Labeling without Labels

Chris Collins, New York University and
T. Daniel Seely, Eastern Michigan University
Oct 8, 2020

Abstract: Despite the use of the expressions 'labels, 'labeling algorithm' and 'projection' (of labels), it is argued that the labeling algorithm in Chomsky 2013 in fact assumes and is constructed within a label-free syntax of just the sort developed by Collins 2002 and Seely 2006.

Keywords: labels, labeling algorithm, label-free syntax, minimal search, simplest Merge, 3rd factor, interfaces, Strong Minimalist Thesis

1. Introduction

Despite reference to *labels* and the development of an explicit *labeling algorithm*, Chomsky's 2013 'Problems of Projection' [Henceforth, PoP] does not appeal to labels at all. Rather, PoP's analysis is entirely consistent with the label-free syntax proposed by Collins 2002 and further developed and defended in Seely 2006. 'Labeling' in PoP is something of a misnomer, for just as in Collins and Seely [henceforth C-S], there are no labels, nor label projection in PoP. Rather, PoP offers a new, and particularly insightful answer to the fundamental question: If there are no labels and no projection, how is the information provided by labels to be derived? PoP's answer is, on the conceptual level, beautifully simple: with respect to categorial identification of a syntactic object at the interfaces, label information is provided by (i) independently available lexical features, and (ii) 3d factor, hence freely available, Minimal Search.

2. Eliminating Labels

In this section we give a brief overview of Collins 2002 and Seely 2006 in order to set the stage for our discussion of PoP.

2.1. Bare Phrase Structure

In the discussion of the definition of Merge, Chomsky 1995 noted that the simplest possible formulation would be Merge(A,B) = {A,B}:

(1) "The simplest object constructed from α and β is the set {α, β}, so we take K [C.C and T.D.S -- the object constructed by Merge] to involve at least this set… Does that suffice? Output conditions dictate otherwise; thus, verbal and nominal elements are interpreted differently at LF and behave differently in the phonological component. K [C.C and T.D.S -- i.e. the object constructed by Merge] must therefore at least (and we assume at most) be of the form {γ {α, β}}, where γ identifies the type to which K belongs, indicating its relevant properties. Call γ the *label* of K." (Chomsky 1995a: 243, see also Chomsky 1995b: 397)

As seen in the quote, Chomsky proposed but immediately rejected the simplest formulation of Merge, arguing instead for the necessity of syntactic category labels as part of the syntactic object formed by Merge.

Collins 2002 and Seely 2006, on the other hand, argued that labels and projections could be completely eliminated from syntactic representations and that, as a result, Merge could be reduced to its simplest form: Merge(X, Y) = {X, Y}.

2.2. Collins 2002

Collins 2002 begins with the account of phrase structure in Chomsky 1995 "Bare Phrase Structure" where Merge(X,Y) combines X and Y, and creates a syntactic object Z with a dedicated, syntactically represented label. For example, if X and Y are lexical items, and X is the label of Z, then Z = {X, {X,Y}}.

Collins notes that Merge is doing two very different things here, (i) combining X and Y into a set {X, Y}, and (ii) giving the combination of {X, Y} a label. Furthermore, to account for how I-languages make "infinite use of finite means", an operation combining X and Y is essential, but giving the created object a label is not:

(2) "In any theory of grammar, there will be a lexicon, a PF component and an LF component. In addition, there will be some operation (called Merge) that combines phrases and lexical items into larger phrases. This operation is a necessary part of any theory of human grammar. It allows us to explain how grammar makes 'infinite use of finite means'. In other words, given two constituents A and B, there must be some way to combine these into a larger constituent {A, B}. What is important to realize is that the notion of a label for {A,B} goes way beyond these elementary considerations, since in addition to the combination of A and B, either A or B has to be designated as the label." (Collins 2002: 43).

