Bare Phrase Structure and Specifier-less Syntax

K.A. Jayaseelan

It is pointed out that "specifiers" render the algorithm of projection overly complex. This consideration lends support to Starke's (2004) reanalysis of specifiers as phrasal heads that project their own phrases — which makes phrase structure a simple sequence of head-complement relations. It is further pointed out that if head-complement relations are represented using dominance in place of sisterhood, to reflect the essentially asymmetrical nature of Merge (Chomsky 2000), a non-branching (partially linear) phrase structure tree is obtained that very naturally eliminates labels and projections. A simple Spell-Out rule then provides a linear ordering of the terminal elements. The linear tree preserves all the major results of antisymmetry.

Keywords: antisymmetry, label, linearization, specifier

1. Introduction

In this article, I suggest a notational innovation in the representation of phrase structure trees (henceforth, PS trees), taking as its background the assumptions of *bare phrase structure* (Chomsky 1995) and *specifier-less syntax* (Starke 2004). This innovation makes PS trees radically simple, and linear.

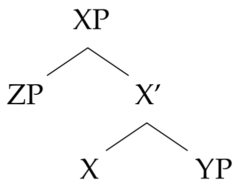
2. Traditional X'-Syntax and the Notion of Specifier

Phrase structure is represented by the following schema in X' syntax:

For helpful comments at various stages of the growth of the ideas presented here, I wish to thank R. Amritavalli, Tanmoy Bhattacharya, Probal Dasgupta, M.T. Hanybabu, Ayesha Kidwai, P. Madhavan, Rosmin Mathew, Daniel Seely, Gautam Sengupta, and Michal Starke. I am also grateful to an anonymous reviewer of this journal for comments that helped me give this paper its final shape.

ISSN 1450–3417
Biolinguistics 2.1: 087–106, 2008
http://www.biolinguistics.eu

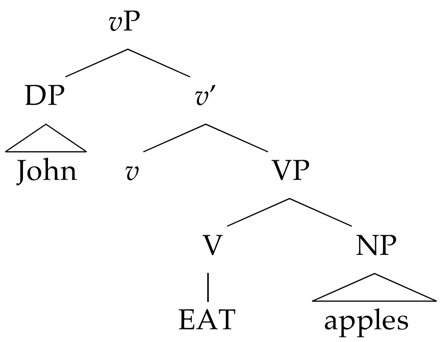
(1)



This embodies the claim that a head can be merged with two phrases, the first merge giving the head a complement (YP) and the second merge a specifier (ZP).

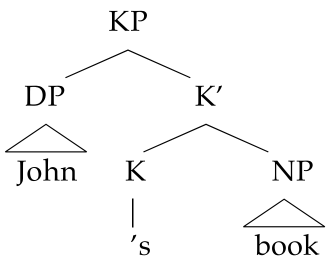
It is, as a matter of fact, not so straightforward to find a case where all three terms — head, complement, specifier — are lexically filled. *Prima facie*, a likely example of this might appear to be a verb phrase consisting of a transitive verb and two arguments, such as *John eats apples*. But this linguistic expression is now commonly represented as:

(2)



Here the lower verb (lexical V) has only a complement; and the higher verb (light verb *v*), which has a complement and a specifier, is (itself) an abstract element.[1] Outside lexical VP, auxiliary verbs have no specifiers; and if adverbial modifiers are in specifier positions of AdvPs (Cinque 1999), the AdvPs have abstract heads. PPs famously have no specifiers. TP and CP, commonly thought of as structural configurations with all three terms, may in fact not be such, as I presently show. If we leave aside TP and CP, the only examples one can readily think of which have all three terms are, in fact, phrases headed by inflectional elements; for instance, *John's book*, which can be argued to have the following structure:[2]

(3)



1 A double object VP may or may not provide an example of all three terms lexically filled, depending on the analysis assumed. Thus, in *give Mary a book*, if [*Mary a book*] is analyzed as a small clause, it must have an abstract head (given antisymmetry). However, there are other analyses in which the structure proposed has (at least initially) all three terms lexically filled (see Larson 1988, among others).

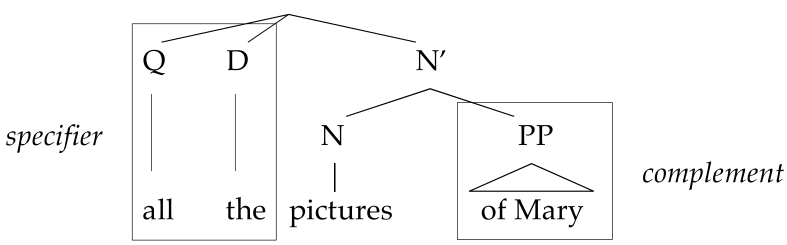
2 See Abney (1987) who, however, analyzes *’s* as a D^0, and the whole structure as a DP.

Possibly motivated by this paucity of examples of phrases with all three terms lexically/phonetically filled, Koopman (1996) proposed a condition that in a phrase, the specifier and the head cannot both be lexically filled at Spell-Out, and tried to derive this result from a modified version of antisymmetry.

Koopman's concern is addressed in a different way by specifier-less syntax, which we come to directly.

It may be useful to recall that "specifier", when Chomsky (1970) first introduced the notion into linguistic theory, was only a "residual category" consisting of all the phrase-internal elements to the left of the head. ("Complements", which were the categories that a head strictly subcategorized for, conveniently came — in English — to the right of the head.) It typically consisted of single-word elements; and when there was more than one of these elements, they could only be treated as "a concatenation of nodes" (Jackendoff 1977: 40).[3] For example, a phrase like *all the pictures of Mary* — if we took *pictures* to be the head of the phrase — could reasonably be represented only as (4):[4]

(4)

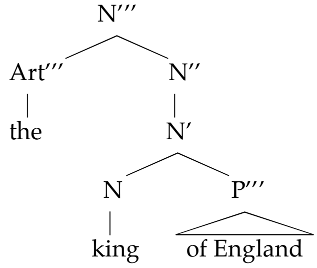


Arguably, it was Abney's (1987) "DP hypothesis" that changed this picture. Each of the single-word elements which were earlier grouped under the rubric of "specifier" now projected its own phrase, and took the phrase projected by the next element as its complement. For example, (4) became (5):

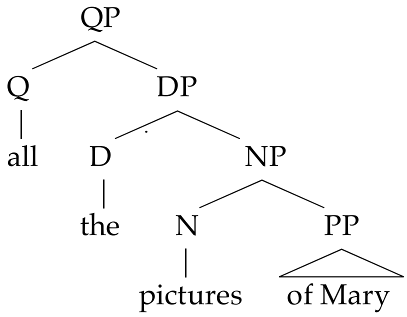
3 As Jackendoff (1977: 14) points out, it is unclear if Chomsky considered the various elements in the specifier to be a constituent; although in his diagrams Chomsky does show them under a single node labeled "specifier", see Chomsky (1970: 211).

