5.3 What Representation is Left Behind after Transfer?

Section 5.2 mainly discussed what part of the representations is (explicably) sent to the interfaces by Transfer—whether phase edges are left behind as Chomsky proposes or the entire phase is transferred.[10] Then, I explained, by appeal to s-selection and phonetic specification of C, why vP phase edges and embedded CP edges remain while a matrix phase edge is transferred along with the phase head complement. This section further considers the Transfer operation, which plays a central role in Chomsky/Richards deduction. I am specifically concerned with the question of what representation remains in narrow syntax after Transfer has applied--i.e. what is the representational output of transfer-application.

In the following two sections (5.3.1 and 5.3.2), I pursue but ultimately reject two possible answers (and also their entailed problems) to the question of what representation remains in narrow syntax after Transfer applies. In Section 5.3.3, I propose the Label-Copying Transfer system and demonstrate that the proposed system is an optimal way to satisfy all the conditions imposed on syntactic derivations and on representations.

5.3.1 Possibility 1: Everything is Left by Transfer: Full Copy.

The first idea is that all aspects of the NS representation are left behind (fully copied) in narrow syntax, after Transfer applies. In this case, all the representations are (somehow) copied and those copies are sent to the interfaces as illustrated in (44) and (45).

[10] Aspects of this section are based on Obata (2010).

(44) **NS:**
[$_{vP}$ John v [$_{VP}$ bought the book]].
[$_{vP}$ John v [$_{VP}$ bought the book]]
Full VP Copy

→ **PHON:** [$_{VP}$ bought the book]

→ **SEM:** [$_{VP}$ bought the book]

(45) [$_{CP}$ C [$_{TP}$ John T [$_{vP}$ <John> v [$_{VP}$ bought the book]]]]
Full VP Copy

Under this scenario, as illustrated in (44), narrow syntax retains the complete NS representation, and while identical copies are sent to each interface. Thus, the next higher phase head C is introduced retaining the entire VP contents as in (45).

How is the narrow syntax computation to be limited to phases if the entire representation retains after Transfer? Since VP persists in the post-Transfer narrow syntax representation (e.g. (45)), C (and T) should have access to these domains, which render the computational workspace unlimited. To carry out cyclic computation, an independent condition i.e. the Phase Impenetrability Condition needs to be stipulated.

(46) Phase-Impenetrability Condition (PIC)
In phase α with head H, the domain of H is not accessible to operations outside α, only H and its edge are accessible to such operations.
(Chomsky 2000: 108)

Under PIC, C and T are prohibited from searching VP or its contents in (45), which already underwent Transfer. In other words, if the entire representation remains in narrow syntax after Transfer, PIC is additionally necessary in order to block "counter-cyclic" computation. If the entire representation remains after Transfer as in (44), narrow syntax needs to retain representations which are by hypothesis never used in the subsequent computation. In other words, superfluous representations which are in fact syntactically inactive are stuck in narrow syntax with no explanation of the (hypothesized) empirical

fact (or hypothesis) that they are inert once derivation proceeds to a higher phase.[11]

5.3.2 Possibility 2: Nothing is Left: "Tree Pruning"

Another possibility is that Transfer of a phase head complement leaves nothing behind in the NS representation after Transfer, contra (44). That is, Transfer sends the representation itself to the interfaces, not a copy. If this is on the right track, narrow syntax representations partially disappear as a consequence of Transfer as illustrated below:

(47) **NS:**
[$_{vP}$ John v [$_{VP}$ bought the book]]. → **PHON:** [$_{VP}$ bought the book]
→ **SEM:** [$_{VP}$ bought the book]
[$_{vP}$ John v]
Representations are pruned

(48) [$_{CP}$ C [$_{TP}$ John T [$_{vP}$ <John> v]]]

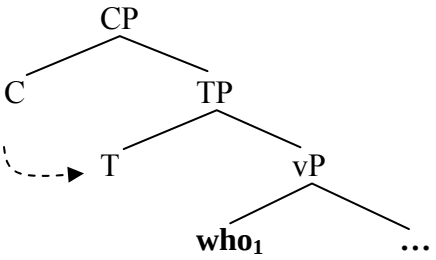
Since, contra (44), VP disappears from narrow syntax as in (48), the higher heads such as C and T have no way to make computational contact with the contents of VP, or with VP itself and this now follows without assuming any independently stipulated principle, i.e. PIC follows, its effects incorporated into the grammatical mechanisms themselves. The reason phase head complements are inaccessible is because they are absent from the NS representations (this arguably being the best way to explain inaccessibility, i.e. absence). Therefore, there is no need to maintain PIC as an independent principle of the grammar to make inter-phasal computation impossible but rather we can have a most natural explanation by saying that syntactic operations or relations can of course involve any

