Unifying Labeling under Minimal Search in "Single-" and "Multiple-Specifier" Configurations[*]

Samuel D. Epstein   Hisatsugu Kitahara   T. Daniel Seely

The University of Michigan   Keio University   Eastern Michigan University

1. Strong Minimalist Thesis

According to the strong minimalist thesis (SMT), language is a "perfect system," meeting the interface conditions in a way satisfying third factor principles (Chomsky 1995). Under SMT, the combinatorial operation of the generative procedure is a simple set formation device called Merge that takes α and β, and forms $\{\alpha, \beta\}$. The "perfect system" must have Merge, and ideally only Merge, and we expect this very simple, Merge-based system to interact with third factor principles such as Minimal Search (MS). That's the ideal picture. Of particular importance is the extent to which we get significant results through the interaction of Merge and MS.

With this in mind, let's examine the framework outlined by Chomsky (2013). First, the single, structure-building operation Merge puts two objects α and β into a relation, and the output of this operation is a two-membered set $\{\alpha, \beta\}$. There are two important points concerning the application of Merge: (i) Merge applies freely as long as it conforms to third factor principles, and (ii) Merge does not encode a label; there is no labeled categorial node above α and β, the categorial status of the set $\{\alpha, \beta\}$ is representationally unidentified.

For a syntactic object SO to be interpreted, however, it is necessary to know what kind of object it is (e.g., nominal, verbal, etc.). So, there must be some process of finding the relevant, category-identifying information of $\{\alpha, \beta\}$, generated by Merge. Chomsky (2013) takes what is referred to as 'labeling' to be the relevant process, and labeling is "just minimal search, presumably appropriating a third factor principle, as in Agree and other operations." Labeling is MS, i.e. MS finds the relevant head(s) allowing for categorial identification at the interface. Suppose SO = {H, XP}, H a head and XP not a head. Here,

MS finds H as the label of SO since H is the first element found with features relevant for categorial identification.[1] Suppose SO = {XP, YP}, neither a head. Here, MS is ambiguous, locating both the head X of XP and the head Y of YP. Left as is, labeling fails and Full Interpretation is violated at CI; the ambiguity is assumed to be intolerable. Chomsky (2013) argues that there are two ways to remedy this situation: (i) modify SO so that there is only one visible head, or (ii) X and Y are identical in a relevant respect, providing the label of the SO. Take a concrete case, $v*P$ with external argument NP (SO_1 = {{N, α}, {$v*$, β}}), where both N and $v*$ are located by MS. One way to label SO_1 is to raise {N, α} to a higher position. This movement leaves behind an invisible copy of {N, α}, whose invisible status follows if γ is taken to be in domain D iff every occurrence of γ is a term of D (Chomsky 2013).[2] Under this algorithm, in SO_1 = {{N, α}, {$v*$, β}}, MS finds the only "visible" head $v*$ as the label of SO_1 (given that the lower copy of {N, α} is invisible). As for SO_2 = {{N, α}, {T, SO_1}}, formed by the movement of {N, α}, Chomsky (2013) proposes that MS identifies the label for SO_2 with the prominent shared features of N and of T, namely the v(alued)Phi on N and u(nvalued)Phi on T, which participate in agreement. The idea is that MS finds the single feature Phi, and Phi will then count as the category identifier (i.e. the label) for SO_2 (see also Chomsky 2015).

2. "Multiple-Specifier" Constructions

We demonstrated above how the labeling theory works in "no specifier" and "single-specifier" configurations: {H, XP} and {XP, YP}. Let us now ask how labeling theory deals with "multiple-specifier" configurations: SO={XP, {YP, {ZP, WP}}}. Consider (1):

(1)   SO = {{X, α}, {{Y, β}, {{Z, γ}, {W, δ}}}}
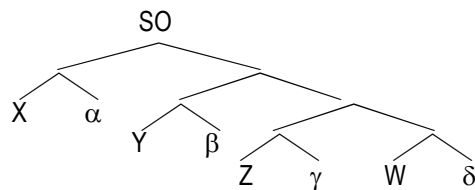
Unless stipulated, we expect the Merge-based system to generate the set-theoretic object (1) SO={XP, {YP, {ZP, WP}}} (with X, Y, Z, W heads and α, β, γ, δ not heads), and MS

