Transfer and Dynamic Access*

Jae-Young Shim

Shim, Jae-Young. 2022. "Transfer and Dynamic Access." X-XX. In minimalist theorizing, the operation Transfer has long been assumed to contribute to reduction of computational complexity by removing from the derivational workspace the syntactic objects in the phase interior. This paper, however, addresses some critical problems with this 'eliminative' Transfer resulting from its stipulative nature. To resolve the addressed problems, this paper adopts the suggestions in Chomsky (2021, Forthcoming) and presents an alternative called Dynamic Access (DA) according to which the interpretive systems directly access the derivation at the phase level.

Key Words computational efficiency, Transfer, Phase Impenetrability Condition, Dynamic Access, interfaces, interpretation

1. Introduction

The notion of computational efficiency has played a crucial role in minimalist theorizing (Chomsky 1995 *et seq.*). The operation Transfer is one such efficiency-driven apparatus in the sense that it contributes to reducing computational complexity by periodically sending a certain portion of the generated structure over to each interface, i.e. the Sensorimotor (SM) systems and the Conceptual-Intentional (CI) systems.

Two major proposals have been made in the literature with respect to how Transfer operates. One is what I call an 'eliminative' Transfer (Chomsky 2001 *et seq.*) by which the phase-head complement, or the phase-interior, is literally eliminated from the derivational

* I would like to thank ...

workspace upon completion of a phase. The other is what we may call a 'closing-off' version of Transfer (Chomsky *et al.* 2019) by which the phase-interior, while remaining in the derivation, gets closed-off from further syntactic computations once a phase is completed.

The present paper addresses some conceptual and empirical problems with the conventional Transfer in terms of computational efficiency, Transfer of root clause, and a redundancy between Transfer and the Phase Impenetrability Condition (PIC). To resolve the addressed issues, this paper adopts the suggestions in Chomsky (2021, Forthcoming) and argues that the interface systems directly access the derivational workspace for interpretation. One major consequence of this proposal is that we can eliminate from the grammar both the operation Transfer and the condition PIC.

The paper is organized as follows. Section 2 reviews the two standardly assumed versions of Transfer, one proposed in Chomsky (2001) and the other suggested in Chomsky *et al.* (2019). Section 3 discusses some conceptual and empirical problems with the standard assumptions arising from the stipulative nature of the operation Transfer. Section 4 presents an alternative and shows how the addressed problems can be resolved. Section 5 concludes the paper with a brief suggestive discussion of future research.

2. Transfer: the Two Versions

The operation Transfer[1] has long been assumed to target only the syntactic objects (SOs) within the phase-interior, sending them over to each interface (Chomsky 2001 *et seq.*).[2] A natural corollary of this assumption is that the transferred SOs become no longer available to further syntactic computations, while the remaining SOs, i.e. the phase head and whatever SOs in its Spec position, are. Consider (1) to see in more detail what SOs Transfer targets and how the derivational workspace is modified after application of

[1] The operation sending generated SOs to the interfaces was initially referred to as 'Spell-Out' (Chomsky, 1995) which was then assumed to strip away only the phonological features from the generated SOs. Later in Chomsky (2001 *et seq.*), however, Spell-Out is incorporated into the more general operation Transfer.

[2] The target domain of Transfer is modified to the R(oot)-interior in Chomsky (2015) since in the framework R rather than the phase head itself is hypothesized to act as a phase head once R inherits what Chomsky calls 'phasehood' from *v**.

Transfer (EA = External Argument; IA = Internal Argument; ← 'applies'; ⇨ 'returns'; the distinction between *v* and *v** is ignored for expository purposes).

(1) {$_{v\text{P}}$ EA, {*v*, {V, IA}}} ← Transfer ⇨ {EA, *v*}

Suppose that all the required operations (e.g. Agree) have completed in the *v*P phase in (1). The operation Transfer will then apply, sending off the SOs within the interior (i.e. V and IA) to each of the interfaces.[3] As a result, EA and the phase head *v*, collectively referred to as the 'edge' (of a phase), are left behind in the derivation as shown in (1), being able to participate in whatever relevant operations at the next phase. Let us call this an 'eliminative' Transfer, or E-Transfer, as certain SOs are literally eliminated and thus disappear from the derivational workspace.

