Bare Phrase Structure and Specifier-less Syntax

K. A. Jayaseelan
Central Institute of English and Foreign Languages, Hyderabad
jayamrit@eth.net

June 2007

(Comments welcome)

Bare Phrase Structure and Specifier-less Syntax

K. A. Jayaseelan

CIEFL, Hyderabad

Abstract: It is pointed out that adopting the position that 'specifiers' are independent, phrasal heads that project their own phrases (Starke 2004) enables us to simplify bare phrase structure further. By using dominance in the place of c-command, we obtain a non-branching (partially linear) phrase structure tree that very naturally eliminates labels and projection. A simple Spell-Out rule then gives us a linear ordering of the terminal elements.

This paper suggests a notational innovation in the representation of phrase structure trees, given the assumptions of bare phrase structure (Chomsky 1995) and specifier-less syntax (Starke 2004). This innovation makes PS trees radically simple, and linear.

1. Traditional X' Syntax and the Notion of Specifier

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

(1)
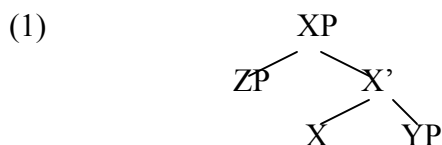
This embodies the claim that a head can be merged with two phrases, the first merge giving the head a complement (YP), and the second merge a specifier (ZP).

It is (as a matter of fact) difficult to find a case where all three terms – head, complement, and 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; e.g. *John eat apples*. But this is now commonly represented as:
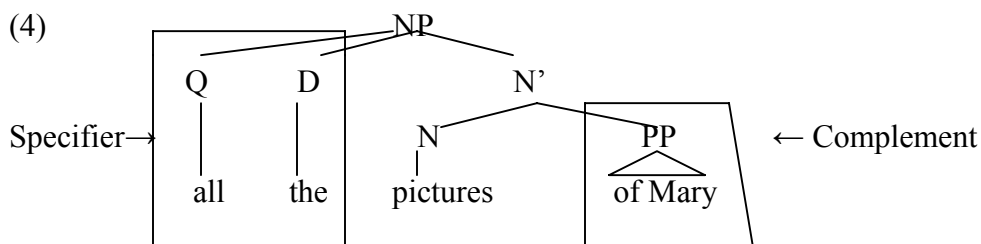
(2)
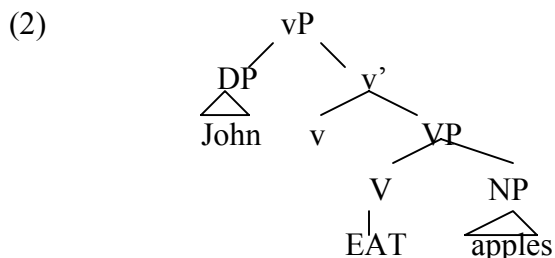
Here the lower verb (V) has only a complement; and the higher verb (v), which has a complement and a specifier, is (itself) an abstract element. Outside lexical VP, auxiliary verbs have no specifiers; and if adverbial modifiers are in specifier positions of AdvPs (Cinque 1999), the AdvPs have abstract heads. PPs famously have no specifiers. The only examples one can readily think of which have all three terms, in fact, are phrases headed by inflectional elements; for instance *John's book*, which can be argued to have the following structure:[1]

(3)

KP
DP   K'
John   K   NP
       | book
       's
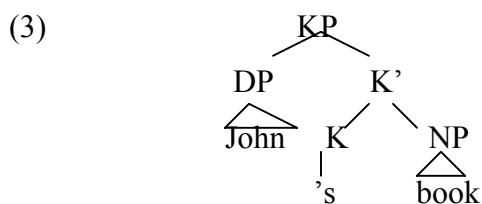
Possibly motivated by this paucity of examples of phrases with all three terms lexically filled, Koopman (1996) proposed a condition that in a phrase, the specifier and the head cannot both be lexically filled at Spell-Out; and tried to derive this result from a modified version of antisymmetry.

