which makes the element in the non-restrictive relative clause invisible to the c-command relation. Therefore, the bound pronoun in the non-restrictive relative clause cannot be licensed from outside. We suggest that the adjunct which is a terminate domain undergoes Parallel Pair-MERGE, which makes elements in the adjunct invisible to the c-command relation.

21) I thank Etsuro Shima for raising this issue.

References

Abney, Steven Paul (1987) *The English Noun Phrase in Its Sentential Aspect*, Doctoral dissertation, MIT.

Bobaljik, Jonathan David and Samuel Brown (1997) "Interarboreal Operations: Head Movement and the Extension Requirement," *Linguistic Inquiry 28*, 345-356.

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

Chomsky, Noam (1995) *The Minimalist Program*, MIT Press, Cambridge, MA.

Chomsky, Noam (2000) "Minimalist Inquiries: The Framework," Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik, ed. by Roger Martin, David Michaels and Juan Uriagereka, 89-155, MIT Press, Cambridge, MA.

Chomsky, Noam (2004) "Beyond Explanatory Adequacy," *Structures and Beyond: The Cartography of Syntactic Structures*, Volume 3, ed. by Adriana Belletti, 104-131, Oxford University Press, Oxford.

Chomsky, Noam (2013) "Problems of Projection," *Lingua* 130, 33-49.

Chomsky, Noam (2019a) "Some Puzzling Foundational Issues: The Reading Program," *Catalan Journal of Linguistics Special Issue, 2019*, 263-285.

Chomsky, Noam (2019b) MIT Lectures, April 10th and 12th.

Chomsky, Noam (2019c) UCLA Lectures, April 29th, 30th, May 1st, and 2nd.

Chomsky, Noam, Ángel J. Gallego, and Dennis Ott (2019) "Generative Grammar and the Faculty of Language: Insights, Questions, and Challenges," *Catalan Journal of Linguistics Special Issue, 2019*, 229-261.

Citko, Barbara (2005) "On the Nature of Merge: External Merge, Internal Merge, and Parallel Merge," *Linguistic Inquiry* 36, 475-496.

Dobashi, Yoshihito (2018) "Termination of Derivation and Intonational Phrasing: A Preliminary Study," *Phonological Externalization* volume 3, ed. by Hisao Tokizaki, 9-23, Sapporo University, Sapporo.

Epstein, Samuel D., Hisatsugu Kitahara, and T. Daniel Seely (2012) "Structure Building That Can't be!," *Ways of Structure Building*, ed. by Myriam Uribe-Etxebarria and Vidal Valmala, 253-270, Oxford University Press, Oxford. [Reprinted in *Explorations in Maximizing Syntactic Minimization*, ed. by Samuel D. Epstein, Hisatsugu Kitahara, and T. Daniel Seely, 2015, 155-174, Routledge, New York.

Kitahara, Hisatsugu (2019) "(Head-)Adjunction by Pair-MERGE," paper presented at the workshop "Pair-Merge under MERGE" in the ELSJ 12th International Spring Forum 2019.

Kono, Tsuguyo (2012) *Eigo-no Kankeisetsu* (English Relative Clauses), Kaitakusha, Tokyo.

Lebeaux, David (1988) *Language Acquisition and the Form of the Grammar*, Doctoral dissertation, University of Massachusetts, Amherst.

Lebeaux, David (1991) "Relative Clauses, Licensing, and the Nature of the Derivation," *Syntax and Semantics 25: Perspectives on Phrase Structure: Heads and Licensing*, ed. by Susan D. Rothstein, 209-239, Academic Press, New York.

McCawley, James D. (1981) "The Syntax and Semantics of English Relative Clauses," *Lingua* 53, 99-149.

McCawley, James D. (1998) *The Syntactic Phenomena of English*, 2nd ed., University of Chicago Press, Chicago.

Nunes, Jairo (2001) "Sideward Movement," *Linguistic Inquiry* 32, 303-344.

Nunes, Jairo (2004) *Linearization of Chains and Sideward Movement*, MIT Press, Cambridge, MA.

Ross, John Robert (1986) *Infinite Syntax!*, Ablex Publishing Corporation, Norwood, New Jersey.

Safir, Ken (2019) "The A/A′ Distinction as an Epiphenomenon," *Linguistic Inquiry* 50, 285-336.

Takahashi, Shoichi and Sarah Hulsey (2009) "Wholesale Late Merger: Beyond the A/A′ Distinction," *Linguistic Inquiry* 40, 387-426.

Vries, Mark de (2006) "The Syntax of Appositive Relativization: On Specifying Coordination, False Free Relatives, and Promotion," *Linguistic Inquiry* 37, 229-270.

Kitami Institute of Technology

165 Koen-cho Kitami, Hokkaido, 090-8507

E-mail: tozawata@mail.kitami-it.ac.jp