Obata (2010), however, points out some empirical problems with this eliminative version of Transfer. Consider (2) (Throughout the paper, irrelevant structures are omitted or simplified.)

(2) a. whose claim that John bought the book did Mary believe?
b. [$_{\text{CP}}$ that [$_{\text{TP}}$ John bought the book]] ← Transfer ⇨ [$_{\text{CP}}$ that]
c. [C [$_{\text{TP}}$ Mary believes [$_{\text{NP}}$ whose claim [$_{\text{CP}}$ that]]]]
d. [$_{\text{CP}}$ [$_{\text{NP}}$ whose claim [$_{\text{CP}}$ that]] did Mary believe]
e. *[whose claim that] did Mary believe [John bought the book]

At some point in the generation of (2a), the embedded CP phase in (2b) is constructed. Once this CP phase is completed, its interior, TP, gets transferred and thus eliminated, leaving behind only the phase head C (= *that*) in the derivational workspace. Derivation continues to form the matrix CP as shown in (2c) and at some point of the derivation, [$_{\text{NP}}$ *whose claim* [$_{\text{CP}}$ *that*]] moves to the matrix Spec-CP as illustrated in (2d). Notice that the resulting structure in (2e) is not what we intend to generate where some portion of the NP, i.e. [*whose claim that*], is spelled out in the matrix Spec-CP, while the rest, i.e. [$_{\text{TP}}$ *John bought the book*], is spelled out in-situ. This unwanted result, however, is unavoidable

[3] If one assumes V-to-v raising and raising of IA, the transferred SOs would be the lower copy of V and of IA.

under E-Transfer as [$_{\text{TP}}$ *John bought the book*], which belongs in the interior of the embedded CP phase, ought to be transferred to the interfaces upon completion of the embedded CP phase before [$_{\text{NP}}$ *whose claim that*] raises.

In order to solve this problem of undergeneration pointed out in Obata (2010), Chomsky *et al.* (2019) revise the nature of Transfer. They argue that the phase interior is not (literally) 'eliminated' by the operation Transfer from the derivational workspace but rather it is rendered opaque to further syntactic manipulations.[4] Call this a 'closing-off' version of Transfer[5] (or C-Transfer) to distinguish it from its eliminative counterpart, i.e. E-Transfer. In this revised model, generation of (2a) proceeds as illustrated in (3) (the transferred SOs are shadowed to highlight the fact that they remain in the derivation).

(3) a. whose claim that John bought the book did Mary believe?
   b. [$_{\text{CP}}$ that [$_{\text{TP}}$ John bought the book]] ← Transfer
      ⇨ [$_{\text{CP}}$ that [$_{\text{TP}}$ John bought the book]]
   c. [$_{\text{TP}}$ Mary believes [$_{\text{NP}}$ whose claim [$_{\text{CP}}$ that [$_{\text{TP}}$ John bought the book]]]]
   d. [$_{\text{CP}}$ [$_{\text{NP}}$ whose claim [$_{\text{CP}}$ that [$_{\text{TP}}$ John bought the book]]] did Mary believe]
   e. whose claim that John bought the book did Mary believe?

The notable step is (3b), where, despite having been transferred, the phase interior [$_{\text{TP}}$ *John bought the book*] is not eliminated but remains in the derivation. Consequently, this TP can move along when [$_{\text{NP}}$ *whose claim* ...] raises to the matrix Spec-CP. The intended structure can thus be generated as shown in (3d,e).

I finish this section by presenting a notably different prediction that the two versions of Transfer make with respect to labeling. Consider (4a) and (4b).