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

It may be well 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 was 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).[2] 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):[3]

(4)

```
                          NP
        Q        D              N'
                          N          PP
       all      the    pictures   of Mary
Specifier→                              ←Complement
```

Abney's (1986) "DP analysis" 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):

(5)

```
              QP
        Q           DP
       all      D        NP
               the    N      PP
                   pictures  of Mary
```
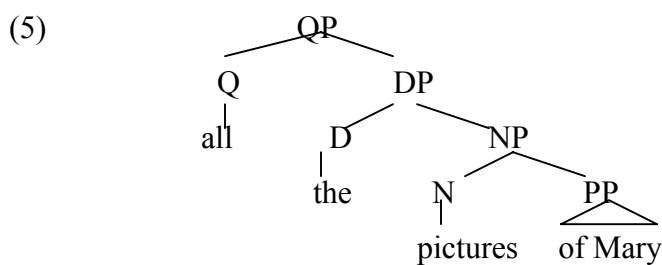
With this development, the X'-schema as we now know it "fell into place". The term 'specifier' was reserved for a phrase which occurred to the left of a projecting X^0 element; since it seemed inconceivable that the phrase could project, it was now analyzed as the specifier of the following X^0 element. 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 the fact is that, if we leave aside the case of v^0 assigning a theta-role in its specifier position,[4] there is little interaction between a specifier and the head. Consider the subject in an English-type language: a definite or specific subject is obligated to be in a position to the left of the tensed verb. The current claim is that it is in SpecTP. But the only relation that the subject has to T^0 is that of fulfilling an EPP requirement of T^0. What is more, it is unclear that the EPP feature actually belongs to T^0, and not to a "grammaticalized" Topic position above T^0. It has long been known that English has a 'high' adverb position which is below the surface position of the English subject but

above the English tense projection; cf. Baker (1981), Pollock (1989, 370, n. 8). Thus, consider a sentence like *John often is unhappy*. Since it is commonly agreed that the tensed copula is under the T-node, if *John* is in SpecTP, where do we place the adverb? If (on the other hand) *John* is in a Topic-like position above TP, one can postulate an intervening adverb position.

Consider another usually cited example of a Spec-head configuration, namely a *wh*-phrase in SpecCP. 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 tidy 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 Focus Phrase. Depending on how high (in the C-domain) the Focus Phrase is generated, and how many functional heads can be generated below it, the auxiliary verb has other possible adjunction sites, e.g. the head of Finiteness Phrase. (Incidentally, the Finiteness Phrase appears to never have a lexically-filled Spec; and the Focus Phrase – unless we analyze the inverting auxiliary verb as moving into its head position – never has a lexically-filled head.)

A 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 SpecKP. 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.[5]

In § 4 we shall show that the notion of 'specifier' introduces a possibly unacceptable degree of complexity into any set theoretical characterization of the operation of Merge, making the notion costly and unintuitive.
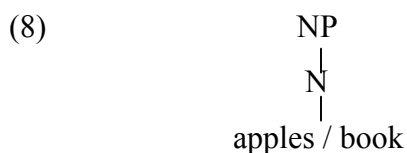
2. Bare Phrase Structure

Improving on the traditional way of representing phrase structure, Chomsky (1995) has proposed that category labels can be eliminated from syntactic representations. In his theory of 'bare phrase structure', 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)
$$\text{eat} \quad [\text{eat} \quad \text{apples}]$$
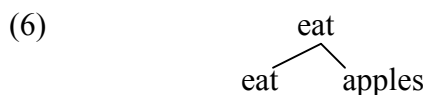
A phrase with a lexically filled specifier will be represented as shown in (7):

