

Labels and Structures*

David Adger
d.j.adger@qmul.ac.uk

APRIL 2011
DRAFT OF A CHAPTER OF A BOOK

1 A Problem and a Proposal

In this chapter, I motivate a new approach to the derivation of structure and its labeling. I do this by removing a stipulation from the definition of Merge, and by replacing functional categories qua lexical items with a function that directly labels structures. The removal of the stipulation in Merge allows unary branching structures which, in the new system, provide the structural carapace for sequences of (mainly functional) categories; the labeling function recurses from the categories assigned to roots, assigning labels to ever larger structures on the basis of a universally given sequence of categories (Cinque 1999, Starke 2001, Adger 2003), obviating the need for functional heads. Unary branching labeled structures provide the locus for morphological realization so that what were previously considered to be functional lexical items are, in the new system, just spellouts of pieces of labeled structure. Standard binary Merge then provides the further structure necessary for negotiating the syntactic and semantic connections between sequences of categories.

Part of the motivation for setting up the system in this way is that it provides us with a solution to a serious problem in the derivation of current phrase structural representations: how structures containing base generated specifiers are labeled.

The architecture of the new system has two repositories of information relevant to the syntax. One is a lexicon of roots (RLex); these have only conceptual/encyclopaedic information (cf. Marantz 1997, Borer 2005b). The other is a set of categories (CLex), on which is imposed some organization, such that subsets of CLex can be mapped to monotonically increasing sequences (that is, we have a linear ordering imposed on some subsets of CLex). These sequences correspond to the Extended Projections of Grimshaw (1991). Merge operates only on elements of RLex and Merge's own outputs. A function Label associates an element of CLex with a structure built by Merge.

This gives a certain two-dimensionality to derivations, which build pairs of structures and associated labels.

*Many thanks to Klaus Abels, Daniel Harbour, Luisa Martí, Linnaea Stockall, Peter Svenonius, for comments on the ideas presented here and also to the participants of Richie Kayne's NYU syntax seminar series April 2011 where this was first presented.

We therefore have:

- (1)
 - a. $R_{Lex} = \{\sqrt{\text{cat}}, \sqrt{\text{jump}}, \dots\}$
 - b. $C_{Lex} = \{N, Cl, Num, D, V, v, T, \dots\}$
- (2)
 - a. Merge applies to elements of R_{Lex} giving Syntactic Objects (SOs) as its outputs, and further applies to those syntactic objects building new SOs.
 - b. Label applies to SOs, associating them with an element of C_{Lex} .

This architecture allows us to address and solve the problem of how structures involving Merge of specifiers are labeled, an unresolved issue for the standard theory developed in Chomsky (2004) et seq.

The theories of Merge and labeling I develop have some interesting corollaries: (i) elements of R_{Lex} cannot have structural complements; (ii) if R_{Lex} and C_{Lex} are not stipulated to be disjoint, then some structures will effectively involve ‘dual’ extended projections; (iii) a certain class of roll-up and remnant movement derivations are ruled out, while a class of structurally similar, but not identical, base generated configurations is allowed into the system. I’ll suggest that these properties of the system lead us to a potentially interesting new perspective on how complements are introduced into structure.

1.1 The specifier problem

The standard view of how syntactic structures are built up in Minimalist theorizing is that lexical items are subject to Merge, defined as follows, where X , Y and $\{X, Y\}$ are all syntactic objects:

- (3) “...we take $\text{Merge}(X, Y) = \{X, Y\}$...” Chomsky (2007, p8)

More explicitly (from Collins and Stabler 2009):

- (4) **Definition 11.** Let W be a workspace and let A, B be syntactic objects where $A, B \in W$ and A and B are distinct ($A \neq B$). Then, $\text{External-Merge}_W(A, B) = \{A, B\}$.

Since no restriction is placed on the provenance of the inputs to Merge, this definition also yields a movement configuration, generated in the same basic way, but with A containing B (or vice versa). This gives the by now familiar distinction between External Merge (EM) and Internal Merge (IM).

The output of Merge then enters into further syntactic operations. Chomsky suggests that the output therefore must have some properties so that these further operations can apply to it:

Each SO generated enters into further computations. Some information about the SO is relevant to these computations. In the best case, a single designated element should contain all the relevant information: the label (the item “projected” in X' -theories; the locus in the label-free system of Collins 2002). The label selects and is selected in EM, and is the probe that seeks a goal for operations internal to the SO: Agree or IM. Chomsky (2008, 141)

Assuming this to be the case, a means of determining the label is necessary. Chomsky suggests (Chomsky 2008, 145):

- (5) a. In $\{H, \alpha\}$, H an LI [LI=Lexical Item, DJA], H is the label
 b. If α is internally merged to β , forming $\{\alpha, \beta\}$ then the label of β is the label of $\{\alpha, \beta\}$

That is, the label is predictable from the internal configuration of the output of Merge: in the head-complement case of Merge at least, the label is the element whose properties are available with minimal search. It follows that in this case the label is H , which is a lexical item. With suitable statements about the timing of operations, it may be possible to extend this intuition to the movement case in (b) above (e.g. Minimal search identifies the LI β in $\{\beta, \gamma\}$, this then probes α in γ and is therefore identified as the label of $\{\alpha, \beta\}$, as it is still ‘active’ until α is raised and Merged). However, Chomsky does not explicitly unify the two parts of the labeling algorithm, and doing so is not trivial. As matters stand, we have a non-uniform labeling algorithm.

However, as noted by Chomsky (2008:145), even this non-unified algorithm is insufficient. There are 3 problematic cases.

The first is the initial step of the derivation (in fact of most extended projections in the derivation) where two LIs Merge. Since both inputs to Merge are LIs, the algorithm determines neither as the label uniquely. Chomsky suggests that in this case either may be the label and that if the wrong choice is made, the resulting structure will be quickly filtered out by the interface systems.

This approach actually takes labeling to be non-deterministic in contrast to the thrust of the labeling algorithm given in (5) (cf Citko 2008): the label may be drawn from either constituent of the relevant SO, and derivations with incorrect choices are filtered out (contra the ‘crash-proof’ approach of Frampton and Gutmann 2002).

Another problematic case is when an LI (α in (6)) is Internally Merged to some higher projection:

- (6) $\{ \alpha \{ \beta, \{ \gamma, \alpha \} \} \}$

In this case, the two subclauses of the labeling algorithm conflict, with (5-a) making the LI α the label, while (5-b) makes β the label. Chomsky, taking labeling to be non-deterministic in this case, and following Donati (2006), suggests that in such a situation both outcomes are in fact possible. For example, in a case where some wh-word has been raised to the specifier of CP, either the wh-determiner itself labels the resulting structure as a DP, or the attracting probe C labels it as a CP. This is then taken to be what happens in an English style free relative construction:

- (7) $[_{CP/DP} \text{What}[D] [_{\bar{C}} C [_{TP} \text{you wrote } \langle \text{what} \rangle]]$

This particular analysis is problematic, as it is unclear that *what* is itself not syntactically complex. In other languages free relative constructions have an entirely different syntax and morphology from indirect wh-questions, casting doubt upon their unification in English. Compare the indirect question and free relative variants in Scottish Gaelic, for example:

- (8) a. Dh'fhaighnich mi dè a sgrìobh thu.
ask.PAST I what that write.PAST you
'I asked what you wrote.'
- b. Leugh mi na sgrìobh thu.
read.PAST F-REL write.PAST you.
'I read what you wrote.'

The indirect question is formed by a fronted wh-expression *dè*, which shares distribution with any other wh-expression (that is, instead of *dè*, 'what', we can also have *cò*, 'who', *cuine*, 'when' etc). The free relative can only be formed with the particle *na*, which is plausibly a contraction of the definite article *an* and the relative particle *a*. Further evidence for this analysis is that in Gaelic certain prepositions carry a special inflection when they appear with a following definite article (although not with their definite expressions, such as, for example, proper names):

- (9) a. ris na caileagan
to.DEF the.PL girls
'with/to the girls'
- b. *ri na caileagan
to the.PL girls
'with/to the girls'
- c. ri Màiri
to Màiri
'with/to Màiri'
- d. *ris Màiri
to.DEF Màiri
'with/to Màiri'

We can take this inflection as diagnostic, then, of the presence of definite article (rather than of syntactic definiteness generally). This inflection appears obligatorily with free relatives:

- (10) Èisd mi ris na sgrìobh thu.
listen.PAST I to.DEF F-REL write.PAST you.
'I listened to what you wrote.'

However, in constructions where the wh-word *dè*, 'what' is in situ (e.g. echo-questions), it does not trigger the definiteness inflection on a P:

- (11) Èisd thu ri(*s) dè?
listen.sc past you to.(*DEF) what
'You listened to what?'