(4) a. {$_{\alpha}$ EA, {vP *v*, {V, IA}}} ← E-Transfer ⇨ {$_{\alpha}$ EA, *v*}
   b. {$_{\alpha}$ EA, {vP *v*, {V, IA}}} ← C-Transfer ⇨ {$_{\alpha}$ EA, {*v*, {V, IA}}}

[4] It is open for debate, however, whether the closed-off domain is immune to syntactic operations in general including e.g. Internal Merge (IM) or can be selectively available to certain operations (e.g. Labeling, Agree).

[5] In fact, the idea that the transferred SOs remain in the derivation was already hinted at in Chomsky (2013), where he suggests that 'while the interior of a phase is immune from further changes, it does not disappear.'

As illustrated in (4a), E-Transfer strips away both V and IA from the derivational workspace, leaving behind EA and *v*. Note that SO α, which was unlabelable before Transfer due to its being an XP(NP)-YP(*v*P) structure[6], turns into a labelable structure after Transfer as *v* has now become the unique head. This implies that under E-Transfer, the labeling-driven account for EA's raising to Spec-TP may lose its merit.[7] Under C-Transfer as shown in (4b), however, a labeling-driven argumentation for raising of EA seems to be debatable depending on how one analyzes the nature of the closed-off SOs. That is, if one regards them to be completely inactive syntactically, the SO α will be able to count as a labelable structure without raising of EA; otherwise, α may well be considered unlabelable so that raising of EA will be required.

3. Problems

Focusing mainly on E-Transfer, this section discusses four potential problems with it: computational efficiency, a redundancy between Transfer and PIC, Transfer of root clause, and complication of the operation Transfer.

3.1 Computational Efficiency and the Unit of Transfer

The rationale behind the phase-based Transfer derives from one of the conceptual pillars of the Minimalist Program, i.e. computational efficiency. As discussed in the previous section, the operation Transfer contributes to minimizing computational load by periodically reducing the number of SOs to compute in the derivation. Note, however, that if computational efficiency is the critical, if not sole, driving force for postulation of the

[6] The Labeling Algorithm (LA) developed in Chomsky (2013, 2015) is designed to locate the closest head in a given syntactic object to determine its label. If more than one such head are found in the structure (e.g. X and Y in {{$_{XP}$ X, ZP}, {$_{YP}$ Y, WP}}), LA cannot determine a unique label due to the arising ambiguity unless there is Agree between e.g. X and Y or one of the merge-mates (i.e. either XP or YP) has moved out.

[7] A labeling-based account of raising of EA may still be maintained if one assumes with Chomsky (2015) that T is weak to serve as a label. See Mizuguchi (2017), Murphy and Shim (2020), and Shim (2018) for discussion of potential problems with the weak T assumption.

operations such as Transfer, we can conceive of alternative units of Transfer that are arguably more efficient when it comes to computational burden. Consider (5).

(5) a. {EA, {*v*, {$_{\text{VP}}$ V, IA}}} ← E-Transfer ⇨ {EA, *v*}
    b. Alternative I: Transfer of both *v* and VP ⇨ {EA}
    c. Alternative II: Transfer of the entire phase ⇨ ∅

Computational complexity gets further reduced as we go from (5a) to (5c) due to the decreasing number of what remains in the derivational workspace. A question that thus immediately arises is: why do we nonetheless opt for the standard Transfer as described in (5a) that seems to reduce computational complexity the least? An answer in Chomsky (2007) goes as follows: 'since the specifiers of a phase head and the phase head itself may be raised in later phases.' To put it differently, if the phase head and/or the SOs in its specifier were to be part of the transferred unit (as illustrated in (5b) and (5c)), no head movement of the phase head to T or raising of EA to Spec-TP would ever be possible. In fact, potential issues with Alternative I and II are not just limited to further raising of the eliminated SOs. Consider (6).