For these reasons, Collins proposed that Merge(A,B) forms a label-free syntactic object {A,B}. Collins then tried to show how areas of syntactic theory where labels are traditionally used could be handled without labels; i.e. that the information provided by and the effects of labels could be derived from independently available principles of the grammar. Collins focused on the following areas with respect to which labels traditionally played a key role:

(3) a. X'-Theory
b. Selection
c. Minimimal Link Condition
d. PF Interface

The general strategy was the following:

(4) "In a highly derivational theory, all syntactic generalizations must be derived from the interaction of economy conditions, the properties of individual lexical items (e.g., X has an uninterpretable feature F, X needs two arguments) and interface conditions (bare output conditions, or legibility conditions)." (pg. 44)

To give an idea of how this strategy worked in Collins 2002, consider his account of c-selection (see Seely 2006 for further exploration of Collins 2002 proposals about c-selection). For concreteness, consider the case of a determiner c-selecting an NP complement (see Collins 2002 for other cases). Using simplest Merge, without labels or projections, we have the following representation:

(5) {the, {picture, {of, Bill}}}

How can the c-selectional feature of *the* be satisfied in this structure? Collins assumes that satisfaction of c-selectional features obeys minimality. Basically, in (5) the determiner *the* probes its sister {picture, {of, Bill}}, and finds the noun *picture*. The noun *Bill* could not have satisfied the c-selection feature of *the*, because *picture* is closer.

In effect, Collins 2002's account of c-selection was a labeling algorithm in the sense of PoP. Information traditionally provided by a syntactic category label is obtained under minimal search from a syntactic object without any syntactic category labels or projections.

2.3. Seely 2006

Seely 2006 adopts and defends key proposals of Collins' 2002 label-free syntax, stressing that the elimination of labels allows for the simplification of the operation of Merge itself, reducing it to the conceptually most desirable form, namely Merge(X, Y)={X, Y}. In short, if there are no labels in the output representation, then there need not be any 'projection phase' of the Merge operation.

As outlined above, Collins considers a number of principles and components of the grammar with respect to which labels have traditionally played a central role, and Collins argues that the effects of labels can be derived from other, independently motivated, principles of the grammar. Seely 2006 takes up another domain where labels figure prominently, namely, in the definition of c-command, arguing that it too can be formulated without appeal to labels.

Seely 2006 approaches the issue of eliminating labels from the point of view of Epstein's (1999) derivational definition of c-command:

(6) Derivational Definition of C-Command (from Epstein 199:329)
X c-commands all and only the terms of the category Y with which X was paired (by Merge or by Move) in the course of the derivation.

Adopting the notion 'label' as characterized in Bare Phase Structure, Seely carefully works through relevant definitions and shows the following:

(7) i. The label of a category is never derivationally c-commanded by any category.
ii. The label of a category never derivationally c-commands any category.

But if derivational c-command is a prerequisite to entering into syntactic relations, it follows that:

(8) “Labels cannot be involved in syntactic operations; labels are syntactically inert, and are, in effect, “already” eliminated (i.e., for all intents and purposes, absent) from the syntax.” (Seely 2006: 184, 189).

In short, once we adopt derivational c-command, labels are, in effect, unnecessary (for the characterization of c-command), taking yet another step toward the elimination of labels.

2.4. Steps towards PoP

This short exposition does not do justice to either Collins 2002 or Seely 2006. The central point is that unlike Chomsky 1995 Bare Phrase Structure, which adopts dedicated, syntactically represented labels and (hence) the complex Merge operation that produces them, C-S argue (with very different arguments from a variety of different angles) for the elimination of syntactically represented labels entirely, clearing the way for *simplest* Merge, defined as follows (for a formalization see Collins and Stabler 2016):

(9) For all syntactic objects X, Y, Merge(X,Y) = {X,Y}

In various works subsequent to Bare Phrase Structure and preceding PoP, Chomsky takes steps toward simplest Merge, as originally formulated (and immediately rejected) in BPS (see (1) above). One obstacle was the need for a syntactic object to be identified at the interfaces. The assumption is that a syntactic object must be categorially identified for interface interpretation. Below, we will refer to this as ‘object identification’ for brevity’s sake. In Chomsky’s earlier work, labels were the means by which such object-identification information was derived. Thus, in Chomsky 2000 *Minimalist Inquires* (MI):

(10) “The operation Merge forms K from A, B. Minimally, K should consist only of A and B, so K = {A, B}. More information is needed about K however: its category (its *label)* … needed for the phonological component and LF interpretation. To ensure that every category has a label, let us say that label(A) = A*,* for A a LI.” (pg. 133)

In the MI approach, the label is not part of the syntactic object, but it is derived by a function *label*.