This suggests that free relatives in Gaelic involve an article (D) taking a relative clause complement. For example, following Adger and Ramchand (2005)'s proposal that Gaelic relatives simply involve the direct binding of a variable from C, we would have:

- (12) [D na [C[REL] you wrote pro]]

A similar analysis might be extended to the English free relative case, reducing the force of Donati's argument and Chomsky's appeal to it to explain this problem for the labeling algorithm.

The final case is however the most problematic. I quote Chomsky here:

The exceptions are EM of non-heads XP, YP, forming {XP, YP}, as in external argument merger of DP to v^*P . The conventional assumption is that the label is v^* . A possibility is that either label projects, but only v^* -labeling will yield a coherent argument structure at C-I. Another possible case is small clauses, if they are headless. A suggestive approach, along the general lines of Moro (2000), is that these structures lack a label and have an inherent instability, so that one of the two members of the small clause must raise. Chomsky (2008, 160, note 34)

This is what I term the Specifier Problem:

(13) In a configuration {XP, YP}, how is the label determined?

Since neither of XP or YP is an LI, Chomsky's algorithm does not apply. Moreover, no obvious considerations based on simplicity of search seem to pertain.

For external Merge of a specifier, Chomsky suggests two possible solutions. One is based on Moro's (2000)'s idea that {XP, YP} structures are somehow too symmetric, and this symmetry has to be disrupted by movement of one or other of XP and YP. Applying this to Merge of the specifier of v^*P , we could say that the specifier has to raise, leaving a structure with just a head (LI), v^* , which provides the label (assuming that the trace can be ignored).

However, this would mean that all base Merged specifiers have to raise, since they will all give rise to the same problem. But the question that is then raised is whether there is always a target for such raising. Take, for example, small clause absolutes in English:

(14) With the vase on the table, the room looks perfect.

There is no evidence that *the vase* has moved. In fact, the lack of expletives in such structures suggests that there is no target position:

(15) *With there a vase on the table, the room looks perfect.

Furthermore, taking Moro's position, we might expect predicate inversion in such constructions, which is also impossible:

(16) *With on the table the vase, ...

Similar considerations apply to complements of small clause taking predicates like *consider* and possibly also to causative *make* and perception *see*. Connected to this empirical problem is the theoretical one of how to ensure that there is always an attractor for the subject that causes it to move.

The other idea Chomsky considers is the same strategy as for external Merge of two LIs: either can be the label but the interface will filter out the incorrect labeling via appeal to the argumental properties of the embedded predicate. This will be the case for initial Merge

of the specifier of v^* , or, more generally, for any subject introducing functional head such as PredP (Bowers 1993), or the head that introduces possessors (Radford 2000) etc.

One problem with this proposal is its inconsistency with the general algorithm. Why not just allow either label to project in general, with the interface filtering out the problematic cases? The answer is that in most other instances it is not possible to appeal to general considerations of coherent argument structure (for example, in Merge of T and Asp no obvious argument structure considerations arise which will determine which of the two projects).¹ Moreover, appeal to C-I properties for this structure, but to syntactic labeling algorithms for the others, seems decidedly unminimalist: either the interface conditions apply generally across the various subcases, or the syntactic system determines the label via some formal property deriving from the functioning of Merge.

Furthermore, Chomsky's appeal to 'coherent argument structure' will not apply to all cases of the Specifier Problem. For possessors in particular, it seems unlikely that the interface will simply filter out the wrong answer by appeal to properties of argument structure. Take DPs like *Anson's picture of Lilly* or *Anson's side of the table*. The argument structure of *picture/side* is irrelevant to the interpretation of the possessor. On initial Merge of *Anson* with whatever structure is built above *picture of Lilly/side of the table* that allows the possessive interpretation, we need to ensure that the label is that of *picture/side*, and not that of *Anson*. This is for two reasons (i) the label is a signal to the conceptual intensional interface as to what the phrase is to mean and projecting the wrong label will give a meaning something like 'Anson who is relevantly related to the picture of Lilly'; (ii) languages treat the two projections differently (for example, Hungarian requires a special possessive morpheme on the noun *picture* when it is possessed, so something must identify it as the syntactic 'possessee').

In fact, we can strengthen the Specifier problem and ask:

- (17) Is there a unified labeling algorithm that applies in the same way to all syntactic configurations?

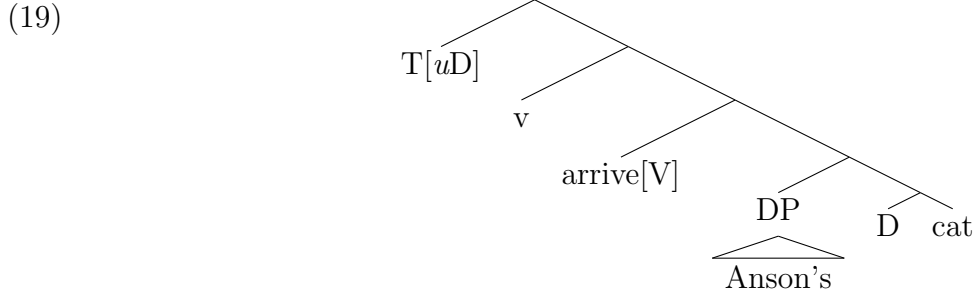
One response to this question is that of Collins (2002): structures are not labeled. The various syntactic relations that elements enter into are asymmetric enough to provide information about which of the two subconstituents of a syntactic object is the head. That information will serve the purposes of labels in a labeled system. As Collins notes, this requires the syntactic system to be sensitive to all sorts of syntactic relations (syntactic selection, agreement, theta-role assignment, EPP etc), with the asymmetry of each relation effectively providing the information about which subconstituent of an SO is taken to be the label.

There have been a number of criticisms of this label-free system (Seely 2006, Hendrick 2007). However, I think the most compelling reason not to adopt this approach is how it interacts with movement theory. Take, for example, a derivation where an unaccusative verb combines with a DP containing a specifier:

¹One could specify that one of T or Asp is the semantic functor, that semantic functors correspond with syntactic argument taking status, and therefore the projecting head is whatever the semantic functor is (see below), but this is not an appeal to argument structure at CI in the same sense, and effectively amounts to just stating which is to be taken as the head.

(18) Anson's cat arrived.

This derivation includes, in a label-free system, a structure of the following form, at the point before EPP driven movement applies:



Now the EPP feature on D attracts the closest DP. But there is no information on the label of the complement of V to ensure that that counts as the closest DP, and we predict generalized possessor raising. There are solutions to this problem, including developing particular theories of pied piping, but these all effectively restate standard labeling.

Adger (2003) suggested an alternative but related solution to these problems that relies on the idea that operations such as Merge are triggered (or at least swiftly checked by the syntactic system). In that system, selectional features are uninterpretable features which have to be checked by a matching feature on the selectee (cf. Chomsky 2000, 133). Merge of a head with its complement will invariably require the head to bear such a feature (irrespective of whether the complement is just an LI or a phrase). The proposal takes the \bar{X} -projection of an LI to be identical to that LI, which allows, for example, v^* to bear a selectional feature (uD) which projects to \bar{v}^* and which can then be checked by Merge of a DP subject under sisterhood. In such a system, the slogan *what selects projects* determines labels. Boeckx (2008) proposes a similar system, but takes the labels to be given by the element that probes for ϕ -features. More recently, Cecchetto and Donati (2010), following Adger (2003) and Boeckx (2008), have suggested that it is always the probe in any syntactic operation that labels the output of that operation.

All of these systems attempt to argue that there is in fact a unified labeling algorithm (for Collins, the unification of labeling is to be achieved by eliminating labeling) and they all adopt the intuition that it is an internal property of a lexical item, inherited by that LI's projections, that determines its capacity to have complements and specifiers. However, none of them solve the essential problem for specifiers: in a configuration $\{XP, YP\}$, how is the label determined? What these proposals do instead is say: in a configuration $\{XP, YP\}$, inspect the heads of XP and YP to see whether they have a property that will determine the label of the whole configuration.

For example, imagine the derivation has reached a point where Y, which bears a selectional feature for Z and a selectional feature for X, has Merged with Z:



In the Adger/Boeckx/Cecchetto/Donati approach, Y labels the new SO. At the next stage of the derivation, XP is Merged and the label for the new structure is to be calculated. For

Boeckx, and Cecchetto and Donati, the asymmetry of selection (that is, the fact that Y still has an X feature to be satisfied) is enough to label the resulting object as Y. For Adger, the whole complex $Y[Z, X]$ labels the mother node in (20), and then the X-feature is satisfied under sisterhood, so that the new object is a YP. In either case, the XP must be first Merged before the Y-X relationship is determined. But now the basic problem re-emerges. We must look at internal properties of both XP and YP to determine the label, but this means that the label of α is not determined by properties of the elements that α immediately contains.²

Furthermore, if a single LI can have both a complement and a specifier, and these have different syntactic properties, the system requires further stipulations to order them. For example, if a ditransitive verb takes both an DP object and a PP as internal arguments, how is this ‘base’ order effected so that one is the complement and the other the specifier? Of course there are solutions to these problems (e.g. allow only one ‘selectional’ feature per head) but these amount to no more than further stipulations.