4 Jackendoff himself, however — in obedience to his proposal of a "three-tier" X′-schema for every category — treated these single-word elements as phrases; cf. (i), which is adapted from one of his diagrams (Jackendoff 1977: 59):

(i)



(5)



The idea that specifier is phrasal had perhaps already been gaining ground prior to this development; cf. Stowell's (1983) "subjects across categories" and Chomsky's (1986) extension of the X′-schema to clausal categories — but the treatment of single-word elements like the definite article still remained a problem, until Abney's work enabled us to treat them as heads. The term "specifier" was now reserved for a phrase which occurred to the left of a projecting X^0-element; since it seemed inconceivable that a phrase could project, it was now analyzed as the specifier of the following X^0-element. If there was more than one such phrase, one had to say that they were multiple specifiers, or postulate an abstract X^0-element ("head") intervening between the phrases.

But in this schema, what interaction was postulated between a specifier and the head? And perhaps more relevantly, what interaction is postulated now? "Specifier" has a very ambiguous status in this regard at the current stage of the theory. Chomsky (2004: 111-112) claims that "a Head–to–Spec relation […] cannot exist (nor the broader symmetric Spec–Head relation, in the general case)."[5] The only relation that is countenanced is a Spec–to–Head probe: Chomsky (2000: 124-125) suggests that an expletive merged in [Spec,TP] checks an uninterpretable feature of T^0 by a probe. Bošković (2007) exploits the same device in his analysis of movement. But a probe only needs c-command and locality; it does not require the "special" relation of a specifier to its head. More specifically, a probe need not be contained in a projection of the goal. So in effect, in the current theory, there is no interaction between a specifier and the head as specifier and head.

Consider the claim that the subject in an English-type language is in [Spec,TP]. But the only relation that the subject has to T^0 is that of fulfilling an EPP requirement of T^0. EPP is only a diacritic which says that a certain head "needs a specifier". Rizzi (2005) has suggested that we can make sense of this requirement in the case of TP if we say that a SubjP (i.e. a subject position) is obligatory in the functional sequence that constitutes IP.[6] We note that Rizzi's SubjP is a separate projection above TP, and that it has an abstract head. Given such an analysis, we can no longer cite TP as an example of a lexically instantiated specifier–head–complement sequence.

---

5 This makes v^0 "assigning" a theta role in its Spec position problematic; see Chomsky (*ibid.*) for discussion. Den Dikken (2006: 22-23) points out that v^0, as it is currently conceived, is a "hybrid element" which is partly functional (in virtue of its parametrically variable morphological features) and partly lexical (because it assigns a θ-role); and he suggests that it should be treated as a purely functional category that does not assign any θ-roles.

6 See also Cardinaletti (2004) for the notion of a SubjP.

Consider another commonly cited example of a Spec–Head configuration, namely a *wh*-phrase in [Spec,CP]. The C^0 here is in itself lexically null; but in English root questions, a tensed auxiliary verb is assumed to move into the C^0-position, either adjoining to it or substituting into it. Here then, one could say, is a clean example of an X′-configuration with all three terms — specifier, head, and complement — lexically filled. But unfortunately for this analysis, it has since been shown that there is no single C^0-head, but several functional projections, in the C-domain (Reinhart 1981, Bayer 1984, Rizzi 1997), and that the English *wh*-phrase (when it moves) moves into a Focus Phrase in that domain (Rizzi 1997). Now it is not certain that the auxiliary verb moves into the head position of this FocP. Depending on how high (in the C-domain) FocP is generated, and how many functional heads can be generated below it, the auxiliary verb has other possible adjunction sites, such as the head of Finiteness Phrase. (Incidentally, this FinP appears to never have a lexically-filled Spec; and FocP — unless we analyze the inverting auxiliary verb as moving into its head position — never has a lexically filled head.)

A third, at first glance strong, argument for a Spec–Head configuration, it might seem, is provided by phrases headed by inflectional elements; e.g., a Case Phrase (KP) headed by a Case morpheme that "requires" a nominal expression to its immediate left. We have already drawn attention to this type of evidence, in (3). Currently this is handled by moving a DP/NP into [Spec,KP]. But the dependency between the nominal expression and the Case morpheme can be expressed by a selectional relation between independent phrases, as Starke (2004) has shown.[7]

In section 5, I will show that the notion of "specifier" introduces a possibly unacceptable degree of complexity into any set-theoretical characterization of the operation Merge, making the notion costly and unintuitive.

3. Bare Phrase Structure (BPS)

Improving on the traditional way of representing phrase structure, Chomsky (1995) proposed that category labels can be eliminated from syntactic representations. In his theory of "bare phrase structure" (henceforth, BPS), the head of a phrase is used as the label of its projections. Thus the VP *eat apples* of (2) will now be represented as:

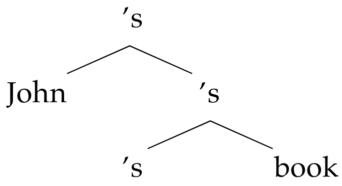
(6) eat

eat apples

A phrase with a lexically filled specifier will be represented as shown in (7):

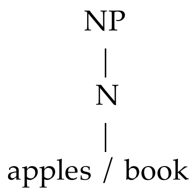
[7] Starke has a notion of "dependent insertion" to cover these cases, and also such cases as the dependency between *wh*-movement and auxiliary inversion in English — more generally, the verb–second phenomenon of Germanic; see Starke (2004) for details.