---

[1] MS finds in {H, XP} both set members H and XP, but since XP, as a non-lexical item, is itself a set, it has no features that can serve as a category identifier.
[2] For earlier analyses of trace-invisibility, see also Chomsky 1995. We will see below that such invisibility follows naturally from MS.

to find the first head(s) for labeling. But what counts as the first head(s) in (1)? Should X be the first head?[3]

Take a concrete case. Japanese exhibits multiple nominative subjects (Kuno 1973), as in (2), schematically represented in (3) (for expository purposes, indices are assigned and the terminal elements are placed in the head-initial order). Each NP is assigned (or valued) nominative case (NOM) by finite T (Saito 2016):

(2) Bunmeikoku-ga   dansei-ga   heikin-zyumyoo-ga   mizikai
     civilized.country-NOM   male-NOM   verage-life.span-NOM   short-Pres.
     'It is in civilized countries that male's average life span is short.'

(3) SO = {{N_1, α}, {{N_2, β}, {{N_3, γ}, {T, δ}}}}
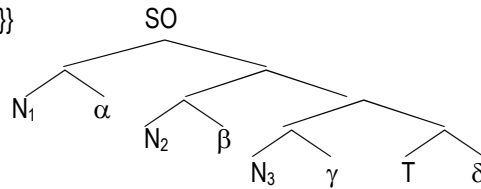
What counts as the first head(s) in (3)? Should N_1? That incorrectly labels (3) as nominal, and fails to capture the fact that NOM appears on all three NPs. Descriptive adequacy requires that, in (3), MS finds the three heads N_1, N_2, N_3, and the finite T (and only those four heads), so that the valuation of NOM by finite T on each NP will take place, and the uCase-vTense pair will count as the label of SO.[4]

The desired result is thus that MS finds all and only those four heads for computation, specifically, valuation and labeling, but at the same time, we have to explain why "multiple-specifier" configurations, such as Japanese multiple subjects (2), are not available for languages such as English. In effect, we are confronted with the problems of undergeneration and overgeneration, and in what follows, we demonstrate how these problems are overcome through the interaction of Merge and MS.

3. Minimal Search Defined and Unified

Let us ask again how MS finds its target(s) and makes it (them) accessible for

---

[3] Sorida (2014) originally asked the question of how labeling works with multiple-specifier configurations.

[4] We assume that uCase and vTense count as matching features; that is, at some abstract level uCase matches vTense for NOM and vTransitive for ACC; abstractly, vTense is NOM and vTransitive is ACC. See Pesetsky and Torrego 2004 for relevant discussion. See also Chomsky 1995 for earlier discussion.

computation (specifically, valuation and labeling). We would like to propose that MS finds a target in the optimal way via the shortest possible path. What counts as "shortest"? Take an H-XP configuration (with no "specifier"), where indices are assigned to each SO for expository purposes. Consider (4):
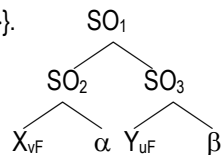
(4) $SO_1 = \{H, \{X, \alpha\}\}$.

$$
\begin{array}{c}
SO_1 \\
\diagup\,\diagdown \\
H \quad SO_2 \\
\diagup\,\diagdown \\
X \quad \alpha
\end{array}
$$
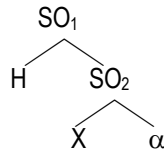
Here we adopt the idea of Chomsky (1995): a shorter path is selected over a longer one (see Ke 2019 for a detailed formal analysis of the nature of MS). As for what counts as "shorter," we propose that the path of α is the set of all SOs of which α is a term. Note that the non-reflexive definition of term is adopted here.[5] Then, the path of α is shorter than the path of β iff the path of α is a proper subset of that of β (cf. Pesetsky 1982, May 1985). Given this shortest set-theoretic search, in (4), the path of H is $\{SO_1\}$, and the path of X is $\{SO_1, SO_2\}$. Since the former is a proper subset of the latter, the former is shorter, and consequently H is selected over X. That is, assuming "shorter" means "properly contained in," it follows that MS selects H over X because the path of H $(=\{SO_1\})$ is a proper subset of the path of X $(=\{SO_1, SO_2\})$; hence, only H counts as an accessible head for labeling.[6]