We are left, then, with two problems:

- (21) a. The Specifier Problem: in $\{\alpha, \beta\}$, where neither α nor β are lexical items, how is the label to be determined?
b. The Labeling Problem: Is there a unified labeling algorithm that will suffice for all cases, and if so, what is it?

1.2 Diagnosis: the problem is heads, not labels

The approach to this problem I will explore takes these problems to emerge because of the LI-drivenness of the system. In Adger 2010b, for rather different reasons, I suggested that labeling should be exocentric, rather than endocentric (see also Boeckx 2010). In the remainder of this chapter, I’ll develop this idea more fully, and show how it provides a solution to the Specifier and Labeling Problems.

Merge is usually understood to be a binary operation. As discussed above, given a binary operation the logical possibility exists that one operand may be part of the other, allowing us to distinguish between binary and singulary transformations (see Chomsky 1961) and resulting in a system with both external and internal merge.

Consider again the formalization of this idea given by Collins and Stabler (2009):

- (22) **Definition 11.** Let W be a workspace and let A, B be syntactic objects where $A, B \in W$ and **A and B are distinct ($A \neq B$)**. Then, $\text{External-Merge}_W(A, B) = \{A, B\}$.

This formalization makes clear that there is a further logical possibility, currently ruled out by the clause in bold: the operands may be identical. If the operands are identical, the output of the Merge operation is a singleton set (see also Guimaraes 2000, Kayne 2010).³

²Cecchetto and Donati (to appear) propose something slightly different. They suggest that in selection “an element in the numeration can probe an element in the computation and trigger external merge”. Aside from the fact that the probe-goal relation is then non-configurational, this still does not solve the problem at hand, since both X and Y are clearly parts of syntactic objects.

³Kayne (2010, 332) actually takes the singleton set $\{x\}$ not to arise from Merge of x with x , as I will do here, but rather via ‘direct formation’, since $\text{Merge}(x, x)$ raises questions about occurrences of x . However, if we take Merge to apply to operations in a workspace (as in the Collins/Stabler approach), multiple

Following these suggestions, the first theoretical proposal I'd like to make is:

- (23) Remove the distinctness condition on Merge

We can schematize the three ensuing possibilities as:

- (24) a. Merge(A, B), A distinct from B, $\rightarrow \{A, B\}$ (External Merge)
 b. Merge(A, B), A part of B, $\rightarrow \{A, B/A\}$ (where B/A signifies A is contained in B) (Internal Merge)
 c. Merge(A, B), A = B, $\rightarrow \{A, A\} = \{A\}$ (Self-Merge)

While external and internal Merge give rise to a syntactic object with a cardinality of 2 (that is, it has a binary structure), Self-Merge gives rise to a syntactic object with a cardinality of 1 (that is, it is a unary structure).⁴

Schematically, one kind of derivation, utilizing only self Merge, will look as follows:

- (25) a. Merge x with x = $\{x, x\} = \{x\}$
 b. Merge $\{x\}$ with $\{x\} = \{\{x\}, \{x\}\} = \{\{x\}\}$
 c. Merge $\{\{x\}\}$ with $\{\{x\}\} = \{\{\{x\}\}, \{\{x\}\}\} = \{\{\{x\}\}\}$
 d. ...

Mixing Self Merge with External Merge will give a derivation of the following general shape:

- (26) a. Merge x with x = $\{x, x\} = \{x\}$
 b. Merge $\{x\}$ with $\{x\} = \{\{x\}, \{x\}\} = \{\{x\}\}$
 c. Merge y with y = $\{y, y\} = \{y\}$
 d. Merge $\{\{x\}\}$ with $\{y\} = \{\{\{x\}\}, \{y\}\}$
 e. ...

The immediate issue to address now is that of the label of these various constituents. Taking the Self Merge derivation first, it appears that we'd expect no labeling to be possible, since no head is Merged (that is, the only LI is x). If it is heads that provide labels, and all structure needs to be labeled, then we could rule out unary branching structures (cf. Kayne 2010).

However, I want to pursue here the idea that the effect of iterated Self Merge is to create an extended projection of the initial root category in the absence of any further merger of heads. I am going to adopt a methodology that simply assumes that work in the cartographic approach to syntactic structure (e.g. Cinque 1999, Rizzi 1997) is along the right lines, and I will further assume that there is a solution to the problem of what gives rise to the cartographic ordering and this solution is not based on one functional head syntactically or semantically selecting the next (see, especially, Starke 2001 and Adger 2003, who take the

occurrences of x pose no special problems.

⁴One might ask whether this is not too 'literal' an interpretation of Merge as a set forming operation. Perhaps Merge actually forms ordered sequences, or multisets. However, these are different theories, with different foundations, and presumably with different consequences. The only way to test such deep distinctions between theories is to work out the consequences of the assumptions that are made. This is what I do here for the claim that Merge forms sets.

extended projection of a root to be given by an interface constraint on Merge, and Williams 2003 who applies this same methodology in theory development). Starke states this as:

- (27) there exists an ‘fseq’ - a sequence of functional projections - such that the output of [Merge] must respect fseq. (Starke 2001, p155).

Williams takes ‘the existence of the functional sequence and its linear structure as axiomatic’ (Williams 2003, 175) and leaves open the ‘mystery’ of the difference between functional embedding (that is the hierarchical ordering of functional categories) and what he calls complement embedding (that is the capacity of a verb or other lexical category to take a whole new functional hierarchy as a complement). Adger, following ideas stemming from Abney (1987) and Grimshaw (1991) defines a Hierarchy of Projections taking, for example, vP to be ‘an extension of the projection of VP, in that it is still verbal, but it adds further semantic information’ (Adger 2003, 135). For Adger (2003), Merge requires either satisfaction of a selectional relationship via feature checking, or satisfaction of the Hierarchy of Projections (the acuteness of the ‘mystery’ raised by complement vs functional embedding becomes especially clear in the partial formalization of the system given by Adger (2010a), where two different definitions of Merge have to be developed—a problem solved here by actually simplifying the definition of Merge).

Adopting this method of theory construction, let us take the extended projection of any root to be given axiomatically, as far as the syntax is concerned. It is simply a property of UG (hopefully to be derived in some fashion, see, for example, Nilsen 2003). This leads to the second major theoretical proposal:

- (28) There are no functional categories qua lexical items

In any particular act of syntactic combination, the label can be given directly, and locally, on the basis of antecedently assigned labeling and the axiomatic functional sequence. The only lexical item necessary is the root of the extended projection, and this provides the initial label. I will take the core lexical category labels to be N, V, and A (following Baker 2003) and assume that these categories label the output of Self Merge of lexical roots (taking roots themselves to be to be labelless (e.g. Marantz 2006, Borer 2005a)).⁵

Rather than a single lexicon consisting of both ‘functional’ and ‘lexical’ LIs, we have:

- (29) a. RLex = $\{\sqrt{1}, \dots, \sqrt{n}\}$, the set of LIs (roots)
b. CLex = $\{l_1, \dots, l_n\}$, the set of category labels

In this system, elements of RLex are in the domain of Merge, as are outputs of Merge. Structure is built from RLex plus Merge. On the assumption (to be revised below) that CLex is disjoint from RLex, elements of CLex are simply labels for the structures built by Merge.

Making explicit the assumption defended above that the extended projections given by UG (however derived) can be treated as axiomatic, I’ll define such extended projections as

⁵These are ‘add on’ assumptions. For example, the system would equally be compatible with the proposal that only N begins an extended projection, with Vs and As being interpretations of higher ‘layers’ in the extended projection of N, as in Kayne (2010).

follows:

- (30) A Universal Extended Projection of a category C (UEP_C) is a sequence drawn of labels drawn from C_{Lex} ($l_s, \dots l_t$), where l_s is the Start Label and l_t is the Terminal Label.

I assume three of these, started by N , V , A (Baker 2003), so we have EP_N , EP_V , EP_A .

We then define the binary Cartesian product of C_{Lex} as a set of Label Transition Functions (LTFs):

- (31) $\Lambda = C_{Lex} \times C_{Lex} = \{ \langle N, Cl \rangle, \langle N, N \rangle, \langle Cl, N \rangle, \langle Cl, Cl \rangle, \langle N, Num \rangle \dots \}$

Λ itself is subject to no constraints; it allows mappings from any category to any other. It is therefore extremely liberal in what it allows. However, for any particular (I)-language, some subset of Λ will exist. Part of the acquisition process is determining what the content of Λ is. Evidence for this in particular languages will be found in the morphology and in the distributional patterns found in the primary linguistic data.