(7)



The representations in (6) and (7) are remarkable not only for the absence of category labels. Note that *apples* in (6), or *book* in (7), is both N^0 and NP; in the traditional representation, this lexical element would be represented with at least the structure shown in (8):

(8)

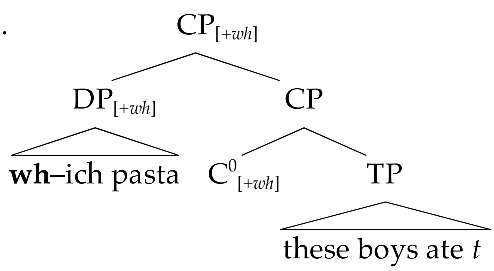


But in BPS, there are no non-branching projections. Chomsky achieves this result by proposing a relational definition of “minimal” and “maximal” projections: A category that does not project any further is “maximal”, and one that is not a projection at all is “minimal”. By this definition, *apples* in (6) or *book* in (7) is simultaneously N^0 and NP.

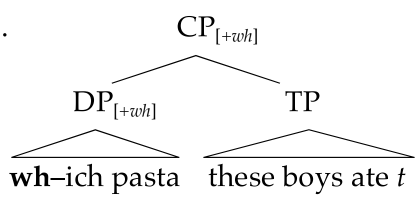
4. Specifier-less Syntax

In a recent paper, Starke (2004) has argued that “specifiers” don’t exist, and that what has hitherto been analyzed as a specifier is a phrase which projects its own, independent phrase. An example is the following, taken from Starke (2004: 252), which shows *wh*-movement represented in the traditional way (9) and in Starke’s theory (10):

(9) I wonder …



(9) I wonder …



In (9), an “invisible head terminal” attracts a *wh*-phrase to its specifier position, and checks its own [+*wh*] feature with that of the moved phrase. In (10), the [+*wh*] feature of the *wh*-phrase directly labels the projection. To legitimize (10), Starke argues, all we need to do is to discard a hidden assumption of the current theory that only X^0 can project. Adopting (10), we eliminate two things: an invisible head and a duplication of features.[8]

In Starke's theory, the *wh*-phrase moves in order to conform to a universal functional sequence ("f-seq") which requires that there should be a phrase bearing the [+*wh*] feature above TP in a question. The mechanisms of Checking Theory (Chomsky 1993) — such as the uninterpretable feature [+*wh*] on the invisible head and the Extended Projection Principle (EPP) — can be dispensed with (but see fn. 9 below).

Note that in (10), the wh-phrase is a phrasal head that takes the TP as its complement. In Starke's theory, phrase structure is radically simple: "[…] syntactic structures are nothing but raw layers of head-complement relationships" (Starke 2004: 264).[9]

8 How can a phrase project? Note that the *wh*-feature, however deeply embedded it is in the *wh*-phrase, must be accessible from outside for selectional processes; otherwise the phrase will not have been "pulled up" into the C-domain in the first place, and it will not satisfy the checking requirements of $C^0_{[+wh]}$ in the traditional configuration. If the feature is "salient" in this fashion, it should not be surprising that the *wh*-phrase can directly satisfy the English question clause's requirement of a *wh*-phrase in its left periphery by projecting this feature.

This should also answer the possible query why *which pasta* in (10) projects its [+*wh*] feature, and not, say, its D-feature. What the position requires is a [+*wh*] phrase.

9 Starke's proposal about *wh*-movement is arguably too cryptic. It ignores many questions — for example: How does successive-cyclic movement take place?

Let us try to fill this lacuna by considering Bošković's (2007) proposal, which contains a careful articulation of the problems involved and proposes a solution; and let us show that the Bošković-solution can be "adopted" into specifier-less syntax. Bošković suggests that movement is not target-driven but driven by an uninterpretable feature on the moving element. Thus, a *wh*-phrase has, say, an uninterpretable feature [*u*K], which must be deleted by an interrogative C^0 with a matching interpretable feature. But [*u*K] must probe C^0, exactly as the uninterpretable features of T^0 must probe an NP with matching features. Since probe is always "downward", [*u*K] must move to a position above C^0; in Bošković's proposal, the *wh*-phrase moves to [Spec,CP]. Successive-cyclic movement is ensured as follows: Given Phase Theory (Chomsky 2000 *et seq.*), the *wh*-phrase must move to the edge of each phase it must escape from, before the complement of the phase head is transferred to Spell-Out; otherwise it will be "frozen in place", and [*u*K] will never be deleted.

In Starke's system, let us say that — modifying it somewhat by importing into it *some* of the mechanisms of Checking Theory — a *wh*-phrase can be marked with an uninterpretable focus feature [*u*Foc], and that this feature will be deleted when the phrase moves into a focus position in the left periphery of an interrogative clause (cf. Rizzi 1997). (The Focus position, we shall say, has an interpretable feature [*i*Foc].) Successive-cyclic movement will be ensured by the same considerations as in the Bošković proposal.

Note that we are saying that [*u*Foc] is deleted — that is, feature matching takes place — simply as part of the movement of the *wh*-phrase into the focus position. (In effect, Move and Agree use different mechanisms to delete uninterpretable features.) But we can also imitate the Bošković system more closely and employ a probe. We can say that the focus position in the left periphery of interrogatives into which the *wh*-phrase moves does not itself bear [*i*Foc], but that it is generated above a head — possibly the head hosting the question operator — which bears [*i*Foc], and that [*u*Foc] probes this head. (Recall our earlier point that a probe needs only c-command and locality; see also Jayaseelan 2007 for some discussion of a focus-above-question-operator configuration — although I use the notion of "specifier" in that paper for convenience.)

5. Eliminating Labels

Returning to the BPS representation, consider (6) again:

(6) eat

eat apples

Prima facie, *eat* not only takes as its sister *apples* in (6), but dominates the string *eat apples*. Similarly, in a phrase which contains a specifier, such as (7), the head dominates a string that contains the specifier as well as the complement. How should we understand this?

In the traditional way of representing phrase structure, domination — more correctly, exhaustive domination — signified an "is a" relation. For example, in (8), *apples* (or *book*) "is a" N(oun) and "is a" N(oun) P(hrase). What does domination signify in (6)? The lexical element *eat* contains the categorial feature [+V]. So the "is a" relation is recoverable in (6). Instead of "extracting" the categorial feature of the head and using it as a label, BPS uses the head itself as a label, which is arguably computationally simpler. As Chomsky (1995: 396) is at pains to point out, all the information needed for further steps in the derivation — e.g., in the case of (6), selection of *eat apples* by the higher head v^0 (or whatever is the higher head that selects it) — is present in the label. Thus the label minimizes search.