How about an XP-YP configuration (with a "single-specifier"). Consider (5):

(5) $SO_1 = \{\{X, \alpha\}, \{Y, \beta\}\}$.

$$
\begin{array}{c}
SO_1 \\
\diagup\,\diagdown \\
SO_2 \quad SO_3 \\
\diagup\,\diagdown \quad \diagup\,\diagdown \\
X_{vF} \quad \alpha \; Y_{uF} \quad \beta
\end{array}
$$

It follows from the path-theoretic account that MS selects both X and Y because neither the path of X $(= \{SO_1, SO_2\})$ nor the path of Y $(= \{SO_1, SO_3\})$ is a proper subset of the other. So, both X and Y are located by MS, and they count as accessible heads for labeling and valuation.

Interestingly, MS, defined as the shortest set-theoretic search, yields the right

---

[5] We assume: X is a term of Y iff $X \in Y$ or $X \in Z$, Z a term of Y. See Chomsky 2019 UCLA lectures for further comment.
[6] We assume that every term in the workspace is visible; hence, Search sees them, but among them, MS determines which terms count as accessible ones.

empirical results also for "multiple-specifier" configurations (SO={XP, {YP, {ZP, WP}}}).
Consider (6):

(6) $SO_1 = \{\{N_1, \alpha\}, \{\{N_2, \beta\}, \{\{N_3, \gamma\}, \{T, \delta\}\}\}\}$
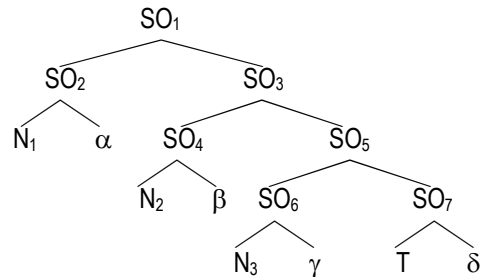
Recall that descriptive adequacy requires that MS finds the three nominal heads N_1, N_2, N_3, and the single finite T (and only these four heads), so that the valuation of NOM by finite T on each NP will take place (Saito 2016). Now notice, under this system, MS finds N_1, N_2, N_3, and finite T (and only them) because there is no proper subset relation among the following four relevant sets: (i) the path of N_1 (= $\{SO_1, SO_2\}$), (ii) the path of N_2 (= $\{SO_1,$ $SO_3, SO_4\}$), (iii) the path of N_3 (= $\{SO_1, SO_3, SO_5, SO_6\}$), and (iv) the path of finite T (= $\{SO_1, SO_3, SO_5, SO_7\}$): Thus, in (6), N_1, N_2, N_3, and finite T (and only these four heads) are located by MS, and Case-valuation takes place among them, and the uCase-vTense pair counts as the label of SO_1.

The proposed analysis also explains why lower copies, left by movement, are not available for computation. Recall the "subject-raising" case, where the movement of {N, α} from Spec-v* to Spec-T renders the lower copy of {N, α} invisible; hence, v* is unambiguously identifiable, repeated in (7):

(7) $SO_1 = \{\{N, \alpha\}, \{T, \beta\}\}$.
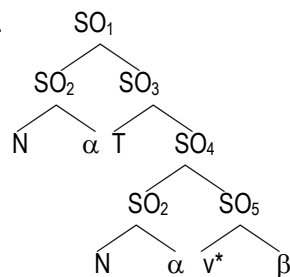
In (7), the higher copy of SO_2 (= {N, α}) is available for computation, but the lower copy of SO_2 (= {N, α}) is not, because the path of the higher copy of SO_2 (= $\{SO_1\}$) is a proper subset of the path of the lower copy of SO_2 (= $\{SO_1, SO_3, SO_4\}$). Thus, the invisible status

of the lower copy naturally follows.[7]

As demonstrated above, MS gives unified and predictively correct results. That is, anything that is found in the shortest possible way counts as an accessible term for computation, where the notion "shorter" is understood in terms of "proper subset" (not a uniquely linguistic concept, hence arguably third factor). There is no need to assume any special device (e.g. Chomsky's (2013) algorithm concerning (in)visibility) to get these results: what counts as an accessible term naturally follows from MS defined as the shortest set-theoretic search.