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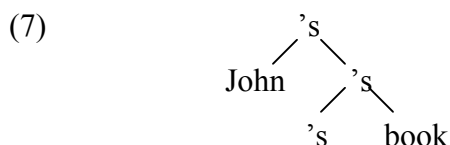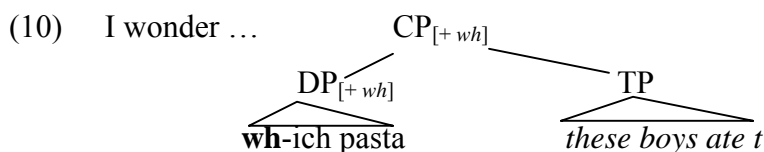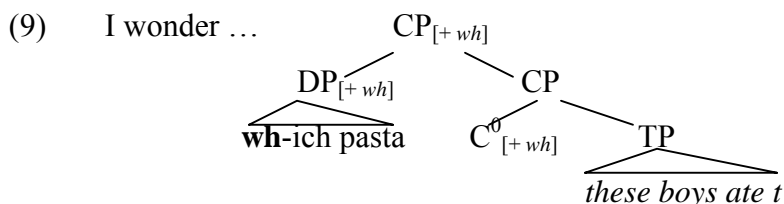
The representations in (6) and (7) are remarkable not only for the absence of category labels. Note that *apples* in (6), or *book* in (7), is both N^0 and NP; in the traditional representation, this lexical element would be represented with at least the structure shown in (8):

(8)
$$\text{NP} \mid \text{N} \mid \text{apples} / \text{book}$$

But in bare phrase structure, 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.

3. 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 (Starke's) is the following, which shows *wh*-movement represented in the traditional way and in Starke's theory ((9) and (10) = Starke's (1) and (2)):

(9)   I wonder …

$CP_{[+\,wh]}$ with $DP_{[+\,wh]}$ (**wh**-ich pasta) and CP, where CP branches into $C^0_{[+\,wh]}$ and TP (*these boys ate t*).

(10)   I wonder …

$CP_{[+\,wh]}$ with $DP_{[+\,wh]}$ (**wh**-ich pasta) and TP (*these boys ate t*).

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.[6]

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 the checking theory – such as the uninterpretable feature [+ *wh*] on the invisible head, and EPP – can be dispensed with.

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).

4. Eliminating Labels

Returning to the bare phrase structure representation, consider (6) again:

(6)
                           eat
                   eat   apples

Prima facie, in (6), *eat* not only takes *apples* as its sister, but dominates the string *eat apples*. Similarly, in a phrase which contains a specifier, cf. (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, bare phrase structure uses the head itself as a label, which is arguably computationally simpler. As Chomsky is at pains to point out (Chomsky 1995: 396), 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) has argued that labels (and projection) ought to be eliminated from phrase structure representations.[7] For Collins, (6) should be replaced by (11):

(11)
                      eat   apples
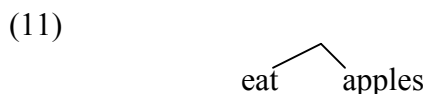
In set notation, whereas (6) would be represented by Chomsky as (12), Collins wants only (13):

(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".[8] 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 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 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 the merge of the specifier.) However we can let this pass, because this is not our main problem with the Collins proposal.

It seems to us that it is a function of notation, whether we are using the graphic notation of phrase structure 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.[9] 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, cf. (14):

(14)   (a, b) ≡ { { a }, { a, b } }

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.[10]

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

          Such a suggestion becomes impossible (however) when we deal with a phrase which has a specifier. Consider (15) (= Chomsky 1995: figure (4)):

(15)
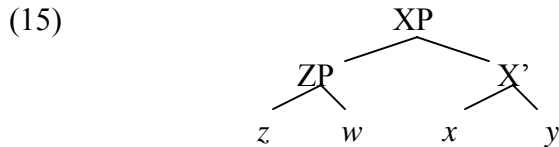
z, w, x, y are terminals. ZP = $\{ z, \{ z, w \}\}$ and X' = $\{ x, \{ x, y \}\}$; up to this point, we can maintain – with a small change on the lines of (12') in the set representation – that the notion of 'head' can be derived from the notion of an ordered pair.

          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 when 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.[11] 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.[12]

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.

There is a plausibility argument here for doing away with the 'specifier' relation. If Merge only makes sets, albeit ordered sets, 'specifier' cannot be a part of Grammar.[13]

5. Bare Phrase Structure Further Simplified

Now if specifiers don't exist, bare phrase structure can immediately be further simplified; (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.[14] The "is a" relation is recoverable in (21), in the same sense in which it is recoverable in Chomsky's

version of bare phrase structure, (6): *eat* contains the feature [+ V]; therefore a structure 'headed' by *eat* is a V(erb) P(hrase).