However, in a proposal that is currently receiving serious attention, Collins (2002) argues that labels (and projections) ought to be eliminated from phrase structure representations.[10] For Collins, (6) should be replaced by (11):

(11)

eat apples

In set notation, whereas (6) would be represented by Chomsky as (12), Collins wants only (13):

(12) { eat, { eat, apples }}

(13) { eat, apples }

Collins adopts a theory of "saturated" and "unsaturated" constituents from earlier researchers. In (11) (or (13)), there are two terms (besides the whole phrase, which is a term). Of these, one term, *apples*, is saturated, because it has no feature which is "unsatisfied".[11] But the other term, *eat*, is (by itself) unsaturated, because it needs an argument to satisfy (what we can think of as) a "theta-role feature". Therefore *eat* selects *apples*, and not vice versa. (This is what we mean

---

10 See also Seely (2006) for an elaboration of this idea.

11 An unchecked (unvalued) Case feature does not make a nominal phrase "unsaturated", Collins maintains; therefore *apples* — even prior to being concatenated with *eat* and getting its Case feature checked (valued) — is saturated.

when we say that *eat* is the "head" of *eat apples*.) Now in any act of binary Merge, one member will be the selector (unsaturated) and the other will be the selectee (saturated). And the computation can tell which is which by only inspecting the two objects that are merged. Therefore, Collins argues, labels are not necessary.

But the computation's task — one may want to point out — becomes more difficult when a specifier is merged with an intermediate projection X′; because now it will have to look "into" the X′ constituent to realize that this constituent is unsaturated. (It is a remaining unsatisfied feature of X^0 — e.g., an EPP feature of T^0 — that induces Merge of the specifier.) However, we can let this pass, because this is not our main problem with Collins' proposal.

It seems to me that it is a function of notation, whether we are using the graphic notation of PS trees or the set notation, to express the *unequal* relation that obtains when two syntactic objects are merged. It is a relation which has directionality: One object is the "pivot", it selects the other.[12] Neither (11) nor (13) expresses this. Observe that (13) is an unordered set. But what we need in this case is an ordered pair, in which the ordering reflects the directionality of the relation.

As is well-known, an ordered set can be represented in terms of unordered sets:

(14) $(\alpha, \beta) \equiv \{ \{ \alpha \}, \{ \alpha, \beta \}\}$

Consider the Chomsky-type representation (12), which we repeat here:

(12) { eat, { eat, apples }}

It is tempting to make a small change in (12), as shown in (12′), and suggest that Chomsky's "label" (or "head") is simply a way of indicating that the set we are dealing with is an ordered pair:[13]

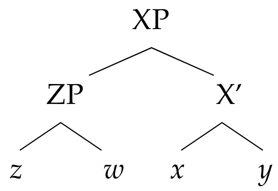
(12′) {{ eat }, { eat, apples }}

Such a suggestion becomes impossible, however, when we deal with a phrase which has a specifier. Consider (15), from Chomsky (1995: 398):

12 Cf.: "Set-Merge typically has an inherent asymmetry. When α, β merge, it is to satisfy (selectional) requirements of one (the *selector*) but not both" (Chomsky 2000: 133).
To emphasize what is perhaps an obvious point: It is not enough that the native speaker, looking at any instance of merge, can tell apart (implicitly knows) the selector and the selectee. The function of linguistic representation is to make explicit the native speaker's knowledge. The traditional phrase structure representation, and also Chomsky's version of BPS, indicated the selector by means of projection and labels. With the elimination of labels, the unequal nature of Merge is unrepresented.

13 Daniel Seely (p.c.) has pointed out that Chomsky could not have adopted (12′), for a good reason: In (12′), both occurrences of *eat* become "terms", going by the "member of a member of the set" definition of "term" (Chomsky 1995).

(15)



Here *z*, *w*, *x*, *y* are terminals; ZP = {*z*, {*z*, *w*}} and X′ = {*x*, {*x*, *y*}}. Up to this point, we can maintain — with a small change along the lines of (12′) in the set representation — that the notion of "head" can be derived from the notion of an ordered pair.

But what is XP? If the notion of "head" is definable in set-theoretical terms as the first member of an ordered pair, we should get (16); but what Chomsky has is (17) (see the discussion of (15) in Chomsky 1995):

(16) {{{ *x*, { *x*, *y* }}}, {{ *z*, { *z*, *w* }}, { *x*, { *x*, *y* }}}}

(17) { *x*, {{ *z*, { *z*, *w* }}, { *x*, { *x*, *y* }}}}

Therefore the notion of "head" is only a linguistic notion, not a set-theoretical notion at all.

How do we get (17)? Consider the stage at which ZP and X′ have been merged, and we have still to find the label:

(18) ?{{ *z*, { *z*, *w* }}, { *x*, { *x*, *y* }}}

We cannot have an algorithm which copies "a member of a member of the set", for this could as well copy "{*x*, *y*}" or "{*z*, *w*}". We need (19):

(19) Copy a member (which is itself not a set) of a member of the set.

If *z* is copied, the constituent shown as X′ in (15) becomes the specifier of ZP. But in fact *x* is copied, and we get (15) (= (17)).

But (19) is overly complex.[14] Note that in a theory like that of Starke (2004) in which "specifiers" are phrases that project, we can have a very simple algorithm, namely the algorithm that generates an ordered set:

(20) Copy a member of the set.[15]

If (20) applies to (18), it can copy "{*z*, {*z*, *w*}}"; in which case "{*z*, {*z*, *w*}}" would be a "phrasal" head that takes "{*x*, {*x*, *y*}}" as its complement. If "{*x*, { *x*, y}}" is copied (instead), the relation would be reversed.

---

[14] Also, (19) by itself is inadequate, since we need the following rule for merging a head and a complement:

(i) Copy a member (which is itself not a set) of the set.

[15] More strictly: "Copy a member of the set and make it the member of a singleton set."