It is plausible to assume that during acquisition of a particular language's Λ certain LTFs are universally ruled out as they do not 'track' the properties of the relevant EP. For example, we might impose the following conditions on elements of Λ universally.

- (32) a. Lower Boundedness of EPs: For any $f \in \Lambda$, l_s (i.e. N , V and A) can be in the domain but not the range of f .
b. Upwards Monotonicity of EPs: if $\langle i, j \rangle \in \Lambda$, where $i \neq l_t$, and $i, j \in EP_C$, then $j \geq i$ in EP_C .

Lower Boundedness ensures that no LTF can map from one category in an EP to the Start Label of that EP, while Upwards Monotonicity ensures that transitions between one category inside an EP and the next will go up, and not down (since the Terminal Label of an EP is the highest category in it, obviously any LTF will have to carry it to a new EP, hence its exclusion from Monotonicity). There may well be other constraints on what can go in Λ (perhaps blocking LTFs that transit from one EP to a new one without including a Terminal Label, for example), but I leave these aside for the moment. The idea is that during development, a child acquiring a language will successively manipulate the LTFs in Λ , subject to whatever universal constraints apply, so that the fully developed language has only a subset of the possible LTFs. In addition, I assume that the labels in the LTFs may have further idiosyncratic properties which will impact on the morphosyntax of the learned language (these correspond to the second-order interface features of Adger and Svenonius 2011). For example, properties that identify a piece of labeled structure as the locus for spellout, or as requiring agreement with a specifier.

We now define a unified labeling function as follows:

- (33) a. If $\alpha, \beta \in \gamma$, then $Label(\gamma) = \text{some } L \in C_{Lex}$, such that there are (possibly non-distinct) f and $g \in \Lambda$ such that $f(Label(\alpha)) = g(Label(\beta)) = L$. (**Transition Labeling**)
b. $Label(\{\sqrt{x}\}) = \text{some } L \in \{N, V, A\}$ (**Root Labeling**)

What (33) does is the following: it says that the label of an SO built by Merge is dependent on (but not identical to) the label of both of its subconstituents. Rather than drawing a functional category from the lexicon and Merging it with some SO, and hence labeling the result, the system capitalizes on the idea that the order of functional categories must be given anyway. So rather than having a functional lexicon, we simply use the antecedently given order of functional categories as the source of labeling information. The label of some syntactic object is L if there is a transition from the labels of that object's subconstituents to L. I will (rather laxly) use a function or an ordered pair notation for LTFs, depending on what makes best expositional sense.

For unary branching structures, the system capitalizes on the fact that $\{A\} = \{A, A\}$, so a label for $\{A\}$ can be calculated by seeing if there are LTFs $\in \Lambda$ that will take us from the label of A to another label, which will be the label of $\{A\}$. Since f and g can be non-distinct, all we need is that there is some function f that will take us from the label of A to that of $\{A\}$.

Assume that Λ for English is partially specified as follows:

$$(34) \quad \Lambda = \{ \langle N, Cl \rangle, \langle Cl, Num \rangle, \langle Num, D \rangle, \dots \}$$

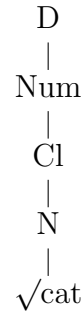
We have, then, the following kind of derivation:

- (35) a. Merge $\sqrt{\text{cat}}$ with $\sqrt{\text{cat}} = \{\sqrt{\text{cat}}, \sqrt{\text{cat}}\} = \{\sqrt{\text{cat}}\}$
- b. Label($\{\sqrt{\text{cat}}\}$) = N by Root Labeling
- c. Merge $\{\sqrt{\text{cat}}\}$ with $\{\sqrt{\text{cat}}\} = \{\{\sqrt{\text{cat}}\}, \{\sqrt{\text{cat}}\}\} = \{\{\sqrt{\text{cat}}\}\}$
- d. Label($\{\{\sqrt{\text{cat}}\}\}$) = Cl since there are f and $g \in \Lambda$ such that $f(N) = g(N) = Cl$ (f and g non-distinct = $\langle N, Cl \rangle$)
- e. Merge $\{\{\sqrt{\text{cat}}\}\}$ with $\{\{\sqrt{\text{cat}}\}\} = \{\{\{\sqrt{\text{cat}}\}\}, \{\{\sqrt{\text{cat}}\}\}\} = \{\{\{\sqrt{\text{cat}}\}\}\}$
- f. Label($\{\{\{\sqrt{\text{cat}}\}\}\}$) = Num since there are f and $g \in \Lambda$ such that $f(Cl) = g(Cl) = Num$
- g. Merge $\{\{\{\sqrt{\text{cat}}\}\}\}$ with $\{\{\{\sqrt{\text{cat}}\}\}\} = \{\{\{\{\sqrt{\text{cat}}\}\}\}, \{\{\{\sqrt{\text{cat}}\}\}\}\} = \{\{\{\{\sqrt{\text{cat}}\}\}\}\}$
- h. Label($\{\{\{\{\sqrt{\text{cat}}\}\}\}\}$) = D since there are f and $g \in \Lambda$ such that $f(Num) = g(Num) = D$

The function 'Label' takes an unlabeled syntactic object as its argument and provides it with a label. This function involves minimal search: it inspects the unlabeled object to see what it immediately contains and uses that information to provide the new label. The new label, however, is not identical to the label of what is contained in the object, rather it is calculated from that label, conforming with the language particular instantiation of the universally given extended projection of the root category.

We can represent this as a tree structure which contains a sequence of labels erected above a lexical category:

(36)



This is a ‘telescoped’ representation in the sense of Brody (2000a), although built by different means.

(36) instantiates the UEP of N as a syntactic object which is the result of a particular derivation in a particular language. This object has a hierarchical structure such that each element contains a further syntactic object (until we hit the start category N which contains only the root $\sqrt{\text{cat}}$). These are the ‘terms’ of Chomsky (1995). Further, each term is associated with a label. Since all of the labels in (36) are drawn from UEP_N , and the labels are organized into a sequence comports with UEP_N by virtue of the labeling function, (36) is an Instantiated Extended Projection of N (so a UEP is just a sequence of labels, while an Instantiated EP is a syntactic object with a particular organization of labels associated with its terms). When we come to define syntactic relations the notion of Instantiated EP will become important.

To see how the whole system works for the Specifier Problem, we’ll look at the most recalcitrant situation: specifier of v^* . For concreteness, take, for example, v^* as it appears in an unergative structure with some verb (say $\sqrt{\text{jump}}$). In order to create a structure where v^* has both a complement and a specifier, there simply have to be, in Λ two LTFs as follows:

$$(37) \quad \Lambda = \{ \dots, \langle V, v^* \rangle, \langle D, v^* \rangle, \dots \}$$

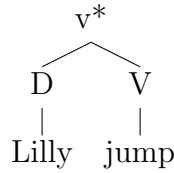
$\langle D, v^* \rangle$ maps from labels in one EP to another, while $\langle V, v^* \rangle$ follows the extended projection of V .

It follows from these, and the unified definition of binary labels that in a binary Merge structure, v^* can label a structure that immediately contains both V and D . More concretely, a derivation of *Lilly jumps* (assuming, probably counterfactually, that *Lilly* is a single lexical item) looks as follows:

- $$\begin{aligned}
(38) \quad & \text{a. Self Merge } \sqrt{\text{jump}} = \{ \sqrt{\text{jump}} \} \\
& \text{b. Label}(\{ \sqrt{\text{jump}} \}) = V \\
& \text{c. Self Merge } \sqrt{\text{Lilly}} = \{ \sqrt{\text{Lilly}} \} \\
& \text{d. Label}(\{ \sqrt{\text{Lilly}} \}) = D \\
& \text{e. Merge } \{ \sqrt{\text{jump}} \} \text{ and } \{ \sqrt{\text{Lilly}} \} = \{ \{ \sqrt{\text{jump}} \}, \{ \sqrt{\text{Lilly}} \} \} \\
& \text{f. Label}(\{ \{ \sqrt{\text{jump}} \}, \{ \sqrt{\text{Lilly}} \} \}) = v^* \text{ since there are } f \text{ and } g \in \Lambda \text{ such that } f(V) \\
& \quad = g(D) = v^*
\end{aligned}$$

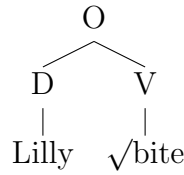
As a tree structure, this looks as follows:

(39)



The same kind of derivation will also be needed for objects of transitives. These also require binary Merge. In this system, objects are introduced via a piece of functional structure, rather than being directly Merged with a root, since it is Self Merge of a root that provides the ‘start category’ for the relevant extended projection (given Lower Boundedness of EPs—see below for further discussion). Following Adger, Harbour, and Watkins (2009), I’ll simply call this functional structure O. An object will then be licensed as there is an LTF that maps from V to O, and similarly one that maps from D to O.