4. Revisiting "Single-" vs. "Multiple-Specifier" Configurations

The remaining task is to explain why "multiple-specifier" configurations exist in some languages, but not in others; for example, multiple subjects appear in Japanese, but not in English. What determines the presence (or absence) of "multiple-specifier" configurations? What separates Japanese from English? We would like to propose that "multiple-specifier" configurations appear iff MS finds one and only one valuing head per agreement-relation; that is, for each unvalued feature *uF-valuee,* there is one and only one valued feature *vF-valuer.* Why? Because every SO, including "multiple-specifier" configurations, must yield a unique identification for labeling, multiple valuers would violate this uniqueness principle. With this requirement, let us examine "single-" and "multiple-specifier" configurations again. First consider the "single-specifier" configuration (8), where N bears vPhi and uCase, while T bears uPhi and vTense:

(8) $SO_1 = \{\{N, \alpha\}, \{T, \beta\}\}.$

---
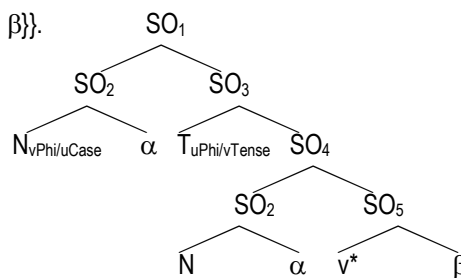
[7] Note further that in (7), elements contained within the lower copy of SO_2 (i.e. N and α) are also invisible. The path of N or α within the higher copy of SO_2 is $\{SO_2, SO_1)$ while the path of N or α in the lower copy is $\{SO_2, SO_4, SO_3, SO_1\}$ where the former path is included in the latter and hence is shorter. In contrast, note that any analysis relying on c-command will not yield this result; the entire higher copy of object SO_2 does c-command the lower copy, but elements *within* this higher copy do not. If a lower copy is invisible only if it is c-commanded by a higher copy, then the lower copies of N and α would be visible.

In (8), MS finds both N and T, and vPhi on N values uPhi on T, and as a reflex, vTense on T values uCase on N (Chomsky 2013, 2015).[8] Notice, for each uF-valuee (uPhi, uCase), there is one and only one vF-valuer (vPhi, vTense); hence, the two pairs, uPhi-vPhi and uCase-vTense, constitute a unique label.

Now turn to the "multiple-specifier" configuration (9), where each N bears vPhi and uCase, while T bears vTense but not uPhi:

(9) $SO_1 = \{\{N_1, \alpha\}, \{\{N_2, \beta\}, \{\{N_3, \gamma\}, \{T, \delta\}\}\}\}$
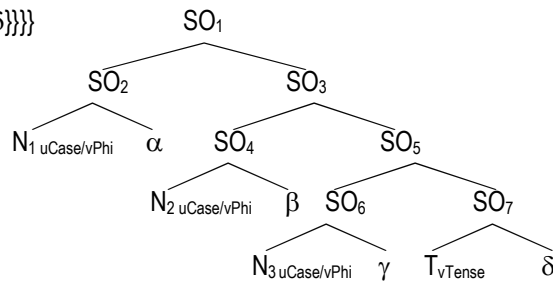
MS finds the three nominal heads N_1, N_2, N_3, each bearing uCase, and the finite T bearing vTense. Saito (2016) argues that vTense values uCases of N_1 N_2, and N_3. Notice, for each uF-valuee (uCase), there is one and only one vF-valuer (vTense); hence, the uCase-vTense pair constitutes a unique label. But suppose finite T had uPhi in the "multiple-specifier" configuration. What would happen? Consider (10):