What (19) points to is not really a difficulty about finding a label (which can be got around by doing away with labels), but a deeper difficulty that inheres in the idea of "second merge". "Second merge" requires the activation, and accessing, of an element embedded in one of two phrases that merge to create the specifier configuration. This element — an unsaturated X^0 element — can be an immediate constituent of the merging phrase that contains it, but it can also be very deeply embedded in that phrase if we are dealing with multiple specifiers. There is a plausibility argument here for doing away with the "specifier" relation. It is likely that Merge, the basic operation of syntax, only makes sets by looking at the immediate properties of the two syntactic objects that merge, that it does not also set in motion a search algorithm that looks deep into these syntactic objects.

6. BPS Further Simplified

The elimination of labels (and the consequent simplification of Chomsky's version of BPS) can, in fact, be achieved in a radically simple way; (6) can be represented as:

(21) eat
  \
    apples

(21) has only terms, no labels. But unlike in Collins (2002), the unequal relation between the selector and the selectee is encoded in terms of dominance.[16] The "is a" relation is recoverable in (21), in the same sense in which it is recoverable in Chomsky's version of BPS, i.e. (6): *Eat* contains the feature [+V]; therefore a structure "headed" by *eat* is a V(erb) P(hrase).

The standard PS tree has three relations: dominance, precedence, and (derivatively) c-command. But our representation (21) has only one relation, which we can think of in terms of dominance, or precedence (see fn. 16), or whatever other ordering device we choose.

But what happens if the "head" is a phrase, as can be the case in specifier-less syntax? Consider (22), which will be represented by Chomsky's version of BPS as (23):

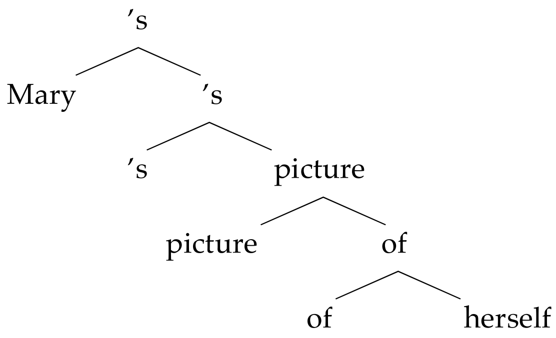
(22) Mary's picture of herself

16 Any way of indicating an ordering relation will do, including precedence:

(i) *eat – apples* (or: *eat*ˆ*apples*)

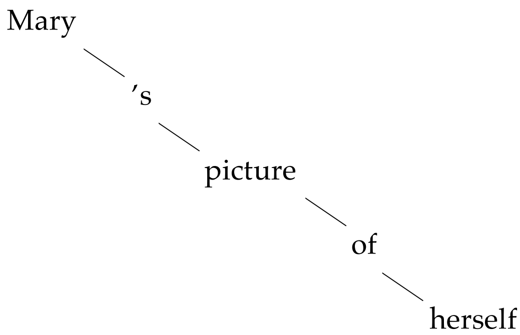
But we shall choose to use dominance in our illustrative examples. The notion of representing the head-complement relation as dominance has in fact a tradition in linguistics, see, e.g., Brody's (1997) "Mirror Theory". (Brody credits the idea to dependency grammar; see, e.g., Hudson 1990.)

(23)



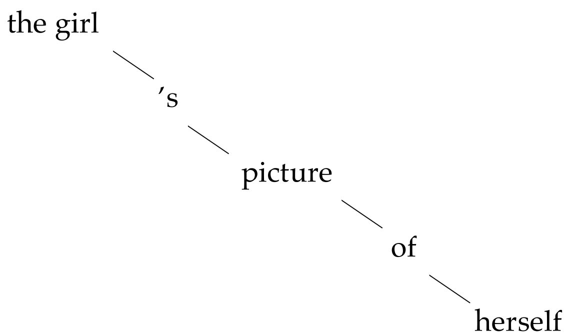
Here *Mary* is treated as a specifier. But if *Mary* is a phrasal head, and if we apply the logic of (21) to this phrase, the representation that we get is:[17]

(24)



How about *the girl’s picture of herself*? Note that *the girl* is not built up as a continuation of the “derivational cascade” (Nunes & Uriagereka 2000) that built up the rest of the phrase *’s – picture – of – herself*. It was built up in a different derivational space and merged as a phrase. We can encode this fact by representing it in the larger phrase as follows:

(25)



Let us stop to consider (25). It embodies a claim that there can be complex mother nodes, with internal structure. Two questions immediately arise: First, how do we make sense of the notion of a phrasal mother node? Second, how can this structure be accommodated to our declared target of a linear PS tree?

---

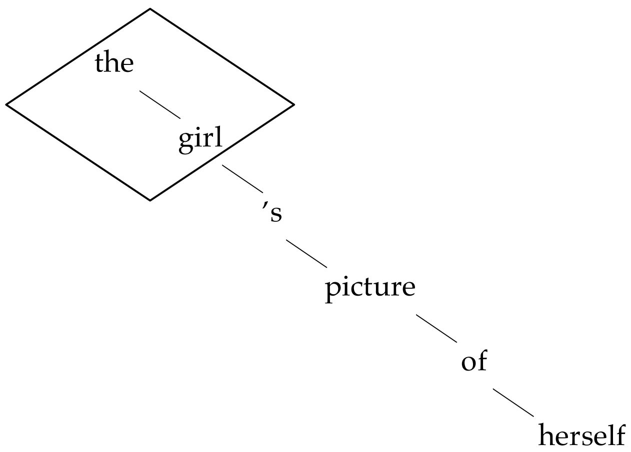
[17] We abstract away from the question whether *Mary* here is in its base position or moved up from a lower position in the phrase. *Mary* selects *’s*, perhaps in order to satisfy a Case feature. (See also fn. 7.)

To answer the first question: In the traditional PS tree, the mother node — bearing a categorial label — signified an "is a" relation with respect to the string it exhaustively dominated. (We said this earlier.) The Chomskyan version of BPS dispensed with any explicit representation of the "is a" relation; although, as we suggested, this relation could be recovered from the categorial feature contained in the label of the mother node. In contrast to both these systems, in our system the mother node–daughter node relation signifies the head–complement relation. Our departure from earlier attempts in the theory to use dominance to represent the head–complement relation (see fn. 16) is that — following the central claim of specifier-less syntax — we postulate phrasal heads. So it should not be surprising that we have phrasal mother nodes. This should be even less surprising if we think in terms of set representation: nothing prohibits the first member of an ordered pair being itself a set.