(40)



Unaccusative and zero-place predicates involve projection of v rather than v^* above O or V respectively (with potentially richer structures required, as in Ramchand 2008).

The two structures presented immediately above look very similar, qua structures, but it is crucial that what is interpreted is the labeled structure, so that it is the presence of O vs v^* that signals the correct interpretation of the single argument in each case. There are, in this system, no true complements of lexical roots, only specifiers of labeled structures built above such roots.

More generally, specifiers are simply the case of Merge(x , y) where $x \neq y$, while complement structures are built when $x = y$. Labels are given in the same way for both: via a labeling function that maps from the information already present in the derivation using (the language particular instantiation of) a universal extended projection sequence.

With this in hand, the Specifier Problem and the Labeling Problem both melt away. Recall that we had:

- (41)
- a. The Specifier Problem: in $\{\alpha, \beta\}$, where neither α nor β are lexical items, how is the label to be determined?
 - b. The Labeling Problem: Is there a unified labeling algorithm that will suffice for all cases, and if so, what is it?

In a system where labels are determined by the LI status of α or β , and where neither is an LI, there is no obvious answer to the Specifier Problem. In a system where the label is determined by whichever of α or β probes, we need to inspect further the properties of the constituents of α and β , or we need to stipulate that the probing capacity is somehow able to project upwards. Even accepting this, some statement of the relationship between probing/selecting etc and the labelhood/locushood of α and/or β needs to be made. The problem arises because of the assumption that it is properties of heads that are relevant.

Under the alternative system I have sketched above, the Specifier problem just does not

arise. We jettison heads and adopt LTFs in their place. These map from one category to another. The higher label is dependent on the lower, but it is not identical to it. There is a single general algorithm for the use of these functions in labeling which applies uniformly to unary and binary structures, providing a positive answer and concrete proposal for the Labeling Problem.

Importantly, any system will need to have some statement of the ordering of the various functional categories. The standard system has heads, and the ordering of functional heads is either given axiomatically (as in Starke 2001, Adger 2003) or by selection. However, any system that gives the order via selection needs to either allow disjunctive selectional requirements (to allow for optional functional heads intervening between selector and selectee) or to assume that there are no optional ‘projections’. In the latter case, statement of the selectional properties of the functional heads is once again simply axiomatic. Given this, the standard system needs to state the ‘transition’ from one functional head to the next, whether via an independent constraint, or via selection. So the standard system has what Williams calls ‘functional embedding’, however executed, plus the Specifier/Unified Labeling Problems. The system I have proposed here as an alternative has the functional embedding problem, but lacks the others.

A major reason that labeling is generally taken to be endocentric is Inclusiveness (Chomsky 1995). However, whether we label structure in the way I have proposed or we label structure by drawing heads from a functional lexicon is actually immaterial. Inclusiveness is trivially satisfied if we build all syntactic structure into lexical items and allow it to project via simple syntactic processes, so it is not the ‘lexical’ nature of the source of information that is crucial to inclusiveness: inclusiveness effectively bars the introduction of descriptive technology during the course of a derivation (cf Chomsky 2008) minimizing the addition of information to the derivation. But adding information via Merge of functional heads, or adding information via LTFs are effectively equivalent in these terms (in fact, Λ plays a role which is equivalent to that of a functional lexicon in terms of how it introduces information into the derivation; the only real difference is that it does not introduce structure).

I conclude that the system presented here is an improvement over the standard Bare Phrase Structure system, at least inasmuch as it sidesteps the Labeling and Specifier Problems but does not increase the general complexity of the system. LTFs replace the lexicon of functional categories, the equivalent of extended projections are required to organize structures in both systems, but in the new system Merge is simplified and there is a unified labeling algorithm. Structure building is sharply separated from labeling, which is taken to be dependent on a language particular instantiation of a universal sequence of categories.

In the next section I’ll explore some of the new system’s theoretical and analytical consequences.

1.3 Syntactic Interpretation

In the new system, complement and specifier are not structurally distinguished by derivational timing, as in the standard First Merge vs Second Merge definitions (Chomsky 1995). In fact, as far as the syntactic computation goes, all cases of Merge give a perfectly symmetrical structure.

However, this seems to be empirically incorrect. Numerous asymmetries must be determined for appropriate interpretation by the interface systems. For example, for a label that is interpreted as a relation, two separate arguments have to be identified (e.g. v^* needs to determine which of its two dependents is the agent and which is the event). Similarly, the spellout systems seem to be sensitive to whether a moved expression can appear to the right or the left of the structure which it targets (that is, movement to a right specifier seems to be disallowed, at least in spoken languages). In the theory developed here, the only information available to the derivation of the structure is the extended projection (EP) of the root built via Label. Let us then define the notions of i-complement and i-specifier based on the existence of an EP relation in a structure:

- (42) a. In a labeled structure where $\alpha \in \gamma$,
 if α and $\gamma \in$ the same (Instantiated) EP and $\text{Label}(\gamma) > \text{Label}(\alpha)$ in that EP
 then α has an i-complement relation to γ
 b. In a labeled structure where $\alpha \in \gamma$,
 if α and $\gamma \notin$ the same (Instantiated) EP or $\text{Label}(\gamma) < \text{Label}(\alpha)$ in that EP
 then α as an i-specifier relation to γ

I take $>$ and $<$ to be defined so that they are sensitive to the linear sequence defined by an Extended Projection.⁶

These definitions require us to have a notion of ‘same Instantiated EP’ (I will drop the ‘instantiated’ from now on, as it will be clear whether we are talking about a syntactic object or a universal organization of labels). I take the relevant notion of sameness to be given by the start category of the EP. An EP starting in N which is the label of the self merge of $\sqrt{\text{cat}}$ is the same EP as an EP starting in N which is the label of the self Merge of $\sqrt{\text{carburetor}}$, and it is not the same EP as one starting in V (which labels, say, the self Merge of $\sqrt{\text{jump}}$). This definition will do some work for us in cases where an EP is started by a root ‘above’ N, V, or A (e.g. if an auxiliary like *will* roots an EP which then starts in the category Modal. This will not be the same EP as that started by V and rooted in, say, $\sqrt{\text{jump}}$.)

We also have a further relation of label-identity between mother and daughter:

- (43) In a labeled structure where $\alpha \in \gamma$,
 if α and $\gamma \in$ the same (Instantiated) EP and $\text{Label}(\gamma) = \text{Label}(\alpha)$ in that EP
 then α has a label-identity relation to γ

Given the way that i-complement and i-specifier are defined, they are almost converses of each other, but not quite, at least under the usual interpretation that the negation of $>$ is \leq rather than $<$. This is because label-identity is taken to be a separate relation.

I-complement and i-specifier relations hold between mothers and daughters and, when there is more than one daughter, provide an asymmetry which can be exploited by the interface systems, both for meaning and linearization.

I will also assume the following two conditions on labeling. The first is an input condition:

- (44) Informational Anti-Entropy: In a structure $\gamma = \{\alpha, \beta\}$, Labeling of γ must increase

⁶The notion that specifiers are ‘higher’ in their extended projection than their host was briefly explored in Svenonius 2004.

the information in the structure.

(44) ensures that any application of labeling to a newly constructed object involves an increase in the information in that object. The information in the structure increases when a new LTF is used which introduces a new transition from one label to another. The reason to adopt (44) is that it rules out situations where structures built by Merge are all labeled with the same label (see below).

As well as this input condition on labeling, there is also an output condition ensuring asymmetry of interpretation:

(45) Asymmetry of Interpretation: In a labeled structure $\gamma = \{\alpha, \beta\}$, γ cannot bear the same relation to different daughters.

We take the relevant relations here to be i-complement, i-specifier and l(abel)-identity. As we will see, (45) rules out cases where two i-complement (or i-specifiers) relations are determined for a single mother.

The definitions of i-specifier and i-complement mean that in a structure like (46), we can uniquely determine both the i-complement and the i-specifier:

(46)

$$\begin{array}{c} X_4 \\ \swarrow \quad \searrow \\ Y_{10} \quad X_3 \end{array}$$

Here the subscripts give the ‘height’ that the category has in the relevant (instantiated) extended projection of some root, and the identity of the letter specifies sameness of that extended projection (e.g., all X’s are in the extended projection of V, and all Ys in the extended projection of N). Given our definitions, X_3 is the i-complement of X_4 , since it is both in the same EP and lower in that EP, while Y_{10} is in a different EP, and hence is the i-specifier of X_4 . Of course Asymmetry of Interpretation is satisfied because the mother bears a different relation to each daughter.

I-complement and i-specifier are equally defined in (47), where X_{10} is an i-specifier of X_4 because, even though it is in the same EP as X_3 , it is higher in that EP.