The standard phrase structure 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. 14), 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 bare phrase structure as (23):

(22)   Mary's picture of herself

(23)
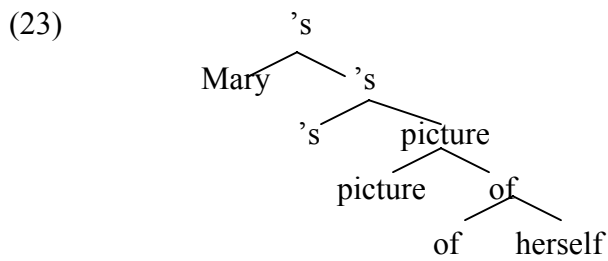
Here *Mary* is treated as a specifier. But if *Mary* is a phrasal head, and if we apply the logic of (21) to this phrase, the representation that we get is:[15]

(24)
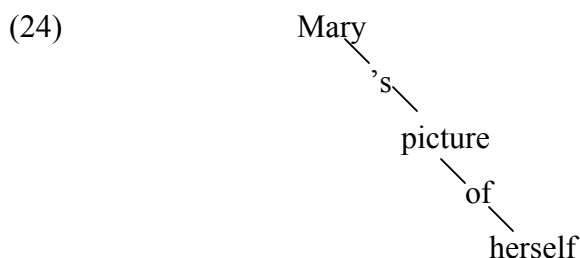
How about *the girl's picture of herself*? Note that *the girl* is not built up as a continuation of the 'derivational cascade' – Nunes & Uriagereka's (2000) term – that built up the rest of the phrase *'s – picture – of – herself*. It was built up in a different derivational space, and merged as a phrase. We can encode this fact by representing it in the larger phrase as follows:

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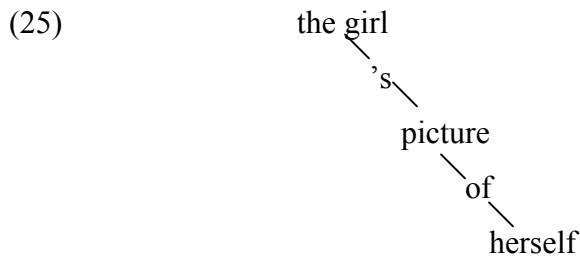
Let us stop to consider (25). It embodies a claim that there can be complex mother nodes, with internal structure. Two questions immediately arise: One, how do we make sense of the notion of a phrasal mother node? Two, how can this structure be accommodated to our declared target of a linear phrase structure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

To answer the first question: In the traditional phrase structure 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 bare phrase structure 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. 14) 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 § 6, 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 projections 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):[16]

(26)

the
girl
's
picture
of
herself
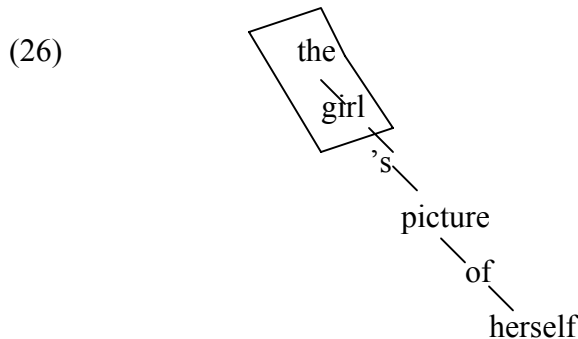
It will be recalled that in the theory of specifier-less syntax, the erstwhile specifier becomes a phrase that – as a whole – takes the phrase it is merged with as its complement; but of course none of its subparts (proper terms) takes the latter phrase as its complement. Thus *the girl* can take the 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) <u>Derivational definition of dominance</u>

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; i.e. 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 a 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):

(15)

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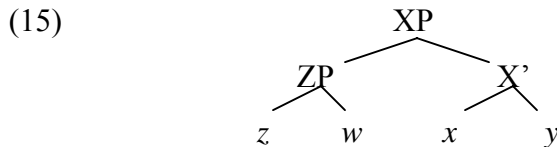
If we adopt the 'first branching node' definition of c-command (Reinhart 1979), X' c-commands ZP, *z*, and *w*. This is an unwanted set of relations; there is no positive evidence of the existence of these relations. For Kayne (1994), these relations also created counterexamples to antisymmetry; for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