And in Chomsky 2005, we find:

(11) a. “Any such system is based on a primitive operation that takes *n* objects already constructed, and constructs from them a new object: in the simplest case, the set of these *n* objects. Call that operation Merge.” (pg. 11)

b. “We would hope that labels are determined by a natural algorithm, perhaps even by methods as far-reaching as those that Chris Collins (2002) has proposed in his label-free analysis.” (pg. 14)

What we’d like to argue in the next section is that Chomsky’s Problem of Projection 2013 adopts the label-free approach, deriving the ‘object identification’ function of labels for the interfaces from a hypothesized 3rd factor principle, that of Minimal Search, which, in turn allows for the elimination of the EPP and ECP. The approach taken in PoP presupposes the adoption of simplest Merge, namely Merge(X,Y) = {X, Y}, with no dedicated label nor a 'projection phase' built into Merge.

3. The labeling algorithm of PoP is consistent with the label-free syntax of C-S.

PoP is a particularly insightful example of what is sometimes referred to as 'eliminative minimalism' (on which see, among others, Epstein, Kitahara, Seely 2015). A central goal of minimalist research has been to reduce the inventory of theoretical postulates, reducing them to what is conceptually necessary and to 3rd factor, i.e. general laws of nature. A structure building device, i.e. Merge, is necessary if there is to be syntax at all. Merge interacts with the interfaces (Conceptual Intentional CI and Sensorimotor SM) in that the objects generated by Merge must be interpretable by the interfaces. Merge is reduced to its simplest form (Merge(X, Y)={X, Y}), yielding an ideal equation of *Merge + 3rd Factor + Interfaces = Language*. As Chomsky (2000) notes this "highly characteristic effort to minimize" "... "provides no machinery beyond what is needed to satisfy minimal requirements of legibility and that it functions in as simple a way as possible" (Chomsky 2000, 112-113). The quest for simplicity is, on the one hand, characteristic of science: "*It has long been clear that the quest for simplicity is closely related to the quest for explanation…*" (Chomsky, *The Minimalist Program,* new preface (2015b, vii)). It is also consistent with what is known about the very recent emergence of language in the species (see Berwick and Chomsky (2016)).

PoP, combined with key additions of Chomsky 2015 *Problems of Projection: extensions* [henceforth PoP+], advances the eliminative research program traced above by reducing two hallmark principles of Government and Binding Theory (see Chomsky 1981), namely the Extended Projection Principle (EPP) and the Empty Category Principle (ECP), to what is referred to as *labeling*. Putting aside the details, the EPP and ECP effectively determine when some element must exit a certain position (for instance an intermediate position with *wh* movement) and when an element must stay in some position (e.g., that the Spec of a tensed TP must be occupied). PoP/PoP+ argues that these effects of the EPP and ECP in fact follow from the way the label of some object is (or must be) determined: Movement out of a position is sometimes forced in that if the element did not move, labeling failure would result; likewise, in certain cases, if the element *did* move, labeling failure would result. Movement (i.e. Internal Merge) applies freely, but when it 'must' or 'cannot' apply is determined, in large part, by labeling.

It is not our purpose to review the technical details of PoP's insightful deduction of EPP/ECP effects (see amongst others Epstein, Kitahara, and Seely 2015). Rather, we address here what we take to be something of a misnomer in PoP, that of *labeling* itself. In PoP, syntactic objects are said to require a label at the interfaces, the system has a detailed 'labeling algorithm,' and, as noted above, the means by which labels are determined allows for the elimination of the EPP/ECP.