(6) a. {EA(=NP), {*v*, {VP V, IA}}}
    b. Alternative I: Transfer of both *v* and VP ⇨ {NP}
        Merge(T, NP) ⇨ {T, NP}
    c. Alternative II: Transfer of EA, *v* and IA ⇨ ∅
        Merge(T, ∅)

Take EA in (6a) to be the typical element, i.e. an NP. If Transfer is so designed to eliminate both the phase head and its interior VP as shown in (6b), NP will be the only SO that remains in the derivation for further computation. Suppose, then, that T is introduced after Transfer, undergoing Merge with this NP. This merger of T and NP, however, would lead to a violation of the selectional requirement of T provided that T typically selects for something verbal, not nominal (see section 5 for more discussion on this). Alternative II in (6c), where no SO remains after Transfer, is ruled out for the same reason, i.e. nothing can satisfy no selectional requirement.[8] It thus may seem that among the three conceivable units of Transfer in (6), the standardly assumed unit (i.e. only the phase-interior) is the only feasible option to lead to a convergent derivation, although it is not the most efficient version of Transfer. I will show in section 4, however, that Alternative I is indeed a necessary unit of Transfer without causing the problem of selection.

3.2 A Redundancy between Transfer and PIC

Transfer is not the only computational burden-reducing apparatus in minimalist theorizing. The condition commonly referred to as the Phase Impenetrability Condition (PIC) is another such theoretical device that is designed to reduce computational complexity. Incidentally, PIC is hypothesized to target the same domain as Transfer. Consider (7), where PH = phase and α-H = edge.

(7) a. PH = $[_{HP}$ α [H β]]
b. Phase Impenetrability Condition (modified from Chomsky 2001)
At the phase ZP containing phase HP, β is not accessible.

Compare, first, PIC as defined in (7b) with the standard E-Transfer in (8).

(8) E-Transfer
PH = $[_{HP}$ α [H β]] ← Transfer (of β) ⇨ [α H]

Notice that once the phase HP is completed, E-Transfer ‘literally’ eliminates β from the derivational workspace as illustrated in (8). The question that immediately arises is, what is it, then, that PIC in (7) blocks access to? In other words, do we need to, or can we even block access to the SOs that have already disappeared and thus no longer exist in the derivation? The closing-off version of Transfer (i.e. C-Transfer) is no less free from this issue. Consider (9).

[8] See, however, Groat (2015) and Ke (2017) where the authors independently argue for a full phase-based Transfer.

(9) C-Transfer

PH = [$_{HP}$ α [H β]] ← Transfer (of β) ⇨ [α [H, β]]

By definition, C-Transfer renders whatever SOs in β inaccessible to further syntactic manipulation.[9] Then, again, do we need to redundantly block access to the SOs that have already become inaccessible by Transfer?

3.3 Root Clause and Transfer

Consider (10), where *v*-to-T raising is ignored for expository purposes.

(10) a. what did John buy?

b. [$_{vP}$ what [John *v* [$_{VP}$ buy <what>]]] ← Transfer of VP[10]

⇨ [$_{vP}$ what [John *v*-buy]]]

c. [$_{CP}$ what C [$_{TP}$ John T [$_{vP}$ <what> [<John> *v*-buy]]]] ←Transfer of TP

⇨ [$_{CP}$ what C]]]

At some point in the derivation of (10a), the *v*P phase is generated and its interior, VP, gets eliminated via Transfer as illustrated in (10b). Derivation continues to build the CP phase in (10c), the completion of which leads to Transfer of the interior TP, leaving behind, as shown in (10c), what does not belong to the interior, i.e. the phase head C and *what* in its Spec. Notice, however, that unlike the lower *v*P phase, we have no additional or higher phase (head) that can activate Transfer to send off the remaining C and *what* to the interfaces. In other words, the CP phase is the final stage of the derivation in (10). Consequently, C and *what* end up not being able to be transferred to the interfaces, which in turn means that they are not able to receive necessary phonological and semantic interpretations.[11] In fact, this problem of having non-transferred SOs in the root CP has

[9] As mentioned in note 3, however, whether the inaccessibility is complete or partial is open for debate.

[10] See Chomsky (2015) where he argues that ‘what’ moves to Spec-VP instead of to the outer Spec-*v*P.