(15')

```
                    XP$_1$
            ZP              XP$_2$
        z       w       x       y
```
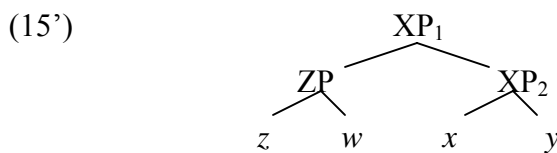
He claimed that a mere segment of a category – in (15'), XP_2 – does not c-command. Chomsky (1995) (see also Epstein 1999) stipulated that an intermediate projection does not c-command, but required that (nevertheless) the intermediate projection has to be present in the tree to prevent – in (15) – *x* 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); and an analysis which claimed that 'specifiers' are in a selectee relation to a following X^0 category. In our analysis, (15) becomes (15''):

(15'')

```
            ZP
                x
                    y
```

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

6. Linearizing the terminal string: a rule of Spell-Out

Note that while our theory yields a partially linear phrase structure tree, we do not yet have a linear ordering of the terminal elements. To see this, consider again (25) or (26). In this structure, we insisted that the terms *the* and *girl* of the merged phrase *the girl*, while they are ordered *inter se* by the relation of dominance, have no dominance relation with respect to the terms of the constituent below the phrase. But linear ordering must be total; i.e., in the present case, for any x, y that are terminal elements, 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 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 grammar (Chomsky 1995). But note that we have also preserved all the major results of antisymmetry (Kayne 1994).

7. Movement in a Linear Tree

How do we do movement in a linear tree?

In a traditional phrase structure 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)
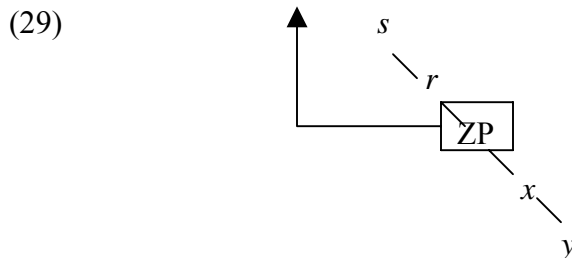
Does the movement of ZP "disconnect" the tree?

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 phrase structure 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 subtree. (As we just said in a previous paragraph, specifier and head 'hang out' from a left branch and therefore are, in that sense, at the bottom of a subtree.) 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, what are the constraints on copying? In this connection, let us adopt an idea of Collins (2002), that a 'saturated' phrase is spelt out.[17] 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 spelt out, it can be copied and merged without any problem.[18]

8. Conclusion: A Note on Phrasal Movement

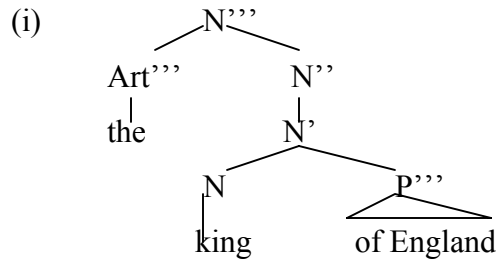We conclude with a note on phrasal movement.

We mentioned Starke's (2004) claim that *wh*-movement in English is in response to a requirement of "f-seq" that there should be a phrase bearing the [+ *wh*] feature above TP in a question (see § 3). Actually, the *wh*-phrase moves in order to fill a Focus position in the C-system (Rizzi 1997); and successive-cyclic *wh*-movement could also be in response to Focus positions in the left peripheries of embedded clauses. It is a parametric property of English that its question operator – which is in fact a disjunction operator in ForceP (Jayaseelan 2001) – is 'paired' with a Focus position (Jayaseelan 2005, 2006); certainly many other languages (e.g. Chinese) do not have this property.

If the subject moves into a "grammaticalized" Topic position above TP (as was suggested in § 1), we can explore the thesis that all 'copy-and-merge' type of phrasal movement (distinguished from the other type of movement noted in fn. 18) is in response to Topic and Focus positions in f-seq. In other words, EPP is a property of Topic and Focus. In the earlier way of representing phrase structure, we would have merged a null Topic or Focus head 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 Topic or Focus feature directly merge, or remerge,[19] with the structure built up by the derivation up to that point, taking the latter as its complement, in a linear tree.