[11] Although it is not clear which view Chomsky adopts, Chomsky (2004) argues based on Nissenbaum (2000) that the distinction between overt and covert movement is made by Internal Merge before or after Transfer. Given this assumption, he implies that at least phonological features are removed from the NS representation as a result of the application of Transfer.

element and only elements existing in the narrow syntactic representation. Furthermore, in (47), unlike the other possibility, narrow syntax does not retain any extra/superfluous representations which are syntactically inert throughout the remainder of the derivation.

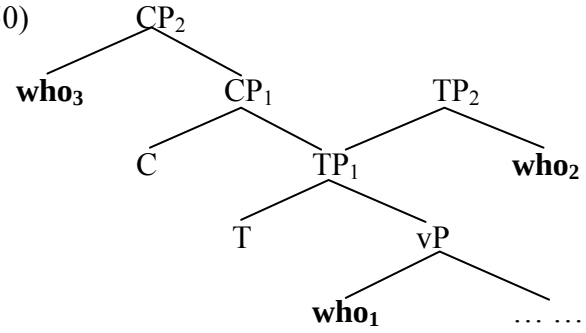
In addition, Epstein, Kitahara and Seely's (2009) analysis implies that this view is supported, (although they do not note the implications of their analysis for the issues under consideration here). According to their argument, Chomsky's (2007, 2008) feature-inheritance operation results ultimately in the creation of a structure with "two peaks"[12] in a derivation for e.g. "I wonder who saw her" as follows:

(49)



Feature-Inheritance from C to T

(50)



Simultaneous Internal Merge of "who" by C and T

Let us focus on the derivation for the embedded CP. Given feature-inheritance, T can work as a probe only after (phi) features are inherited from C as in (49). (See Chapter 2

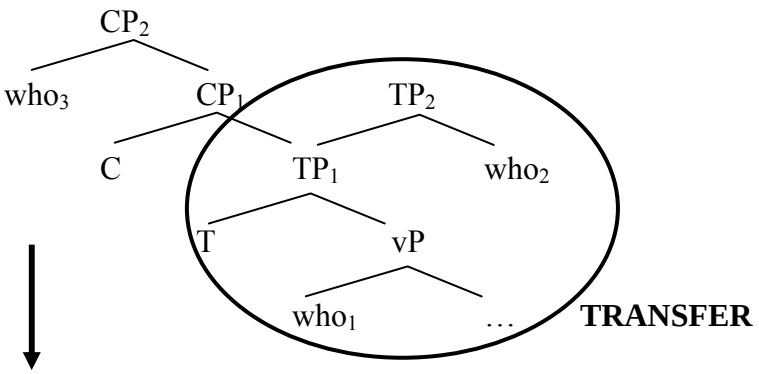
[12] In set theoretic terms, there is set-intersection in (50) i.e. TP1 is a member of >1 set, succinctly, CP1={C, TP_1}, TP2={TP_1, who_2}.

for relevant discussion.) Chomsky (2007, 2008) suggests that in this situation, C and T simultaneously attract the single element "who$_1$" to their edge positions. As illustrated in (50), however, "who$_2$" attracted by T creates a structure with two peaks assuming cyclicity/extension bars "syntactic infixation" of the subject into Edge-TP after C has been merged. Assuming derivational c-command in Epstein et al. (1998) and Epstein (1999), which is defined in (51), neither CP$_2$ nor "who$_3$" c-commands "who$_2$" since there is no derivational relation between them:

(51) *Derivational C-Command:*
X *c-commands* all and only the terms of the category Y with which X was paired/concatenated by Merge or by Move in the course of the derivation.
(Epstein 1999: 329)

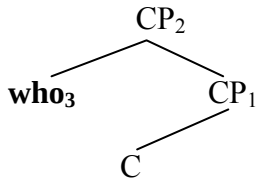
Therefore, it seems to be impossible to decide which projection, CP$_2$ or TP$_2$, is the topmost "root" category necessary for continuing higher derivations (or alternatively they propose a semantic composition failure is indeed by the non-null set interpretation in (50). Epstein, Kitahara and Seely (2009) suggest that by transferring one of the two root projections i.e. TP$_2$ in this case, the "offending" two-peak structure is destroyed and only CP$_2$ survives in narrow syntax as follows, which makes further application of Merge possible.[13] (See Epstein, Kitahara and Seely 2009 for further details.)