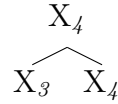
(47)

$$\begin{array}{c} X_4 \\ \swarrow \quad \searrow \\ X_{10} \quad X_3 \end{array}$$

If X_{10} and X_3 have the same (token) root, (47) would be impossible to derive since X_{10} would have to have moved from inside X_3 , and this would entail a violation of the order of the extended projection. However, if X_{10} and X_3 are built above different roots, which give rise to the same extended projection, then our definitions lead to X_3 being the ‘head’, since the i-complement relation is defined between X_4 and X_3 .

A more interesting case is (48). No i-specifier is defined: the daughter X_4 is not an i-specifier of the mother, as it is not higher in the EP. However, X_3 is an i-complement of the mother, and the lower X_4 bears the label-identity relation to its mother, so Asymmetry of Interpretation is satisfied:

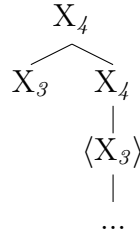
(48)



There are two possibilities for how this structure is derived. The first is that it arises via movement of X_3 from inside X_4 . However, this would entail the following kind of derivation:

- (49)
- a. build X_3
 - b. Self-Merge X_3 and Label the structure as X_4 , using a label transition function $\langle 3,4 \rangle$
 - c. Merge X_3 with X_4
 - d. Label resulting structure, using label transition functions $\langle 3,4 \rangle$, $\langle 4,4 \rangle$

(50)

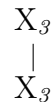


Such a derivation violates Informational Anti-Entropy, since $\langle 3,4 \rangle$ has already been used and since $\langle 4,4 \rangle$ is recursive and no new label is introduced. Whatever requirements obtain between X_4 and X_3 have already been met earlier in the derivation. Informational Anti-Entropy bars unmotivated operations (it is an economy condition).

The other possibility is that X_3 and X_4 in (48) have different roots, which, however, have the same extended projection. In this case X_3 is an i-complement of X_4 . Asymmetry of Interpretation is satisfied as X_4 bears one relation to one daughter (i.e. label-identity to the lower X_4), and a different relation to the other (i-complement).

In unary branching structures which simply recurse, neither an i-complement not an i-specifier can be established, leaving only l-identity:

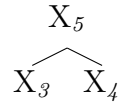
(51)



However, although $\langle 3,4 \rangle$ is a new LTF, no new label is introduced, and so (51) falls foul of Informational Anti-Entropy. Mindless iteration of labeling is ruled out.

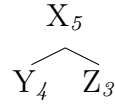
Another possible structure would be as follows:

(52)



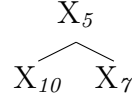
In such a structure, both X_4 and X_3 satisfy the conditions to be i-complements of X_5 , but Asymmetry of Interpretation requires that X_5 bear a different relation to each daughter. In (52), this is impossible, since there are two i-complements. Of course, the same would hold when both daughters are i-specifiers:

(53)



Since neither Y nor Z is in the same EP as X , they both count as i-specifiers, violating Asymmetry of Interpretation. If both daughters are higher but in the same EP, then once again they both count as i-specifiers:

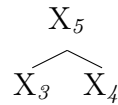
(54)



The same holds for any mix of higher X s or Y/Z s.

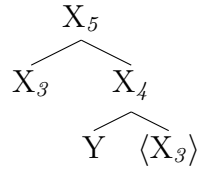
An important consequence of this system is that it rules out certain rollup movement derivations within a single extended projection. Consider again the structure where we have two i-complements:

(55)



Exactly this kind of configuration will arise if X_3 is raised from within X_4 :

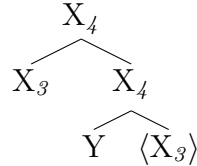
(56)



Here both X_3 and X_4 satisfy the conditions to be i-complements of X_5 , so this labeling violates Asymmetry of Interpretation.

The only possible way to generate this configuration would be to recurse X_4 :

(57)

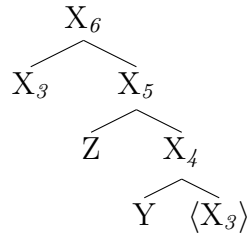


Here X_3 bears an i-complement relation to the higher X_4 , while the lower X_4 bears a label-identity relation to the higher X_4 , satisfying Asymmetry of Interpretation.

But in this situation, the label transition function that requires X_3 to be contained within X_4 is already satisfied (since X_3 is in X_4 before any movement takes place), which violates Informational Anti-Entropy. In this case, the effect of that input constraint on labeling effectively implements the idea that movement must be motivated. In (57), there is no motivation for movement of X_3 .

Furthermore, a remnant roll-up derivation that extracts some Z from X_3 , will require a category higher than X_3 (for the same reasons—if Z is to move from inside X_3 it must be moving to some higher layer in the extended projection, or the structure would be ‘satisfied’ with its in-situ position):

(58)



Once again, X_6 has two i-complements, ruling out this derivation.

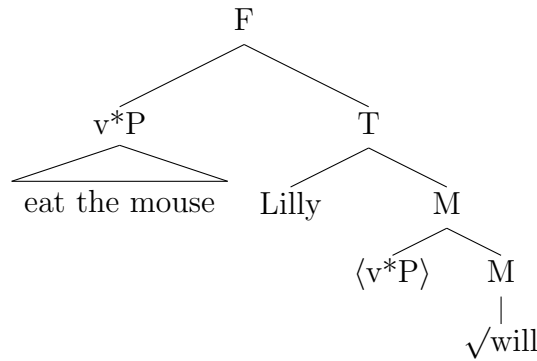
In order for a rollup or rollup-remnant movement derivation to take place, it must be the case that the movement is to a different extended projection.

For example, in a VP topicalization construction such as (59), the moved VP must actually be part of a different extended projection from the auxiliary:

(59) ... and eat the mouse Lilly certainly will!

This rules out a set of standard analyses for these cases (where *Lilly* raises from the vP-internal subject position, followed by movement of the vP to some focus position in the C-domain). However, if the modal in (59) begins its own (instantiated) EP, as in (60), then the non-finite VP can be in a specifier of the auxiliary, and can raise:

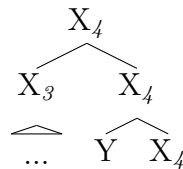
(60)



In this configuration, there are two relevant EPs: that started by M (call it EP_M) and that started by V (EP_V). The SO labeled v^* is in EP_V but not in EP_M (since EP_M starts at M, which is higher than v^*), therefore v^*P is an i-specifier of M. Once v^*P moves to F, it is still in EP_V and not EP_M , and hence still an i-specifier.

Our definitions do actually allow a situation where the complement of a category (that is, a structure labeled with a lower category in the same EP) may appear structurally higher than a specifier of that category, mimicking the roll-up movement structure. This happens when we have an l-identity relation to one of the daughters in a base generated structure. For example, if we could start an EP with X_4 , we could do the following:

(61)



In this case X_3 is an i-specifier of X_4 if X_4 can ‘start’ a different EP. However, it can also be

interpreted as an i-complement of the highest X_4 , since we can take the instantiated EP to begin inside X_3 . We will return directly to this case in the next section.⁷

Our definitions of specifier and complement are steadfastly non-structural and contrast with the usual derivational definitions (First Merge vs Subsequent Merge) or representational definitions (e.g. sister of an X^0 ; sister of \bar{X}). They are more similar to the notions of interpretive specifier and selected dependent in the representational system of Brody and Szabolcsi (2003). However, the definitions here do share an important commonality with the standard view, in that specifier is taken to be (almost) the converse of complement. Complements are defined in terms of extended projections, and specifiers are a byproduct of that definition.

2 Implications of the Proposal

2.1 No complements of lexical roots

The system set up above has an interesting corollary: when a root is Merged, Asymmetry of Interpretation requires the two elements in the structure to bear a different relation to the mother. To determine whether this holds, we inspect the relevant extended projection. However, elements of RLex (that is lexical roots) are not in an extended projection with their mother when they Self Merge—they don't have a label until they are Self Merged. The definitions of i-complement and i-specifier, however, require the labels of the relata. This means that roots are neither the i-complements nor the i-specifiers of the structure that immediately contains them. This, in turn, comports with the fact that they constitute the base case for labeling recursion.

The relation between a root and the category that contains it in a unary branching structure is, then, not strictly 'in the syntax', either for labeling purposes, or for the purposes of Asymmetry of Interpretation. However, what of a situation where a root directly Merges with a previously constructed syntactic object. How would this be labeled?

⁷One further structure remains to be considered: a unary branching complement line where each X_n is contained in some X_{n+m} will be well formed, as an i-complement relation can always be determined. But what of a unary structure where there is a single i-specifier:

- (i)
$$\begin{array}{c} X_4 \\ | \\ Y_3 \end{array}$$

Here Y has Self Merged, and there is an LTF taking us from Y_3 (say a projection of V) to X_4 (say a nominal functional category).

I will not consider these in the discussion here. However, in this structure we are missing a root for one of the extended projections. If all projections are built above a root category, then these could be ruled out as follows:

- (ii) **Rootedness:** If $l \in \text{CLex}$ labels an SO, there must be a containment path of i-complement relations to a root (i.e. the semantic interpretation of each category requires an (extended) concept to apply to).