Now with respect to the second question: The tree in (25) is *not* linear — at least, not yet. While *the girl* stands in an ordering relation of dominance to the elements below it, the proper terms of that phrase — *the* and *girl* — stand in no relation to the elements below it. The total linear ordering of the terminal elements of the PS tree is a question that we take up in section 7, where it is implemented by a rule of Spell-Out. But in the meanwhile, what (25) achieves should not be lost sight of: We have here represented the head–complement relation in an asymmetrical fashion, correctly reflecting the asymmetrical nature of this relation; moreover, this representation very naturally eliminates projection and labels.

It should be pointed out further that the phrase *the girl* is internally ordered by the relation of dominance, so that we could equally well have represented (25) as (26):[18]

(26)



[18] The function of the box drawn into (26) is only to preclude the possible misunderstanding that *girl* takes (the structure headed by) *'s* as its complement. The box is not a theoretical construct that we need (or make use of); it is not "real".

(26) already indicates why it is "easy" for the Spell-Out rule to achieve total linear ordering; all it has to do is to "wipe out" the box! ((26) ought to also dispel any possible suspicion that by admitting complex (phrasal) mother-nodes, we are covertly making use of the c-command relation.)

It will be recalled that in the theory of specifier-less syntax, the erstwhile specifier becomes a phrase that, as a whole, takes the phrase it is merged with as its complement; but, of course, none of its subparts (proper terms) takes the latter phrase as its complement. Thus, *the girl* can take the KP headed by *'s* as its complement — but that operation does not make the KP the complement of *the* or *girl*.

In order to implement this idea in terms of dominance, we can adapt Epstein's (1999) idea of "derivational c-command" and speak of "derivational dominance":

(27) *Derivational Definition of Dominance*

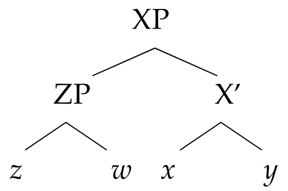
If α is merged with β, α the selector, α dominates all the terms of β.

(27) does not mention the terms of α; so these do not dominate β's terms. And since domination is an antisymmetric relation, no question arises of a reciprocal domination by β of α's terms. Also, it is important to note that any element which may now be merged above the structure shown in (25) or (26) will dominate *the* and *girl* separately; that is, a merged phrase is an unanalyzed unit (in effect, a "word") for the elements below it but not for the elements above it.

The definition (27) gives us the right result for an ungrammatical phrase like **Mary$_i$'s brother's picture of herself$_i$*, wherein *herself* cannot take *Mary* as antecedent. The explanation now is that only *Mary's brother* dominates *herself*, not *Mary*. The reader can readily see that the relation of dominance does all the work of the erstwhile relation of c-command.

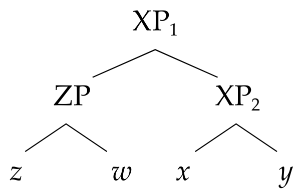
In fact dominance does better than c-command, because it avoids certain problems created by c-command. Consider (15), repeated here for convenience:

(15)



If we adopt the "first branching node" definition of c-command (Reinhart 1979), X' c-commands ZP, *z*, and *w*. This is an unwanted set of relations; there is no positive evidence of the existence of these relations. For Kayne (1994), these relations also created counterexamples to antisymmetry, which is why he re-analyzed specifiers as adjoined phrases:

(15')



He claimed that a mere segment of a category — in (15'), XP$_2$ — does not c-command. Chomsky (1995) (see also Epstein 1999) stipulated that an intermediate projection does not c-command, but required that (nevertheless) the intermediate projection has to be present in the tree to prevent — in (15) — *x* and *y* from c-commanding the terms of ZP. All these complications arose, one can now see, because of an inadequate graphic representation that showed syntactic objects that merge in a symmetric relation (as sisters) on the one hand, and an analysis which claimed that "specifiers" are in a selectee relation to a following X^0 category on the other. In our analysis, (15) becomes (15′′):

(15′′) ZP

$$\begin{array}{lll} \mathrm{ZP} & & \\ & \diagdown & \\ & & x \\ & & & \diagdown \\ & & & & y \end{array}$$

There is no question here of *x* or *y* dominating the terms of ZP.

At this point, we wish to dispel a possible misconception that may have arisen. Our discussion so far led up from specifier-less syntax to a proposal about non-branching PS trees. But the two are, in fact, independent issues. Our non-branching PS tree is not contingent on the elimination of "specifier" from the grammar. Thus, consider (15″) again. In this configuration, ZP can still be analyzed as a specifier, if one so wishes. (One can define "specifier" derivationally as a "second merge", or in some other way, exactly as before.[19]) That is, given the possibility of phrasal mother nodes, *any* analysis of phrase structure that does not crucially appeal to left–to–right ordering can be "translated" into a non-branching PS tree. All one has to do is to "push up" all the constituents on left branches into the "bole" of the tree. Therefore, a non-branching PS tree is not in itself a very interesting idea, and it is not the core of my claims about phrase structure representation. But note that the "pushing up" operation changes relations: The left-branch constituent is no longer in a symmetrical relation with the right-branch constituent, and this is what is significant. The substance of my proposal about phrase structure is, then, that Merge should be *asymmetrically* represented.

7. Linearizing the Terminal String: A Rule of Spell-Out

Note that while our theory yields a partially linear PS tree, we do not yet have a linear ordering of the terminal elements. To see this, consider again (25) or (26). In this structure, I insisted that, while they are ordered *inter se* by the relation of dominance, the terms *the* and *girl* of the merged phrase *the girl* have no dominance relation with respect to the terms of the constituent below the phrase. But linear ordering must be total; that is, in the present case, for any terminal elements *x*, *y*, it must be the case that either *x* dominates *y* or *y* dominates *x*.

To obtain a total ordering of the terminal elements, let us propose a rule

[19] But now, of course, dominance will no longer uniformly represent the head–complement (selectional) relation.

that applies in Spell-Out:

(28) *Rule of Spell-Out*

If α dominates β, the terms of α dominate β.