(52)



[13] Epstein, Kitahara and Seely (2009) is another analysis to explain why TP, not the entire phase CP, is transferred, although it is not clear to me how two-peak structures undergo reassembly at the interface necessary for global computation such as Condition C.

(53)



If the representation still remains in narrow syntax after Transfer, the two-peak structure still exists and prevents further derivations. That is, their mechanism implies that representations (including one of the two peaks) "entirely disappears" from the narrow syntax by Transfer as illustrated in (53).

For these reasons, it seems to be reasonable to further pursue the possibility that no representation is left behind by Transfer, as in (47). That is, parts of the representation in narrow syntax cyclically "disappear" from narrow syntax. (This would then represent “tree-pruning” as proposed in earlier transformational approaches to deletion.) The next section points out some theoretical problems regarding this possibility and suggests the Transfer Label-Copying system as a general solution.

5.3.3 Label-Copying

If we understand Transfer as a type of "deletion" as discussed in the last section, the elided/transferred part has to be somehow recoverable under the recoverability condition requiring that no information be lost by deletion as discussed most recently in Chomsky and Lasnik (1995). But how is this possible? That is, recoverability seems clearly violated under this eliminative analysis of Transfer (= the nothing-is-left option). Furthermore, there is another problem regarding how the No Tampering Condition (NTC) is satisfied given this approach:

(54) No Tampering Condition
Merge of X and Y leaves the two SOs (= syntactic objects, MO) unchanged.
(Chomsky 2008: 138)

If NTC is extended so that it constrains not only the operation Merge but also the representations derived, a Transfer operation, which obliterates parts of the representation, always breaks up the relation between a phase head and its complement, a relation which is built by Merge. In (47) and (48), the sister relation of v-VP is broken by Transfer by "deleting" VP. Therefore, the current system violates Generalized-NTC in this sense.

How is "Transfer-as-deletion" executable at all in grammars incorporating both Recoverability of Deletion and the Generalized-NTC? To render Transfer-deletion compatible with both of these principles I propose, the following Label Copying Transfer system:

(55) *Label-Copying Transfer*
The transferred phase head complement leaves a copy of only its label when it undergoes Transfer.

Transfer then leaves only the label of the phase head complement behind in the narrow syntax. Given this system, let us see below how the derivation demonstrated in (47) is slightly altered:

(56) **NS:**
$[_{vP}$ John v $[_{VP}$ bought the book]] → **PHON:** $[_{\mathbf{VP}}$ bought the book]
→ **SEM:** $[_{\mathbf{VP}}$ bought the book]

VP-Transfer and Label-Copying
$[_{vP}$ John v $[_{\mathbf{VP}}$]]

(57) $[_{CP}$ C $[_{TP}$ John T $[_{vP}$ <John> v $[_{\mathbf{VP}}$]]]

When VP is transferred in (56), only the copy of its label is left behind in the narrow syntax representation while PHON and SEM each receive the identical copies of the entire VP with its complete internal representation. Notice that PIC is still deduced in this

system since the elements within VP do not exist in narrow syntax. Now, let us consider the problems regarding Generalized-NTC and the recoverability condition mentioned earlier. First, how is the recoverability condition satisfied? Since the identical copy at SEM and PHON retains all the information of VP, the identical (but "empty") label VP in narrow syntax can later recover its internal structure at the interfaces as follows:

(58) **NS:** **PHON/SEM**:

Step1: [$_{vP}$ John v [$_{VP}$ bought the book]]

VP-Transfer [**$_{VP}$** bought the book]

Step2: [$_{vP}$ John v [**$_{VP}$**]]

Step3: [$_{CP}$ C [$_{TP}$ John T [$_{vP}$ <John> v [**$_{VP}$**]]]

Step 4: *CP-Transfer* [$_{CP}$ C [$_{TP}$ John T [$_{vP}$ <John> v **[$_{VP}$ bought the book]**]]

When the empty label-copy "VP" is transferred as a part of CP (=Step3), the identical copies which were transferred last time are re-inserted into the empty label at the interface. That is, at the interface, by refilling the designated labels, the transferred pieces are re-assembled--as they must be--in order to generate complete sentential (recursive, unbounded) representations at the interface levels. In the above case, the internal structure of VP is re-inserted into the empty VP, which is transferred as a part of TP. Here, the copied labels serve as "guideposts" and the (temporarily) lost information from narrow syntax (i.e. VP-contents) is recovered before reaching the interfaces. What motivates the re-assembly operation? By observing the principle of Full Interpretation, labels lacking internal contents must be filled by its contents, otherwise the representation is uninterpretable. Therefore, the recoverability condition can be satisfied under this approach. The second problem, recall, concerned Generalized-NTC: Transfer as a type of deletion destroys a sister relation created by Merge of a phase head and its complement.