[11] Note that the problem of non-transferred SOs is not just limited to *wh*-constructions.

already been noted in Chomsky (2001: 108) where he writes: "S-O [i.e. Transfer] must be able to spell out PH[ase] in full, or root clause would never be spelled out." Yet the question remains as to how, i.e. other than stipulating a rule designed only for root clauses, what general principle or mechanism can we refer to for Transfer to exceptionally target 'all' the SOs in the root CP?[12]

3.4 Complication of Transfer

Within the minimalist tradition, each lexical item is assumed to consist of a set of phonological (PHON) and syntactic/semantic (SYN/SEM) features, as illustrated in (11).

(11) a. boy : PHON = [+bilabial, ...]; SYN/SEM = [$i\phi$, ...]
b. T: PHON = [?][13]; SYN/SEM = [$u\phi$, ...]

PHON features are, by definition, interpretable only at SM, whereas SYN/SEM features are interpretable only at CI (but see below for the CI-uninterpretable SYN/SEM features). What this implies is that PHON and SYM/SEM feature sets of *boy* in (11a) ought to be separated out at some point in the derivation, so that each set can be handed over to its relevant interface; otherwise, *boy* cannot be given its proper phonological and syntactic/semantic interpretations. Let us, then, assume this is the case, i.e. Transfer not only has the ability to transmit the relevant SOs to the interfaces but in so doing, it can also distinguish between PHON and SYN/SEM features of an SO, sending the former to SM and the latter to CI. What Transfer can do (or is required to do), however, does not stop here when we consider (11b).

Declarative sentences (e.g. *John likes Mary*) as well end up having non-transferred C as long as one assumes every sentence is CP.

[12] See also Obata (2010) where the author hypothesizes such an 'exceptional' rule she calls the 'Root Transfer Hypothesis' according to which full phase Transfer is exceptionally allowed in the root clause.

[13] The question mark for T's PHON feature content reflects the unclarity arising from Chomsky's (2008) claim that the ϕ-features of T are crucial to phonetic realization of T. In other words, the PHON feature content of T may be null if the ϕ-features of T are enough to identify or determine its phonetic realization.

The ϕ-features of T have long been assumed to be CI-uninterpretable despite their being grouped together with other CI-interpretable SYN/SEM features (e.g. [tense]). What, then, would be necessary to guarantee that nothing but CI-interpretable features is handed over to CI? Chomsky (2008) suggests that the (CI-uninterpretable) ϕ-features of T are deleted before reaching CI, while they should survive to SM if they have phonetic realization. What this implies is that Transfer can not only distinguish between PHON and SYN/SEM features but also know which (SYN/SEM) features should be selectively sent to SM, but not to CI. This multi-tasking nature of Transfer not only complicates the operation itself but can also be a potential burden to the minimalist spirit to minimize the content of UG.

Below is a summary of the problems discussed so far:

(12) *Problem 1*: Unit of Transfer (i.e. why the phase interior only?)
Problem 2: Redundancy between Transfer and PIC
Problem 3: Transfer of Root Clause (i.e. non-transferred matrix C and its Spec)
Problem 4: Complication of Transfer (i.e. Separation/Distinction)

4. Solution

4.1 Dynamic Access and Conditions

Chomsky (2021, Forthcoming) presents a more refined minimalist framework in which he suggests that 'postulation of the interface levels is superfluous; it is enough to say that extra-linguistic systems access derivation.' What this suggestion implies is that in this revised framework, the conventional interpretive systems, i.e. SM and CI, are no longer viewed as a 'passive' recipient standing by for relevant information to be delivered but rather they are regarded as an 'active' system directly accessing the derivation.[14] This shift in perspective on the nature of the interfaces, on the one hand, and the way interpretation

[14] Chomsky (2021) also suggests that 'access to the syntax by language-external systems can in principle take place at any stage of the computation.'

proceeds on the other, offers some advantages to resolve the problems with Transfer discussed in the previous section. I hence adopt this perspective and call it, for convenience, 'Dynamic Access (DA)' defined in (13).