On the surface, then, PoP, with its apparent appeal to labels, would seem at first glance to be entirely incompatible with the elimination of labels and label projection as advocated in Collins and Seely (C-S): PoP *has* labels, C-S do not. In fact, some authors seem to interpret PoP as involving syntactically represented labels. Thus, Rizzi 2015's insightful analysis of Criterial Freezing with *wh*-movement, based on PoP's labeling algorithm, seems to involve, crucially so, the presence of dedicated labels (for detailed discussion see Epstein, Kitahara, and Seely 2015, chp 11). We suggest, however, that a deeper look leads to exactly the opposite conclusion: PoP in fact advances the central goals of a label- and projection-free syntax. In this section, we present the case for such a conclusion.

3.1 PoP adopts simplest Merge

PoP adopts the conclusions of Collins 2002 and Seely 2006 regarding the definition of simplest Merge (rejecting the earlier conclusions of BPS):

(12) Merge (X, Y) = {X, Y}

There is no 'projection phase' of Merge and the syntactic object that results from the application of Merge does not contain a dedicated label. As noted in section 2 above, Merge in BPS *has* a 'projection phase;' i.e. BPS Merge takes syntactic objects X, Y and puts them into the set {X, Y}; but BPS Merge *then* puts the set {X, Y} into an additional set containing {X, Y} and what is referred to as the label of the final derived set; thus PBS produces {L, {X, Y}}, where L is a dedicated (i.e. syntactically represented) label. Clearly, PoP, by adopting (1), i.e. simplest Merge, eliminates dedicated syntactic labels.

3.2 The 'labels' of PoP are the result of 3rd factor Minimal Search

So, what is meant in PoP by 'the label of a syntactic object'? If there are no dedicated syntactic labels, then how can PoP refer to 'a label'? As noted in section 2 above, C-S, in eliminating labels, implicitly ask the question:

(13) If there are no labels, how is the information encoded by labels derived?

Traditionally, labels play an important role in the syntax; they provide information for, among others, c-selection, X'-theory, the PF interface, *etc*. The information provided by labels must be encoded somehow, and both Collins and Seely provide means for doing just this; specifically, by arguing that the information provided by labels can be obtained from *independently motivated principles of the gramma*r. For C-S, labels are eliminated, their role taken over by other principles, a research program in line with Chomsky's Strong Minimalist Theory (see Chomsky 2000). It is in the spirit of C-S, that we can ask question (13) with respect to the analysis of PoP. In adopting simplest Merge, syntactic objects in the PoP analysis, do not have labels (clearly not in the sense of BPS). So, how is the information encoded by labels derived? The answer in PoP is particularly insightful:

(14) The information is derived via 3rd factor minimal search.

What information does PoP focus on? PoP assumes that syntactic objects must be identified at the interfaces (at both CI and SM). Thus, PoP states:

(15) "For a syntactic object SO to be interpreted, some information is necessary about it: what kind of object is it?" (PoP, p. 43)

An unidentified SO is not interpretable at the interfaces. Presumably, then, an unidentified SO will induce interface crashing, or be gibberish (see Collins 2017 for a relevant discussion of the kinds of information needed at the interfaces).

Historically, syntactically dedicated labels encoded this object-identification information, from standard PS rules, through X-bar theory, and up to and including BPS. Thus, in standard PS-rules like:

```
(16)    VP → V NP   =         VP
                              / \
                             V  NP
```

the ‘V’ of ‘VP’ identified the syntactic object as ‘verbal’, distinguished from NP, AdjP, *etc.* Similarly, in X-bar theoretic representations. With the transition from graph-theoretic to set-theoretic representations of phrases (the transition taking place with BPS), the object identifying label was encoded, as pointed out above, in the label; i.e., the label L of {L, {X, Y}} serves as the identifier of the entire object; hence {V, {V, NP}} is identified as ‘verbal’.

For PoP, object-identifying information cannot be provided by syntactically-represented labels for the simple reason that there *are no such labels* (i.e. there is no ‘VP’ above V+NP). The identification information of the label-less syntactic object {X, Y} must be provided only by what is already present in {X, Y}. That is, it must be provided by X and/or Y, since that’s all there is. And this is precisely what PoP does. Consider a simple verb phrase. As noted above, with a classic graph-theoretic representation as in (16), the label VP is providing the information that the object, namely V+NP, is ‘verbal’. Deconstructing the label, we see that it has two ‘parts’; the ‘V’ and the ‘P’. V provides the information ‘verbal’, but note that the V of ‘VP’ is just a copy of what’s already part of the syntactic object, namely the verb V itself. The ‘P’ provides the information that it’s a phrase (and not a bare verb); hence VP ≠ V.