Again, this is no longer problematic for the current system because the copied label is left behind in narrow syntax as in (56), so that the relation of v-VP never changes, and so satisfying Generalized-NTC. Both of the problems are solved under the present approach, while PIC--as an independent stipulation--can be abolished. In this sense, the proposed system based on (55) can be a natural implementation of Transfer observing conditions imposed on syntactic computation. (See Narita 2009 for another approach to the issue of how Transfer affects narrow syntactic representations.)

The derivational procedure illustrated in (58) is the main idea I suggest in this chapter. Reconsidering the two possibilities reviewed in the last two sections (i.e. the everything-is-left approach vs. the nothing-is-left approach), the idea proposed here is a "compromise" located "in between" in the sense that there is a label copy left behind but its contents disappear. With respect to NTC issues, I argued that Transfer breaks sister relations between e.g. v and VP/C and TP violating Generalized-NTC. This is why there is a need to leave a copied label behind. Recall that Section 5.2 tackled the issue of what mechanisms render Transfer of just the phase head complement possible in embedded phases but not in matrix phases. The mechanism can be induced from satisfaction of s-selection/clause-typing by appeal to arbitrary selection of the two derivational options: feature-inheritance or Agree. Under this system, the asymmetries between root phases and embedded phases are explained. Considering these matrix/embedded asymmetries issues in the context of "Label-Copying Transfer", only in vP and embedded CP, copied labels remain in narrow syntax but not in matrix CP. Since the edges of vP and embedded CP are left behind because of s-selection, Transfer sends VP for vP and TP for embedded CP to the interfaces. As mentioned above, Transfer of phase head complements breaks

sister relations. To prevent it, the copied label of VP/TP is left behind. On the other hand, matrix CP is not s-selected by any element. This is why the entire CP is allowed to undergo Transfer in this case under the reformulated Transfer system. Since the entire CP is transferred, Transfer does not break any of the sister relations. Therefore, the CP label is NOT left in this case, which observes Generalized-NTC. In other words, matrix CP phases leave neither its edge nor its copied label in narrow syntax leading to no superfluous representations in narrow syntax. The proposed label copy system is compatible with the Transfer system reformulated in Section 5.2.

5.4 On Reassembly: How can Syntactic Objects "Bigger than a Phase" Undergo Internal Merge?

This section further examines the Transfer Label Copy system proposed in Section 5.3. Recall that in the proposed system, Transfer creates a label temporarily in narrow syntax and by re-filling the internal structure with the identical copies (= re-assembly), the representation can come to consist of legible objects at the interfaces satisfying the principle of full interpretation. The cases to be examined here seem to empirically and independently motivate the proposed re-assembly system based on label-copying. Consider e.g.:

(59) a. Mary believed the claim that John bought the book.
b. **Whose claim that John bought the book** did Mary believe _?
(60) a. I think John will buy the book.
b. **John will buy the book**, I think _.
(61) a. It was denied that John bought the book.
b. **That John bought the book** was denied _.

In these cases, the underlined phrases--each bigger than a phase--undergo Internal Merge (IM) which affects both linearization and semantic interpretation. That is, in (59b) it must

somehow be the case that the entire DP "whose claim that John bought the book" undergoes successive cyclic movement to the edge of the matrix CP before Transfer applies to it, or any of its sub-terms. But in (59b), for example, at the derivational point at *whose claim [$_{CP}$ that [$_{TP}$ John bought the book]]* is built, the TP is transferred. Now, suppose that there is nothing left behind in narrow syntax after Transfer, not even a label copy. Under that assumption, this TP no longer bears phonological or semantic features, since these have already been transferred, immediately after this TP was merged with C (= "that"). Consequently, by "the time" IM applies to the entire wh-DP underlined in (59b), this TP inside that DP, has already had its phonological and semantic features removed since Transfer already applied to this TP. The resulting output of such wh-DP-fronting at the phonological component is then predicted to be (62): i.e. the TP transported by wh-movement, is incorrectly predicted to lack PHON-features, predicting the incorrect linearization:

(62) The Representation at PHON of (59b)
***Whose claim that** did Mary believe **John bought the book**?