((28) in effect "wipes out" the box in (26)!)

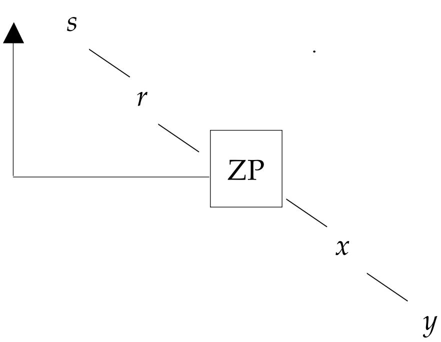
Linearization of the terminal elements, then, is a matter of the PF component of the grammar (Chomsky 1995).

8. Movement in a Linear Tree

How do we do movement in a linear tree? In a traditional PS tree, a specifier "hung out" conveniently in a left branch, so that it could be moved (leaving a trace) without disturbing the rest of the tree. A head X^0 also was on a left branch, and so could be similarly moved — if one wanted head-movement — without disturbing the rest of the tree. The movement of a complement presented no problem whatever, since one was only moving a constituent from the bottom of the tree.

In a linear tree, all but movement from the bottom of the tree (corresponding to complement movement) appears *prima facie* to be problematic. Consider (29):

(29)



Does the movement of ZP "disconnect" the tree? Actually, the problem with moving ZP in (29) is that it looks like the movement of a non-constituent. *X* and *y* "depend" from ZP. How can one move a node without taking along the nodes that depend from it?

Chomsky (1993) proposed that movement is "copy-and-merge"; this is now a standard assumption of minimalist research. But the traditional PS tree is so conceived as to facilitate our thinking in terms of the physical removal of a constituent (in cases of movement). All movement is from the bottom of a tree, albeit a sub-tree. (As just said, specifier and head "hang out" from a left branch and therefore are, in that sense, at the bottom of a sub-tree.) We can see that the traditional phrase structure notation is far from innocent.

If we graduate to thinking in terms of "copy-and-merge", the question to ask is: What can be copied? Or, alternatively put: What are the constraints on copying? In this connection, let us adopt an idea of Collins (2002), that a "saturated" phrase is spelled out.[20] Let us now build on this idea and say that a spelt-out phrase can be copied. Returning to (29), if ZP is a saturated phrase and therefore spelled out, it can be copied and merged without any problem.

9. The Linear Tree and Antisymmetry

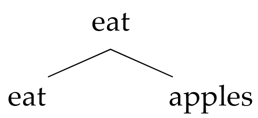
Specifier-less syntax, adopted here, is inconsistent with antisymmetry (Kayne 1994); for if XP takes YP as its complement (cf. (10), where the *wh*-phrase takes TP as its complement), XP and YP will asymmetrically c-command each other's proper terms, and linear ordering will fail.

However, we now briefly show that all the major results of antisymmetry are unaffected within our framework; in fact, these results are also predictions of the linear tree.

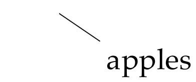
Thus consider the "Head Parameter" — that is, the claim that in UG, the head of a phrase has a choice between taking its complement to the left or to the right. The Head Parameter is inadmissible, given the antisymmetric framework. It cannot even be stated with respect to the linear tree.

In fact, no operation that crucially refers to "left" or "right" is now statable. Any seeming rightward movement of a constituent XP must be formulated (given the linear tree) as two movements: a movement of XP to the top of the tree, followed by the movement of a "remnant" to the top of XP. These are, of course, precisely the movements dictated by antisymmetry.

Chomsky (1995) pointed out that if "bare phrase structure" were to replace the traditional way of representing phrase structure, a problem would arise for the Kaynean framework: In every case where a complement is a single-word element, the linear ordering of the head and the complement will fail. Thus consider (6), and note the problem that there is no asymmetric c-command relation to invoke the LCA:

(6)


But the problem arose because of the representation of head and complement as sisters. The solution for the problem is the linear tree:

(21)


[20] See also Uriagereka (1999) and Nunes & Uriagereka (2000) for the idea that a moved phrase is spelled out prior to movement and that a spelled out phrase is treated like a "word" by the syntax.

10. Conclusion

I suggested that linguistic theory took a wrong turn when it postulated the X′-schema (1) that incorporated a relation of "specifier". I showed that the paradigm cases of a Spec–Head configuration allow, or require, other analyses. Moreover, the notion of a "second merge" introduces an arguably unacceptable degree of complexity into the algorithm of projection. Merge, the basic operation of syntax, can be maximally simple if we do away with "specifier".

We also suggested that Merge should be asymmetrically represented, to reflect the unequal relation between a selector and the selectee. We proposed that the selector-selectee relation be represented by dominance. This yielded a non-branching PS tree that imposed a partial linear ordering on the terminal elements, which could be converted into a total linear ordering by a simple operation of Spell-Out. Moreover, our partially linear tree yielded all the predictions of antisymmetry that didn't have to do specifically with the X′-schema (which I reject). The central claim of antisymmetry was that "if two phrases differ in linear order, they must also differ in hierarchical structure" (Kayne 1994: 3). This follows without stipulation in our schema, because here the linear order *is* the hierarchical structure.

Chomsky (2004: 112) has suggested that in natural language, displacement (internal Merge) is induced by "scopal and discourse-related (informational) properties". The cartographic analysis of sentence structure posits positions in a functional sequence which encode these types of meaning — e.g., TopP, FocP, SubjP. In the earlier way of representing phrase structure, we would have merged a null head marked Top^0, Foc^0, or $Subj^0$ and moved a phrase into its Spec position. But in the type of phrase structure representation argued for in this paper, we can let a phrase with the appropriate feature merge directly with the structure built up by the derivation up to that point, taking the latter as its complement, in a linear tree.

References

Abney, Steven. 1987. The English noun phrase in its sentential aspect. Cambridge, MA: MIT dissertation.

Bayer, Josef. 1984. COMP in Bavarian syntax. *The Linguistic Review* 3, 209-224.

Bošković, Željko. 2007. On the locality and motivation of Move and Agree: An even more minimal theory. *Linguistic Inquiry* 38, 589-644.