(10) $SO_1 = \{\{N_1, \alpha\}, \{\{N_2, \beta\}, \{\{N_3, \gamma\}, \{T, \delta\}\}\}\}$
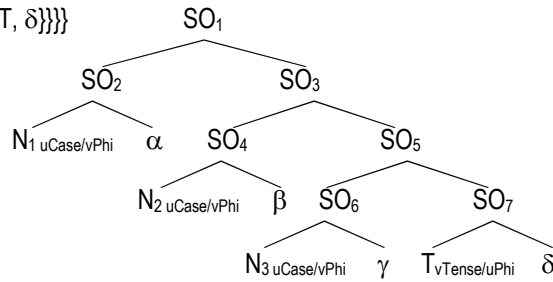
In (10), the three nominal heads N_1, N_2, N_3, bearing vPhi, each participate in valuing uPhi on finite T; hence, there would be no single phi-valuer because the three nominal heads N_1, N_2, N_3 bear distinct phi-sets (even if they accidentally bear same values), and each participate in phi-valuation, thereby failing to yield a *unique* phi-label for SO_1.

---

[8] Under Chomsky's (2013, 2015) system, Case valuation is a reflex, i.e. an automatic consequence, of Phi valuation. We assume that T bears vTense, and in (8) there are, in effect, two instances of valuation: uPhi valued by vPhi, and uCase valued by vTense (see footnote 5 above).

Discussion

So, what separates Japanese from English with respect to the licensing of "multiple-specifier" configurations? We would like to end this paper with the following three possibilities: The first possibility is that unlike English, Japanese has no uPhi (see (9) and (10)). That is, the presence of uPhi on English T blocks "multiple-specifier" configurations. If there were two or more distinct vF-valuers for one uF-valuee, then a labeling failure would result (for earlier proposals, see Fukui 1986, Kuroda 1988; for recent discussion, see Saito 2016, Sorida 2014). The second possibility is that Japanese bears uPhi (as in (10)), but unlike English uPhi, Japanese uPhi has no morpho-phonological realization. That is, if there were two or more distinct vF-valuers for one uF-valuee, then there would be no way to realize such multiple phi-sets on the single head. But unlike English, Japanese has no such mopho-phonological realization of uPhi; hence this externalization problem can be circumvented (Kitada personal communication).[9] And the third possibility is that UG has uF, and in English, uF is realized as uCase on N, and as uPhi on T, whereas in Japanese, uF is realized as uCase on N but remains as uF on T.[10] Consider (11):

(11) $SO_1 = \{\{N_1, \alpha\}, \{\{N_2, \beta\}, \{\{N_3, \gamma\}, \{T, \delta\}\}\}\}$
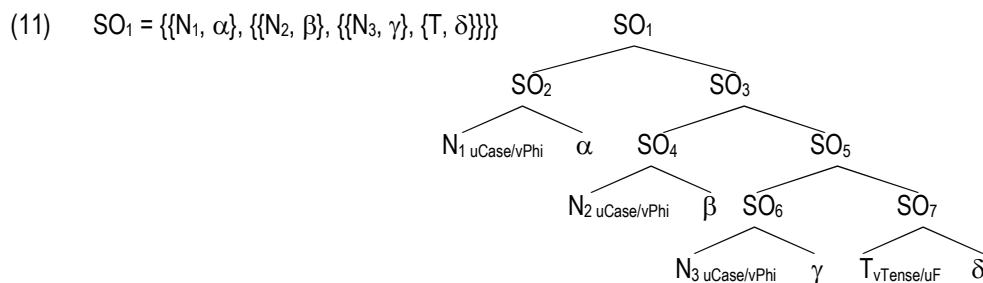
Suppose UG has unvalued feature uF. In English, uF is realized as uPhi that matches Phi and gets Phi-valued. By contrast, in Japanese, uF is realised as uF with no property that matches any F but gets no value. Suppose valuation takes place only if a valuer has some unvalued feature (i.e. the activity condition, Chomsky 2000). Then, English finite T values uCase on N as long as uPhi on T remains unvalued, meaning: only once, whereas Japanese finite T values uCase on N as long as uF on T remains unvalued, meaning: continuously.

---

[9] Though interesting, it is unclear on this approach how a unique label is determined.

[10] The notion of uF is somewhat similar to that of "edge feature" (Chomsky 2000, 2001) in that in Japanese uF on T matches with any feature, but it has no specific properties, so it cannot be valued; hence it remains unvalued throughout a derivation. In short, uF alone requires just matching, while uF plus a property (e.g. uCase or uPhi) requires matching and valuation for convergence.