(13) **Dynamic Access**
Interpretive systems (directly) access the derivation for interpretation at the phase level.

Once we adopt a DA-based interpretive system, postulation of the operation Transfer loses its merit because interpretation of SOs in this DA-based model need not be mediated by an independent (additional) operation but by definition is carried out through a direct access to the derivation by the interpretive systems.

As will be discussed in more detail in section 4.2, our DA-based model can immediately offer a more principled solution to the complication problem of Transfer (*Problem 4* in (12)) since it is more intuitive and reasonable to hypothesize that each interpretive system (including the conventional SM and CI) can only interpret what it is supposed to than to stipulate that Transfer, which can hardly be regarded as an 'interpretive' operation, can somehow make such distinctions (between SM-interpretable and CI-interpretable counterparts on the one hand and between CI-interpretable SYN/SEM features and CI-uninterpretable counterparts on the other).

Let's now turn to two conditions we will assume.

(14) **Accessibility Condition (AC)**
No syntactic access is allowed to 'fully' interpreted SOs.

(15) **Condition on Full Interpretation (CFI)**
An SO can be 'fully' interpreted iff it bears no unvalued features.

Note that neither (14) nor (15) is stipulative. In fact, both naturally follow when we consider the nature of interpretation. Suppose, for instance, that an SO has no unvalued features and hence is 'fully' interpreted by CI at the time of its access. What this means in the conventional terms is that the given SO has been sent to CI and thus it now belongs to CI. Consequently, it follows that the syntax can no longer access the SO.

4.2 Derivations and Solutions

With the mechanism of Dynamic Access in (13) and the two natural conditions, i.e. AC and CFI in (14) and (15), respectively, let us now consider (16) to see how our DA-based approach can replace the conventional Transfer and how it can resolve the addressed problems with the conventional Transfer. For expository simplicity, I confine our discussion to the interpretation by CI. Consider (16), where syntactic inaccessibility of SO is indicated by outline font.

(16) a. John likes Mary

b. vP = {John$_{\text{[Case:]}}$, *v*, like, Mary$_{\text{[Case:Acc]}}$} ← CI

⇨ = {John$_{\text{[Case:]}}$, *v*, like, Mary$_{\text{[Case:Acc]}}$}

c. CP = {C, John$_{\text{[Case:Nom]}}$, T, <John$_{\text{[Case:Nom]}}$>}} ← CI

⇨ = {C, John$_{\text{[Case:Nom]}}$, T, <John$_{\text{[Case:Nom]}}$>}}

At some point in the derivation of (16a), the *v*P phase in (16b) is generated. Upon completion of this *v*P phase, CI inspects all the SOs within the phase for interpretation. In so doing, it will notice that unlike *Mary*, EA *John* bears a feature that is uninterpretable to it, namely, its unvalued Case feature.[15] Consequently, *John* cannot be fully interpreted by CI under CFI in (15) and hence remains accessible to further syntactic computations, whereas *Mary* (as well as *v* and *like*) is rendered no longer accessible (i.e. it gets transferred in the conventional terms) as it bears no unvalued features (i.e. AC in (14)). Notice that in our DA-based system the unit of transfer need not be stipulated to be (only) the phase-interior because it naturally follows from the possibility of full interpretation of an SO by the interface system. Hence, *Problem 1* in (12), i.e. the stipulative nature of the unit of Transfer, is resolved. Further, our DA-based system does not require a (stipulative, redundant) condition such as PIC to preclude (syntactic) access to the transferred SOs; the inaccessiblity of an SO naturally follows from its full interpretation. Hence, the redundancy problem (i.e. *Problem 2*) in (12b) does not arise in our system. Additionally, *Problem 4* in

[15] I assume here that structural Case is a feature that NPs inherently bear. See, among others, Chomsky (2001) where the author hypothesizes that structural Case is a reflex of agreement.

(12d), i.e. the complexity problem of a multi-tasking nature of Transfer operation, does not also arise in our system because which feature should be sent to which interface is determined not by a syntactic operation (e.g. the conventional Transfer) but solely (and naturally) by the interpretive system.