Brody, Michael. 1997. Mirror theory. Book ms., University College London.

Cardinaletti, Anna. 2004. Toward a cartography of subject positions. In Luigi Rizzi (ed.), *The Cartography of Syntactic Structures*, vol. 2: *The Structure of CP and IP* (Oxford Studies in Comparative Syntax), 115-165. New York: Oxford University Press.

Chomsky, Noam. 1970. Remarks on nominalization. In Roderick A. Jakobs & Peter S. Rosenbaum (eds.), *Readings in English Transformational Grammar*, 184-221. Waltham, MA: Ginn and Company. [Republished in Chomsky, Noam. 1972. *Studies on Semantics in Generative Grammar* (Series Minor 107), 11-61. The Hague: Mouton.]

Chomsky, Noam. 1986. *Barriers* (Linguistic Inquiry Monograph 13). Cambridge, MA: MIT Press.

Chomsky, Noam. 1993. A minimalist program for linguistic theory. In Ken Hale & Samuel Jay Keyser (eds.), *The View from Building 20: Essays in Linguistics in Honor of Sylvain Bromberger* (Current Studies in Linguistics 24), 1-52. Cambridge, MA: MIT Press. [Reprinted in Chomsky, Noam. 1995. *The Minimalist Program* (Current Studies in Linguistics 28), 167-217. Cambridge, MA: MIT Press.]

Chomsky, Noam. 1995. Bare phrase structure. In Gert Webelhuth (ed.), *Government and Binding Theory and the Minimalist Program* (Generative Syntax 1), 383-439. Cambridge, MA: Blackwell.

Chomsky, Noam. 2000. Minimalist inquiries: The framework. In Roger Martin, David Michaels & Juan Uriagereka (eds.), *Step by Step: Essays in Minimalist Syntax in Honor of Howard Lasnik*, 89-155. Cambridge, MA: MIT Press.

Chomsky, Noam. 2004. Beyond explanatory adequacy. In Adriana Belletti (ed.), *The Cartography of Syntactic Structures*, vol. 3: *Structures and Beyond* (Oxford Studies in Comparative Syntax), 104-131. New York: Oxford University Press.

Cinque, Guglielmo. 1999. *Adverbs and Functional Heads. A Cross-Linguistic Perspective* (Oxford Studies in Comparative Syntax). New York: Oxford University Press.

Collins, Chris. 2002. Eliminating labels. In Samuel David Epstein & T. Daniel Seely (eds.), *Derivation and Explanation in the Minimalist Program* (Generative Syntax 6), 42-64. Malden, MA: Blackwell.

den Dikken, Marcel. 2006. *Relators and Linkers: The Syntax of Predication, Predicate Inversion, and Copulas* (Linguistic Inquiry Monograph 47). Cambridge, MA: MIT Press.

Epstein, Samuel David. 1999. Un-principled syntax: The derivation of syntactic relations. In Samuel David Epstein & Norbert Hornstein (eds.), *Working Minimalism*, 317-345. Cambridge, MA: MIT Press.

Hudson, Richard. 1990. *English Word Grammar*. Oxford: Blackwell.

Jackendoff, Ray. 1977. *X′ Syntax: A Study of Phrase Structure*. Cambridge, MA: MIT Press.

Jayaseelan, K.A. 2007. Remarks on the two peripheries. Paper presented at the *International Symposium of the Cambridge–Connecticut–Hyderabad–Nanzan–Siena–Tsing Hua Consortium for Linguistics*, Hsinchu City. [National Tsing Hua University, 15-17 December 2007.]

Kayne, Richard S. 1994. *The Antisymmetry of Syntax* (Linguistic Inquiry Monograph 25). Cambridge, MA: MIT Press.

Koopman, Hilda. 1996. The spec–head configuration. *Syntax at Sunset: UCLA Working Papers in Linguistics* 1, 37-64. [Republished in Koopman, Hilda. 2000. *The Syntax of Specifiers and Heads* (Routledge Leading Linguists), 331-365. London: Routledge.]

Larson, Richard. 1988. On the double object construction. *Linguistic Inquiry* 19, 335-391. [To be republished in Larson, Richard. Forthcoming. *Essays on Shell Structure* (Routledge Leading Linguists). London: Routledge.]

Nunes, Jairo & Juan Uriagereka. 2000. Cyclicity and extraction domains. *Syntax* 3, 20-43.

Reinhart, Tanya. 1979. The syntactic domain for semantic rules. In Franz Guenther & Siegfried J. Schmidt (eds.), *Formal Semantics and Pragmatics for Natural Languages* (Studies in Linguistics and Philosophy 4), 107-130. Dordrecht: Reidel.

Reinhart, Tanya. 1981. A second COMP position. In Adriana Belletti, Luciana Brandi & Luigi Rizzi (eds.), *Theory of Markedness in Generative Grammar: Proceedings of the 1979 GLOW Conference*, 518-557. Pisa: Scuola Normale Superiore.

Rizzi, Luigi. 1997. The fine structure of the left periphery. In Liliane Haegeman (ed.), *Elements of Grammar: Handbook of Generative Syntax* (Kluwer International Handbooks of Linguistics 1), 281-337. Dordrecht: Kluwer.

Rizzi, Luigi. 2005. On some properties of subjects and topics. Ms., University of Siena.

Seely, T. Daniel. 2006. Merge, derivational c-command, and subcategorization in a label-free syntax. In Cedric Boeckx (ed.), *Minimalist Essays*, 182-217. Amsterdam: John Benjamins.

Starke, Michal. 2004. On the inexistence of specifiers and the nature of heads. In Adriana Belletti (ed.), *The Cartography of Syntactic Structures*, vol. 3: *Structures and Beyond* (Oxford Studies in Comparative Syntax), 252-268. New York: Oxford University Press.

Stowell, Tim. 1983. Subjects across categories. *The Linguistic Review* 2, 285-312.

Uriagereka, Juan. 1999. Multiple Spell-Out. In Samuel David Epstein & Norbert Hornstein (eds.), *Working Minimalism* (Current Studies in Linguistics 32), 251-282. Cambridge, MA: MIT Press.

K.A. Jayaseelan
The English and Foreign Languages University (EFLU)
Department of Linguistics
Hyderabad 500 007
India
jayamrit@eth.net