_____

That is, English uPhi realizes morpho-phonologically (sometimes vacuously), while Japanese uF never realizes morpho-phonologically (because there is no value).

Also notice, the third possibility is arguably consistent with the uniformity hypothesis: "In the absence of compelling evidence to the contrary, assume languages to be uniform, with variety restricted to easily detectable properties of utterances" (Chomsky 2001). Under this hypothesis, UG expects that finite T uniformly values Case iff it bears unvalued features, while linguistic variation comes down to the problem of externalization. So, in English, uF is realized as uPhi, and uPhi gets Phi-valued, so Case-valuation happens only once. In Japanese, however, uF matches but remains unvalued, so Case-valuation may happen continuously, thereby allowing "single-" as well as "multiple-specifier" configurations to appear. Now, if English changes from uPhi (back) to uF, then it becomes like Japanese; if Japanese changes from uF (on)to uPhi, then it becomes like English. One might argue that such language change is hard to imagine under the assumption that English has uPhi (or uF) but Japanese has no counterpart of it.

Finally, let's ask what would happen if there were no uF. Chomsky (2013, 2015) suggests that uF marks phases, so without uF, there would be no way to mark them. In addition, we would like to suggest that, without uF, there would be neither "single-" nor "multiple-specifier" configurations, and there would be no phrasal movement. It is uF that makes available exocentric structures or XP-YP configurations including "multiple-specifier" configurations.[11]

References

Chomsky, N. 1995. The minimalist program. Cambridge, Mass.: MIT press.

Chomsky, N. 2000. Minimalist inquiries: the framework. In R. Martin, D. Michaels & J. Uriagereka (Eds.), Step by step: essays on minimalist syntax in honor of Howard Lasnik (pp.89-155). Cambridge, MA: MIT Press.

Chomsky, N. 2001. Derivation by phase. In M. Kenstowicz (Ed.), Ken Hale: A life in language (pp.1-52). Cambridge, MA: MIT Press.

Chomsky, N. 2013. Problems of projection, Lingua, 130, 33-49.

Chomsky, N. 2015. Problems of projection: extensions. In E. D. Domenico, C. Hamann & S, Matteini (eds.), Structures, strategies and beyond: studies in honor of Adriana Belletti (pp.3-16). Amsterdam: John Benjamins.

_____

[11] See also Fujita, Uchibori, and Kitahara 2017 for further discussion on the role of uF.

_____

Chomsky, N. 2019. UCLA Lectures. [https://linguistics.ucla.edu/noam-chomsky/]

Fujita, G., A. Uchibori, and H. Kitahara 2017. Why do unvalued features exist in languages? Unpublished manuscript.

Fukui, N. 1986. A theory of category projection and its applications. Doctoral dissertation, MIT, Cambridge, MA.

Ke, A. 2019. The syntax, semantics and processing of agreement and binding grammatical illusions. Doctoral dissertation, The University of Michigan, Ann Arbor, MI.

Kuno, S. 1973. The structure of the Japanese language. Cambridge, Mass.: MIT Press.

Kuroda, S.-Y. 1988. Whether we agree or not: A comparative syntax of English and Japanese. Linguisticae Investigationes 12. 1-47

May, R. 1985. Logical form: its structure and derivation. Cambridge, Mass.: MIT Press.

Pesetsky, D. 1982. Paths and categories. Doctoral dissertation, MIT, Cambridge, Mass.

Pesetsky, David and Esther Torrego. 2004. Tense, case, and the nature of syntactic categories. In Jacqueline Gueron and Jacqueline Lecarme. eds. *The syntax of time*. pp. Cambridge, Mass.: MIT Press.

Saito, M. 2016. (A) Case for Labeling: Labeling in languages without phi-feature agreement. The Linguistic Review 33 (Special Issue on Labeling), 129-175.

Sorida, M. 2014. Multiple-branching structures in syntax. Proceedings of the 16th Seoul International Conference on Generative Grammar (SICOGG 16), 411-419.

©Samuel D. Epstein
The University of Michigan
sepstein@umich.edu

©Hisatsugu Kitahara
Keio University
kitahara@icl.keio.ac.jp

©T. Daniel Seely
Eastern Michigan University
 tseely@emichi.edu