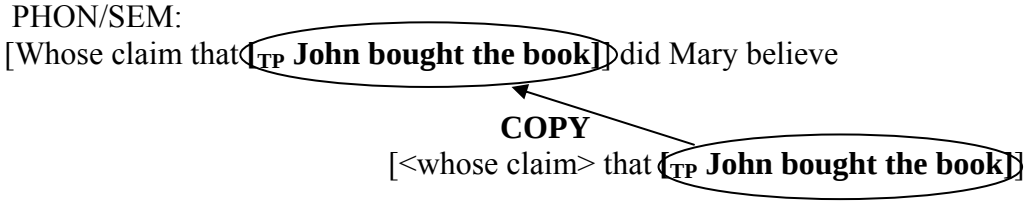
Thus, given the standard cyclic Transfer system, the cornerstone of phase-based derivation, it seems impossible to derive any sentences in which constituents bigger than a phase undergo IM, suggesting the possibility of infinite undergeneration.

Regarding this issue, the alternative system proposed in Section 5.3, incorporating label-copying and re-assembly predicts that the wh-DP moved to the edge of the matrix CP contains the empty label TP. After all the representations are transferred from narrow syntax, (63) is obtained:

(63) *PHON&SEM Representations*
[Whose claim that **[$_{TP}$]**] did Mary believe [<whose claim> that
[$_{TP}$ John bought the book]]

Recall, "before" the entire wh-DP underwent IM, TP within this DP was transferred. Based on the Label-Copying Transfer system stated in (55), the empty label is left behind after Transfer, so that the DP at the matrix Spec-CP includes the empty label "TP" whose internal structures are replicated by the identical copies already transferred satisfying the recoverability condition. However, notice that a representation containing an empty node causes violation of the full interpretation principle, (an interface Bare Output Condition), so that (63) is not a legitimate interface representation itself. That is, re-filling/re-assembly is required as a last resort operation as sketched in the last section. The crucial difference between accounts with and without label-copying is that the label-copying system creates an "imperfect"/illegitimate representation such as (63) (presumably crashing, but perhaps gibberish, or both) as the result of the application of Transfer. Therefore, the procedure rendering illegitimate objects legitimate can be invoked as a last resort. In the case of (63), therefore, the empty label TP within the wh-DP occupying the edge of the matrix CP is "re-filled" by copying the identical TP copy of the post-verbal position at the interfaces:

(64) PHON/SEM:



Again, the copied label TP left behind in narrow syntax and the identical copies at PHON/SEM serve as the minimal guideposts necessary for re-assembly at the interface. Of course, re-assembly at the interface is necessary, given phase-based derivation, even independent of the issues raised here. As a result, the representation is regarded as a legitimate object at the interfaces. As mentioned in the previous section, the operations

involved in label-copying are not stipulated but rather are induced from independently motivated conditions such as Recoverability, FI and Generalized-NTC. That is, the proposed system is arguably a natural implementation of Transfer satisfying those conditions and it can also explain cases in which phrases bigger than a phase undergo IM. Also, from a cross-linguistic point of view, movement of phrases bigger than phases is not limited to the English cases discussed here but also exists in other I-languages. For example, scrambling in Japanese can also target CP as follows:

(65) a. Taro-ga Jiro-ni [Hanako-ga ringo-o tabe-ta to] it-ta.
Taro-Nom Jiro-to Hanako-Nom apple-Acc eat-Past C tell-Past
"Taro told Jiro that Hanako ate apples."

b. [**Hanako-ga ringo-o tabe-ta to]** Taro-ga **t** Jiro-ni it-ta.
Hanako-Nom apple-Acc eat-Past C Taro-Nom Jiro-to tell-Past
"Taro told Jiro that Hanako ate apples."

In the above example, the entire CP undergoes scrambling to the edge of the matrix CP. That is, this phenomenon also requires some sort of re-assembly procedure of transferred pieces. The system proposed in this section presents one of the possible ways to render phasal reassembly possible in a form observing independently motivated conditions such as (Generalized-)NTC, Recoverability and full interpretation.

Finally, it is noteworthy that the current approach is inconsistent with Collins (2002) analysis suggesting the possible elimination of labels. In the analysis presented in this section, labels play a central role especially in reassembling transferred pieces and also in rendering it possible that Transfer is applied in conformity with Generalized-NTC--without breaking a sister relation.

5.5 Summary

Throughout this chapter, I have specifically examined detailed mechanisms of the