Consider now the simplest-Merge representation adopted by PoP for the VP of (16), namely {V, NP}. The information that it’s a phrase is already (and inherently) encoded by the set brackets {… }. It’s a ‘phrase’ because it’s set; hence the information ‘phase’ follows automatically and need not be represented by a label (thus we eliminate the ‘P’ of ‘VP’). What about the information that the set (i.e. the phrase) is ‘verbal’? Somehow, we need to retrieve the relevant features (verbal vs nominal, *etc*) that are inherently borne by individual lexical items. The object-identification information of phrases does arise out of the blue; in fact, it’s provided by lexical material. In (16) the ‘verbal’ of VP is clearly derived from the fact that its head is a verb; it’s the lexical features of the verb that ultimately serve as the identifier of the larger object. With PoP’s representation {V, NP}, the identifying features are located in V via the independently available, 3rd factor, principle of Minimal Search:

(17) “For a syntactic object SO to be interpreted some information is necessary about it: what kind of object is it? Labeling is the process of providing that information. … We assume, then, that there is a fixed labeling algorithm LA that licenses SOs so that they can be interpreted at the interfaces, operating at the phase level along with other operations. *The simplest assumption is that LA is just minimal search, presumably appropriating a third factor principle, as in Agree and other operations.”* [our emphasis, C&S], (PoP, p. 43).

What is called 'labeling' is nothing other than minimal search "appropriating a third factor principle." With respect to the SO {V, NP}, at the phase level, Minimal Search MS 'looks into' the set and finds its two members: V and NP. NP is itself a set and *qua set*, it has no object-identifying features, which is to say that a set has no lexical features, in fact it has no linguistic features at all (a set is not a lexical item). The lexical item V, on the other hand, bears relevant lexical features, in this case the features 'verbal'. This featural information is automatically provided by 3rd factor Minimal Search, and the information is used by the interfaces to identify the object; i.e. the information 'verbal' is appealed to for object-identification. MS blindly provides the results of search; in the case of {V, NP}, MS basically says: I found a set (=NP) and a verbal thing V; i.e. I found the two members of the set I'm searching. The interfaces in fact can use the information 'verbal' and do so, interpreting the object as such; the interfaces avail themselves of information that is automatically given for free by MS.

At bottom, that's really all there is to it—beauty in simplicity. Again, the expressions 'label', 'labeling algorithm' and 'projection' of labels in PoP are something of a misnomer; they are used rather informally to mean 'object identification information' and 'minimal search.' 'Labels', 'labeling algorithm' and 'projection' in PoP are not real (i.e., part of the syntactic representation) in the sense of the traditional (i.e., Principles and Parameters and Bare Phrase Structure) uses of these terms. In PoP they have no theoretical status. *We could just as well say 'lexical features' (for 'label') and minimal search (for labeling algorithm) with no loss of actual content.*

3.3 Putting it all together

PoP argues further that:

(18) "In the best case, the relevant [C.C. and T.D.S -- object-identifying] information about SO will be provided by a single designated element within it: a computational atom, to first approximation a lexical item LI, a head." (PoP, p. 43)

This is entirely natural in that *more* than a single designated element would provide conflicting information to the interfaces, identifying the object simultaneously as an X and a Y. The system developed in PoP appeals to this natural uniqueness principle to derive ECP and EPP effects, as noted above. Again, further review of the technical details of the PoP system goes beyond our present concerns. Our central point is that all the technical apparatus in PoP reduces to PoP's new, and insight answer to the question of (2)

(19) If there are no labels, how is the information encoded by labels derived?