ABNEY, S. 1986. The Noun Phrase in its Sentential Aspect. Doctoral dissertation, MIT.

BAKER, C. L. 1981. Auxiliary-adverb word order. *Linguistic Inquiry* 12(2):309-315.

BAYER, J. 1984. COMP in Bavarian syntax. *The Linguistic Review* 3, 209-274.

BRODY, M. 1997. Mirror theory. Ms., University College London.

CHOMSKY, N. 1970. Remarks on nominalization. In R. Jacobs & P. Rosenbaum (eds.) *Readings in English Transformational Grammar*, 184-221. Massachusetts: Ginn, Waltham. Also in N. Chomsky (1972) *Studies on Semantics in Generative Grammar*, 11-61. The Hague: Mouton.

CHOMSKY, N. 1993. A minimalist program for linguistic theory. In *The View from Building 20*, ed. K. Hale and S. J. Keyser. Cambridge, Mass.: MIT Press.

CHOMSKY, N. 1995. Bare phrase structure. In *Government and Binding Theory and the Minimalist Program*, ed. G. Webelhuth, 383-439. Cambridge, Mass.: Blackwell.

CHOMSKY, N. 2000. Minimalist inquiries. In *Step by step*, ed. by R. Martin, D. Michaels & J. Uriagereka, 89-155. Cambridge, Mass.: MIT Press.

CINQUE, G. 1999. *Adverbs and Functional Heads. A Cross-Linguistic Perspective*. New York: Oxford University Press.

COLLINS, C. 2002. Eliminating labels. In *Derivation and Explanation in the Minimalist Program*, ed. S. D. Epstein and T. D. Seely, 42-64. Oxford: Blackwell.

EPSTEIN, S. D. 1999. Un-principled syntax: the derivation of syntactic relations. In *Working Minimalism*, ed. S. D. Epstein and N. Hornstein, 317-345. Cambride, Mass.: MIT Press.

HUDSON, R. 1990. *English Word Grammar.* Oxford: Blackwell.

JACKENDOFF, R. 1977. *X' Syntax: A Study of Phrase Structure*. Cambridge, Mass.: MIT Press.

JAYASEELAN, K. A. 2001. Questions and question-word incorporating quantifiers in Malayalam. *Syntax* 4(2): 63-93.

JAYASEELAN, K. A. 2005. Remnant movement and word order. Paper presented at the 5th Asian GLOW Colloquium, Jawaharlal Nehru University, New Delhi.

JAYASEELAN, K. A. 2006. Topic, Focus and adverb positions in clause structure. Paper presented at the Cambridge-Hyderabad-Nanzan Joint Seminar on Functional and Lexical Categories, Nanzan University.

KAYNE, R. 2004. *The Antisymmetry of Syntax.* Cambridge, Mass.: MIT Press.

KOOPMAN, H. 1996. The spec head configuration. In *Syntax at Sunset*, ed. E. Garrett and F. Lee, 37-64. (*UCLA Working Papers in Syntax and Semantics* 1.) Department of Linguistics, UCLA. [Also in H. Koopman, *The Syntax of Specifiers and Heads*, 331-365. London: Routledge. 2000.]

KOOPMAN, H. & A. SZABOLCSI. 2000. *Verbal Complexes*. Cambridge, Mass.: MIT Press.

NUNES, J. & J. URIAGEREKA. 2000. Cyclicity and extraction domains. *Syntax* 3(1): 20-43.

POLLOCK, J-Y. 1989. Verb movement, Universal Grammar, and the structure of IP. *Linguistic Inquiry* 20 (3): 365-424.

REINHART, T. 1979. The syntactic domain for semantic rules. In *Formal Semantics and Pragmatics for Natural Languages*, ed. F. Guenther and S. J. Schmidt. Dordrecht: Reidel.

REINHART, T. 1981. A second COMP position. In A. Belletti, L. Brandi, and L. Rizzi (eds.) *Theory of Markedness in Generative Grammar*. Pisa: Scuola Normale Superiore.