Let us now consider the next step illustrated in (16c) to see how our system can handle the problem of transfer of root clause, i.e. *Problem 3* in (12c). Once the *v*P phase is completed, derivation continues to construct the next phase CP, in which the Case feature of *John* is valued via agreement between T and *John*. Upon completion of this CP phase, CI will again inspect all the SOs as illustrated in (16c). Notice, however, that unlike in the *v*P phase, no SOs in CP bear unvalued features. In other words, all the SOs in this CP can be fully interpreted by CI. Consequently, there arises no issue of transferring C (and whatever SOs in its Spec) in our DA-based system.

I would like to finish this section with a possible solution to the selection problem mentioned in passing in relation to *Problem 1*. That is, given that once the *v*P phase is completed, T is introduced and will undergo merger with (the remaining) EA, how can one justify such a merger between the 'nominal' EA and T requiring something 'verbal'? To answer this question, I adopt Chomsky's (2021) suggestion below ((17b) is modified from Chomsky 2021b: 34):

(17) a. John arrived yesterday and met Bill
b. John arrives every day at noon and met Bill yesterday
c. *John buys and sell books

(17a,b) show that tense need not be shared in conjuncts, while (17c) shows that agreement or ϕ-features must be. Chomsky (2021), based on these examples, reincarnates INFL to replace the conventional (finite) TP projection with tense being a feature of *v*. If this is on the right track, the clausal architecture would look something as below:

(18) INFL $\{_{vP}$ EA ... $\}$

Note that this shift in the clausal architecture allows us to resolve the aforementioned problem of selection, i.e. the merger can now be seen as not between the nominal EA and

T requiring something verbal but between the two ϕ-feature bearing elements.

5 Concluding Remarks

Focusing on the operation Transfer, this paper addressed major problems with how it works, i.e. the idea that the operation literally removes from the derivational workspace a portion of structure within a phase. With the notion of DA that the interfaces are an active mechanism directly accessing the derivation for interpretation (Chomsky 2021, Forthcoming) and the introduction of the two conditions, the Accessibility Condition (AC) and the Condition on Full Interpretation (CFI), both of which naturally follow from the nature of interpretation, this paper presented a more principled account of why certain SOs must remain active in the derivation, while others become no longer accessible to (further) syntactic computations.

Many other related issues remain to be resolved, of course, but I would like to conclude this study by briefly discussing one of the issues that I think should be addressed in future research. Consider (19).

(19) a. what did you buy?
b. what ... you [$_{v\text{P}}$ <what> [$_{v\text{P}}$ <you> buy <what>]]

A'-movement has long been considered with little controversy to be successive cyclic as illustrated in (19b). Let's, then, examine the status of the three occurrences of *what* from our DA-based point of view. Once the *v*P phase is completed, CI will inspect all the SOs for interpretation. In so doing, it will notice that the two occurrences of *what* are not fully interpretable as each bears an uninterpretable feature.[16] Consequently, both will remain in the derivation. Derivation continues to construct the next CP phase where *what* in outer-Spec-*v*P moves to Spec-CP. Once this CP phase is completed, CI will once again inspect all the SOs. Note that of all the three occurrences of *what*, the one in its final

[16] Our analysis does not hinge on the specific nature of the uninterpretable feature(s) *what* bears (i.e. it could be a *wh*-feature or a Q-feature); what matters instead is that it does bear some uninterpretable feature that will eventually match with that of C.

landing site (i.e. in Spec-CP) can only be fully interpreted. This makes us wonder how the other two occurrences, i.e. *what* in outer-Spec-*v*P and *what* in-situ, can be fully interpreted or ‘transferred’ in the conventional terms.[17] I suspect that this issue may be resolved if we adopt the operation FORMCOPY (FC) proposed in Chomsky (2021, Forthcoming). Consider (20).