For PoP, it reduces to (i) the axiom that an object must be identified to be interpreted at the interfaces, (ii) lexical features, and (ii) 3rd factor minimal search. Just as in C-S, there are no labels nor label projection in PoP. PoP is squarely within the label-free syntax approached developed, and pursued, in C-S.

To repeat the program for the study of labeling phenomena of Collins 2002:

(4) "In a highly derivational theory, all syntactic generalizations must be derived from the interaction of economy conditions, the properties of individual lexical items (e.g., X has an

> uninterpretable feature F, X needs two arguments) and interface conditions (bare output conditions, or legibility conditions).” (pg. 44)

In the case of PoP, the “economy conditions” are minimize search, needed to find the label. The “properties of individual lexical items” are the lexical items themselves or formal features of the lexical items (e.g., +Q or phi-features). The interface condition is the axiom that all syntactic objects need to be identified (by the labeling algorithm).

We can highlight this point (the parallel programs of C-S and PoP), but comparing the similarities between two labeling algorithms (the one in PoP and the one proposed for c-selection in Collins 2002 and discussed in Seely 2006, see section 2 above).

(20) Reason for search:
a. Collins 2002: c-selection feature of head needs to be satisfied
b. Chomsky 2013: object identification at the interfaces

(21) Target of the Search:
a. Collins 2002: formal features or lexical items
b. Chomsky 2013: formal features or lexical items

(22) Search domain:
a. Collins 2002: sister of head with c-selection requirement
b. Chomsky 2013: syntactic object being transferred

(23) Starting point of search:
a. Collins 2002: lexical item bearing a c-selectional feature
b. Chomsky 2013: root of syntactic object being transferred

(24) Search strategy:
a. Collins 2002: minimal search (Minimality)
b. Chomsky 2013: minimal search

(25) Point in the syntactic derivation when search is done:
a. Collins 2002: at Merge
b. Chomsky 2013: at Transfer

4. Conclusion

Despite terminology, PoP/PoP+ is in fact a completely label-free syntax, entirely consistent with the framework established by Collins 2002 and Seely 2006. In fact, PoP/PoP+ adopts simplest Merge (Merge(A,B) = {A,B}), which was defended in C-S. In addition, the labeling algorithm can be seen as a reply to the question implicitly posed by C-S: If there are no labels, how is the information traditionally encoded by labels derived?

Acknowledgments: We thank Erich Groat, Hisa Kitahara and Dennis Ott for helpful comments on an earlier draft of this paper.

References

Berwick, Robert, and Noam Chomsky 2016. *Why only us: language and evolution.* The MIT Press.

Chomsky, Noam. 1981. *Lectures on government and binding*. Dordrecht: Foris.

Chomsky, Noam. 1995. The minimalist program. Cambridge, MA: MIT Press.

Chomsky, Noam. 2000. Minimalist Inquiries. In Roger Martin, David Michaels and Juan Uriagereka (eds.) *Step by step*, 89–155. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 2005. Three Factors in Language Design. *Linguistic Inquiry* 36, 1-22.

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

Chomsky, Noam. 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.

Collins, Collins. 2002. Eliminating Labels. In Epstein, S. D. and Seely, T. D. (eds.), Derivation and Explanation in the Minimalist Program, 42-64. Oxford: Blackwell.

Collins, Chris. 2017. Merge(X,Y) = {X,Y}. In Leah Bauke, Andreas Blümel, and Erich Groat (eds.), *Labels and Roots*, 47-68. De Gruyter Mouton.

Collins, Chris and Edward Stabler. 2016. A Formalization of Minimalist Syntax. *Syntax* 19, 43-78

Epstein, Samuel D., T. Daniel Seely, Hisatsugu Kitahara. 2015. *Explorations in Maximizing Syntactic Minimization.* Routledge Leading Linguists Series. Routledge.

Rizzi, Luigi. 2015. Cartography, criteria, and labeling. In Ur Shlonsky (ed.), Beyond the functional sequence, 314–338. New York: Oxford University Press.

Seely, T. Daniel. 2006. Merge, Derivational C-command, and subcategorization in a label-free syntax. In Boecks, C. (eds.), *Minimalist Essays*, 182-217. Amsterdam: John Benjamins.