RIZZI, L. 1997. The fine structure of the left periphery. In *Elements of grammar. Handbook of generative syntax*, ed. L. Haegeman. Dordrecht: Kluwer.

SEELY, T. D. 2006. Merge, derivational c-command, and subcategorization in a label-free syntax. Ms., Eastern Michigan University.

STARKE, M. 2004. On the inexistence of specifiers and the nature of heads. In *Structures and Beyond*, ed. A. Belletti, 252-268. New York: Oxford University Press.

URIAGEREKA, J. 1999. Multiple spell-out. In *Working Minimalism*, ed. S. D. Epstein and N. Hornstein, 251-282. Cambridge, Mass.: MIT Press.
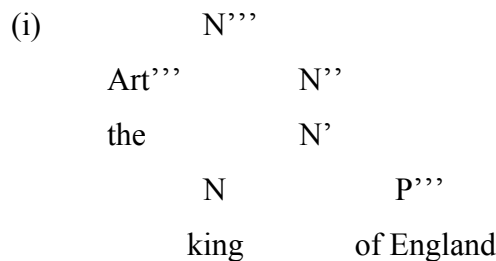
Figure in fn. 3:

(i)
```
              N'''
         ╱         ╲
    Art'''          N''
      │              │
     the            N'
              ╱          ╲
             N            P'''
             │          ╱    ╲
            king      of England
```

---

[1] Cf. Abney (1986). Abney (however) analyzes *'s* as a D^0, and the whole structure as a DP.

[2] 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, figure (51)).

[3] 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)
```
                     N'''
         Art'''              N''
          the                N'
               N                  P'''
             king           of England
```

[4] And significantly, the head in this case is an abstract element. See Starke (2004) for an analysis which argues that there is no v^0.

[5] 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 V-2 phenomenon of Germanic; see Starke (2004) for details.

[6] 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.
[7] See also Seely (2006) for an elaboration of this idea.

[8] 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.

[9] Cf. Chomsky (2000: 133): 'Set-Merge typically has an inherent asymmetry. When α, β merge, it is to satisfy (selectional) requirements of one (the *selector*) but not both.'

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 bare phrase structure, indicated the selector by means of projection and labels. With the elimination of labels, the unequal nature of Merge is unrepresented.

[10] 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).

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

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

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

[13] Can we avoid (19) by appealing to the notion of an 'unsaturated' element (Collins 2002)?

When two phrases are merged – according to the current theory of 'specifier' – one of the phrases must contain an X^0 element with an unsatisfied feature; while the other phrase must not contain such an element or the derivation will crash (see Collins 2002 for arguments). The unsaturated X^0 element can be an immediate constituent of the merging phrase that contains it, but can also be very deeply embedded in that phrase if we are

dealing with multiple specifiers. It is conceivable that we can mark this element in some fashion, say with a feature [–saturated]. But if the copying algorithm 'sees' only sets and members of sets, it will not be sensitive to any such feature and won't be able to pick the right element to copy. A copying algorithm that is made sensitive to the feature composition of the members of sets seems to be a complication compared to which our (19) – which we described as 'overly complex' – is simpler.

[14] Any way of indicating an ordering relation will do, including precedence:

      (i)   eat — apples   (OR   eat^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. Michael Brody's 'Mirror Theory' (Brody 1997). (Brody credits the idea to dependency grammar; see e.g. Hudson 1990.)

[15] 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. 5.)

[16] The function of the 'box' in (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!

[17] See also Uriagereka (1999), Nunes & Uriagereka (2000), for the idea that a moved phrase is spelt out prior to movement; and that a spelt-out phrase is treated like a 'word' by the syntax.

[18] Although not strictly relevant to the thesis of this paper, let me note that there could be a second type of movement that leaves no copy (a 'physical removal' type of movement). The removal of all right-of-V material by 'stacking' in order to facilitate the verb's 'picking up' of inflection by phrasal (VP) movement – first proposed in Koopman & Szabolcsi (2000), and developed in Jayaseelan (2005) – could be a movement of this type.

---

[19] The insertion of *there* (which we can assume has a [+ Topic] feature as a lexical property) would be a case of simple merge (i.e. external merge) in a Topic position.