The answer to the question of whether Rootedness holds will depend on the analysis of category changing morphology, which is not my focus here.

Recall the definition of the function Label:

- (62) a. If $\alpha, \beta \in \gamma$, then $\text{Label}(\gamma) = \text{some } L \in \text{CLex}$, such that there are (possibly non-distinct) f and $g \in \Lambda$ such that $f(\text{Label}(\alpha)) = g(\text{Label}(\beta)) = L$. (**Transition Labeling**)
 b. $\text{Label}(\{\sqrt{x}\}) = \text{some } L \in \{N, V, A\}$ (**Root Labeling**)

If some root (for example $\sqrt{\text{picture}}$) were to merge with some previously constructed SO (let's say the PP *of Lilly*), then we have (simplifying the structure of *of Lilly*):

- (63) a. $\text{Merge}(\sqrt{\text{picture}}, \{\text{of Lilly}\}) = \{\sqrt{\text{picture}}, \{\text{of Lilly}\}\}$
 b. $\text{Label}(\{\sqrt{\text{picture}}, \{\text{of Lilly}\}\}) = L$ if there are LTFs $f, g \in \Lambda$ such that $f(\text{Label}(\sqrt{\text{Lilly}})) = g(\text{Label}(\{\text{of Lilly}\})) = L$
 c. by hypothesis, $\text{Label}(\{\text{of Lilly}\}) = P$, and assume that there is an LTF $g = \langle P, N \rangle$
 d. But $\sqrt{\text{picture}}$ is not in the domain of Label, since Root Labeling applies only to $\{\sqrt{\text{picture}}\}$

This means that no Label can be determined. We therefore rule out the following:

- (64) $*\{\sqrt{\text{root XP}}\}$

We will explore this consequence for nominals in later chapters, but in brief it means that expressions like (65) cannot have the structure attributed to them by most theories of syntax since Chomsky 1970 (but cf. Kayne 2010):

- (65) The colour of the car

Rather the structure of such examples must be as follows:

- (66) $[[_N \sqrt{\text{root}}] \dots \text{PP}]$

That is, the PP ‘complement’ is Merged in a position outside of the projection of the lexical root.

This then raises two analytical questions:

- (67) a. The Ordering Question: given the PP is hierarchically above the N, why is it to its right, assuming that syntax disallows rightward specifiers (Kayne 1994);
 b. The Aetiology Question: given the PP is hierarchically above the N and outside of its projection, how is the semantic relation between the noun and the PP negotiated?

The same issues, of course, arise for verbal structures. The following is ruled out:

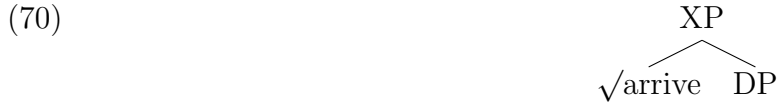
- (68)
- $$\begin{array}{c}
 V \\
 \swarrow \quad \searrow \\
 \sqrt{\text{arrive}} \quad \text{DP}
 \end{array}$$

Rather we have:

(69) $[[_V \sqrt{\text{root}}] \dots \text{DP}]$

Here the DP must actually be a specifier of some element within the extended projection of V. There are of course many proposals that separate the root from its object, generating arguments of the verbs in specifier position (Travis 2000, Borer 2005b, Ramchand 2008 etc); on the theory of phrase structure developed here, the alternative standard view is not an option.

It is important to see just where the system developed here differs from the standard system. In the standard system it is possible to Merge an XP with a root, as follows:



Moreover, this structure is usually interpreted as involving an internal argument of a predicate.

On the proposal here, what is available is:



but in addition, we allow various elements of functional structure to appear before introduction of the specifier:



That is, the current system allows the dissociation of argument introduction from the lexical root entirely, something which is unexpected on the standard view. We are then left with an empirical question: is there evidence for such dissociation? That is, do we find cases where syntactic functional structure is built above a lexical root before the introduction of the argument. If we do, then the current system is superior to the standard one.

Within the empirical domain of verbs, it is now usually assumed that the internal argument is a complement, and that all functional structure that is associated with argument introduction is Merged in a single domain (Ramchand (2008)'s 'first phase'). However, the question is open for arguments of pure (that is, underived) nominals, and it is these that I build the argument on here in the remaining chapters. I will argue that even for complements of apparently relational nominals, a fair amount of functional structure, including that supporting adjectives, can be Merged before the PP complement is added. The syntactic

system developed here has this possibility as a consequence, while it would be a surprise under standard assumptions.

Before turning to the empirical arguments, I draw out two more important properties of the system developed here.

2.2 Order

The system developed here produces purely symmetrical structures, and so, obviously, cannot adopt the Linear Correspondence Axiom of Kayne (1994) directly. However, syntax, as argued forcefully by Kayne and others, does appear to be rather asymmetrical in terms of how syntactic operations relate to linear order. I propose we stipulate this into the system as follows:

(73) An i-specifier is on a left branch.

Assume 3 categories A, B and C and the following information (where f, g, h ... are LTFs):

(74) a. $f(B) = g(C) = A$
b. $B, A \in \text{the same EP}$

With this information about extended projection, the tree in (75-a) is ruled out, since C is an i-specifier but is on a right branch:

(75) a. *

$$\begin{array}{c} A \\ \swarrow \quad \searrow \\ B \quad C \end{array}$$

b.

$$\begin{array}{c} A \\ \swarrow \quad \searrow \\ C \quad B \end{array}$$

In general, this system will also disallow internal Merge of an i-specifier to a higher rightwards position, since internal Merge will create a new i-specifier structure (cf also Abels and Neeleman 2007 who attribute this to a parsing effect). Recall that movement of an i-complement to a higher position via a roll-up derivation is ruled out by Asymmetry of Interpretation.

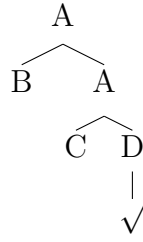
However, there are cases where specifier like elements appear to the right of their associated complements in surface order. To tackle this, I'd like to follow Brody and Szabolcsi (2003) and Adger, Harbour, and Watkins (2009) and take it to be an effect of base generating apparent complements as higher specifiers.

Following the core idea, although not the execution, of Brody and Szabolcsi (2003), we allow the following LTFs in Λ , where we take D to be the start of the EP given by Root Labeling.

(76) $f(B) = g(C) = h(D) = i(A) = A$

Since $i(A) = A$, we have recursion on A:

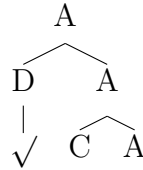
(77)



Recursion of A here allows multiple left daughters. On the assumption that $A, D \in EP_i$, but $C, B \notin EP_i$, then each SO conforms to Asymmetry of Interpretation, since D is i-complement and C the i-specifier of the lower A, while B is the i-specifier of the higher. The system as it stands then allows multiple specifiers.

We might also allow:

(78)



Continuing to assume that D and A are in the same EP, this would give us a situation where D is an i-complement of A, since the EP built above the root contains D and A and D satisfies the conditions to be the i-complement of A, while C would be an i-specifier. This would correspond to an order here the i-specifier of a category appears to the right of the apparent i-complement of that category, thus base generating roll-up structures.

2.3 Introducing Arguments

In general, however, the kind of derivation sketched in the previous subsection as a means of reversing the order of complement and specifier will be ruled out. This is because, if A is above D in an extended projection (as it needs to be given the requirement that labeling functions are monotonically increasing), A will be an element of CLex, and hence the lowest subtree of (78) will not be a syntactic object, since the structure labeled by A does not contain a root (recall that A is not a functional head, it is a label for structure, and all structure is built, ultimately, from roots).

However, this argument contains a hidden assumption. Our basic architecture is:

- (79)
- a. $R\text{Lex} = \{\sqrt{1}, \dots, \sqrt{n}\}$, the set of LIs
 - b. $C\text{Lex} = \{l_1, \dots, l_n\}$, the set of category labels
 - c. $\Lambda = \{f_1, \dots, f_n\}$, the set of label transition functions in LxL .

We have been assuming that RLex and CLex are disjoint so far, which will rule out the structure displayed in (78). However, if $R\text{Lex} \cap C\text{Lex} \neq \emptyset$, then we have the possibility of a ‘semi-functional’ category, which can both be an input to Merge and a category label.