(20) a. John$_2$, ... [$_{vP}$ John$_1$ likes Mary]
b. FC → <John$_2$, John$_1$>

To distinguish between copies and repetitions,[18] Chomsky (2021, Forthcoming) proposes an operation called FORMCOPY which is hypothesized to assign a copy relation to identical SOs at the phase level, as illustrated in (20b). Once the relation is established by FC, the each occurrence of *John* is identified as a copy of the other.[19] In other words, the two *John*'s in (20b) refer to the same *John*. Let’s, then, return to (19b) to see how FC may resolve the issue we raised. ((19b) is simplified in (21) to focus on *what*).

(21) a. [$_{CP}$ what$_3$ ... [$_{vP}$ what$_2$ [$_{VP}$ ... what$_1$]]]
b. FC → <what$_3$, what$_2$>, <what$_2$, what$_1$>

In the course of derivation, the two pairs of copies in (21b) are formed by FC as shown in (21b). Note that the SOs in these pairs are, by definition, identical to each other (i.e. *what*$_3$ = *what*$_2$ = *what*$_1$). Therefore, if *what*$_3$ is fully interpretable, the other two, i.e. *what*$_2$ and *what*$_1$, can be or must be. I suspect this line of approach may provide a basis for a solution to the problem of (fully) interpreting intermediate copies generated in A’-movement.

[17] In fact, this problem of full interpretation of so-called ‘copies’ persists in A-movement as well.

[18] The term ‘copies’ refer to multiple occurrences of a single SO, while ‘repetitions’ to those of unrelated SOs.

[19] This is a bit of simplification of Chomsky’s (2021, Forthcoming) framework. To be ‘confirmed’ as legitimate copies, the copy pair must additionally satisfy one of what Chomsky calls the Language-Specific Conditions, namely, Theta Theory.

References

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

Chomsky, Noam. (2001). "Beyond Explanatory Adequacy." *MIT Occasional Papers in Linguistics 20*. Cambridge, MA.: MITWPL.

Chomsky, Noam. (2007). "Approaching UG from below." In Uli, S. and H-M. Gärtner (eds.), *Interfaces + Recursion = Language?* 1-29. Berlin: Mouton de Gruyter.

Chomsky, Noam. (2008). "On Phases." In R. Freidi, C. P. Otero and M. L. Zubizarreta (eds.), *Foundational Issues in Linguistic Theory: Essays in Honor of Jean-Roger Vergnaud*, 133-166. Cambridge, Mass.: MIT Press.

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

Chomsky, Noam. (2015). "Problems of Projection: Extensions." In Elisa Di Domenico, Comelia Hamann and Simona Matteini (eds.), *Structures, Strategies and Beyond - Studies in Hnor of Adriana Belletti*, 3-16. Amsterdam & Philadelphia: John Benjamins.

Chomsky, Noam. (2021). "Minimalism: Where are we now, and where can we hope to go." *Gengo Kenkyu* 160, 1-41.

Chomsky, Noam. Forthcoming. "Genuine Explanation." In Bocci, G., D. Botteri, C. Manetti, V. Moscati (eds.), *Issues in Comparative Morpho-syntax and Language Acquisition*.

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

Groat, Erich. (2015). "Total Transfer, Dynamic Labeling, and Transfer Remnants." In Grewendorf, G. (ed.), *Remnant Movement*, 257-320. Berlin/Boston: Walter de Gruyter.

Ke, Alan. (2017). "Full Phase Transfer." Ms. University of Michigan.

Mizuguchi, Manabu. (2017). "Learnability and Interpretability." *Studies in Generative Grammar* 27(2), 327-365.

Murphy, Elliot and Jae-Young Shim (2020). "Copy Invisibility and (Non-)Categorial Labeling. *Linguistic Research* 37(2), 187-215.

Obata, Miki. (2010). *Root, Successive-Cyclic and Feature-Splitting Internal Merge: Implications for Feature-Inheritance and Transfer*. Ph.D. dissertation. University of Michigan, Ann Arbor.

Shim, Jae-Young. (2018). <ф,ф>-less Labeling. *Linguistic Research* 54(1), 23-39.