- (80)
- a. $f(B) = g(C) = h(A) = A$
 - b. $B, A \in C\text{Lex}$
 - c. $A \in R\text{Lex}$

In fact much empirical work has proposed such ‘semi-functional’ or ‘light’ categories over the years, including ‘little v’ and ‘poss’, the head that introduces possessors. In general, previous theoretical accounts of such categories take them to have the argument taking properties traditionally associated with lexical categories, but the syntax and semantics of functional categories. In the system developed here, the existence of a special set of categories follows from imposing no disjointness condition on the membership of RLex and CLex. My suggestion is that human language uses precisely this set of categories to introduce arguments. That is, the notion of theta-domain is otiose; arguments are introduced by categories that have both root like, and functional category-like properties (cf. the Voice category of Kratzer (1996) and the applicative categories of Pylkkänen (2008)).

We can then say:

$$(81) \quad \text{RLex} \cap \text{CLex} = \{ v^*, \text{poss}, \dots \}$$

Here v^* and poss , although elements of RLex, are legitimate labels, so LTFs can apply to them directly.

Take v^* , for example. We gave a derivation for an unergative verb above which posited the following structure:



This is licensed by Λ containing:

$$(83) \quad f(V) = g(D) = v^*$$

However, assume v^* is also a member of RLex (taking roots to generally have ‘conceptual’ meanings, one might assume that v^* means something like ‘act’ or ‘do’ or ‘cause’, and indeed there may be different ‘flavours’ of v^* with different root meanings but the same category. See Folli and Harley (2007). and that Λ contains $h(v^*) = v^*$, then, in addition to the kind of derivation that led to (82), we also have the following:



The derivation of this is as follows (I use category labels as shorthand for SOs in this derivation):

- (85)
- a. $\text{Merge}(\sqrt{\text{Lilly}}, \sqrt{\text{Lilly}}) = \{\sqrt{\text{Lilly}}\}$
 - b. $\text{Label}(\{\sqrt{\text{Lilly}}\}) = D$ (Root Labeling)
 - c. $\text{Merge}(\sqrt{v^*}, \{\sqrt{\text{Lilly}}\}) = \{\sqrt{v^*}, \{\sqrt{\text{Lilly}}\}\}$

- d. $\text{Label}(\{\sqrt{v^*}, \{\sqrt{\text{Lilly}}\}\}) = v^*$ since $\exists h, g: g(D) = h(v^*) = v^*$
- e. $\text{Merge}(\sqrt{\text{jump}}, \sqrt{\text{jump}}) = \{\sqrt{\text{jump}}\}$
- f. $\text{Label}(\{\sqrt{\text{jump}}\}) = V$ (Root Labeling)
- g. $\text{Merge}(\{\sqrt{v^*}, \{\sqrt{\text{Lilly}}\}\}, \{\sqrt{\text{jump}}\}) = \{\{\sqrt{v^*}, \{\sqrt{\text{Lilly}}\}\}, \{\sqrt{\text{jump}}\}\}$
- h. $\text{Label}(\{\{\sqrt{v^*}, \{\sqrt{\text{Lilly}}\}\}, \{\sqrt{\text{jump}}\}\}) = v^*$ since $\exists h, f: f(V) = h(v^*) = v^*$

In terms of Asymmetry of Interpretation, when the subject and the ‘light verb’ v^* Merge, the new SO is labeled v^* , and it can be related to its left daughter only via the i-specifier relation (*Lilly* is not in the same EP as v^*). At the next level up, following Merge of the main verb *jump*, the new SO is labeled v^* again. The issue is then whether this bears an i-complement or an i-specifier relation to its left daughter.

Let us consider the topmost v^* and the v^* sister to V . These two labels are in the same EP (EP_{v^*}) and bear a relation of label identity. What of the topmost v^* and V ? These are both in EP_V rooted in $\sqrt{\text{jump}}$, and hence could be related by an i-complement relation. However, they are also members of different EPs. v^* is in EP_{v^*} but V is not. On this basis, V is an i-specifier to v^* . On either parse, Asymmetry of Interpretation is satisfied. As far as the systems of semantic interpretation are concerned, V can be taken to be the i-complement of v^* , just as in the standard derivation. Furthermore, the label v^* still has an i-specifier (*Lilly*), so the semantic relation v^* specifies between *Lilly* and *jump* works just as usual. However, V is also an i-specifier, so if it were to move further, it could bear an i-specifier relation to a higher element in the extended projection of v^* . We effectively have a dual rooted extended projection in just this one case, where a single element is both a member of RLex and CLex.

This system as a whole now gives us two ways of generating arguments: they are either i-specifiers of functional categories in a single rooted EP (as in the standard ‘little v ’ analysis), or they are i-specifiers of a rooted yet functional category, in a dual rooted EP.⁸ In the latter case, the prediction is that the argument may come after the ‘main’ verb (although this is not quite forced), with the constituent corresponding to the i-complement being Merged as a higher left daughter.

I propose that this is exactly the derivation that is found when arguments are introduced by prepositional elements, and that the prepositional elements are a realization of this extra structure. That is, if English allowed (86), then it would be derived as above:

(86) *It was jumped by Lilly.

Of course, English does not allow (86) with unergatives, however, many languages do (see, e.g. Perlmutter 1978):

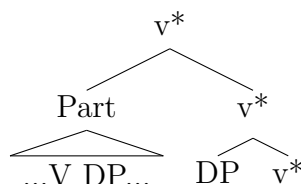
(87) Er wordt door de kinderen op het ijs geschaatst
 It was by the children on the ice skated
 ‘The children skated on the ice.’

⁸As a historical aside, the two derivations proposed here for argument introduction are reminiscent of the two approaches to argument introduction explored within generative semantics, for example Ross (1972) and Fillmore (1968), where arguments are introduced as subjects of higher predicates, or as nominals dominated by elements with pure semantic role labels.

Furthermore, the structure we have derived bears a marked similarity to a smuggling structure for passives (Collins 2005), although base generated rather than derived via VP movement, and we can appeal to its properties in much the same way that Collins does to answer questions about the passive.

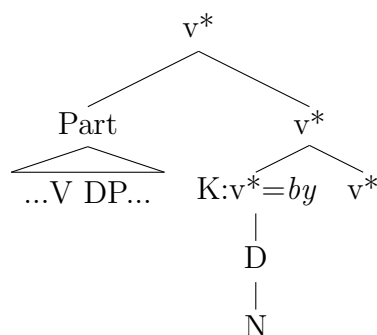
Collins proposes that the participial VP raises into the specifier of a functional head he calls Voice, projected above vP, with the true object then raising from inside this specifier position to the surface subject position. He takes Voice then to be realized as the preposition *by* in English. The analysis proposed here takes this structure to be generated not via movement, but directly. There is no lower VP trace:

(88)



In Collins' proposal, the preposition *by* is the realization of a higher Voice head. I will propose in following chapters that prepositions are actually the spellout of a piece of nominal functional structure (the case projection K) and a piece of verbal functional structure (in this situation, v*), so that in (89), the structure K+v* is spelled out at K as *by*. This can be implemented by taking K's category feature to have v* as a value:

(89)



Of course, this structure is also similar to the 'flipped' structures argued for by Brody and Szabolcsi (2003) and especially Adger, Harbour, and Watkins (2009). While Brody and Szabolcsi use such flipped structures rather generally, and argue that they are restricted by the mapping to morphology, the system here restricts these structures to argument introducing heads. This comports with the major arguments developed by Adger, Harbour, and Watkins (2009) who show the usefulness of such 'flipped' structures in capturing reversed base generated orders of arguments in Kiowa.⁹

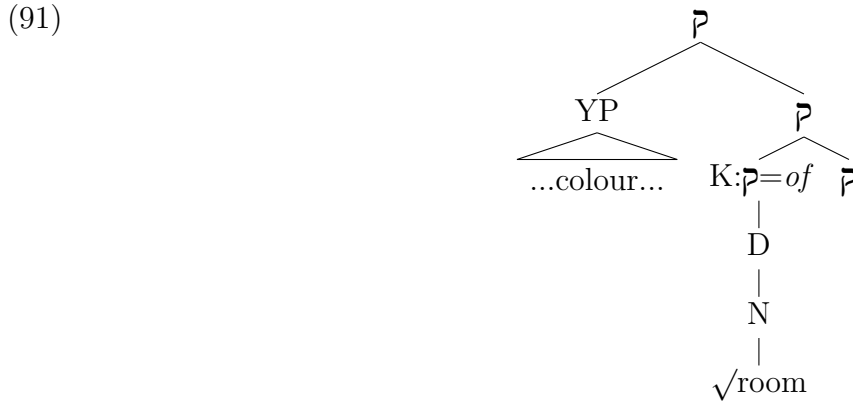
This proposal for the introduction of arguments is what we will exploit in answering the question of the aetiology of PP 'complements' to N, that was raised in section 2.1: in (90), how is the semantic relationship between the root and the PP negotiated?

⁹Aside from rightwards arguments introduced by PPs, these flip structures have been proposed for adjuncts to VPs. My system does not allow this, unless these are in fact really semantic arguments of some kind. My hunch is that they are not, and that adjuncts are integrated into structure via some other means (for example, that proposed in Chomsky 2004).

(90) $[[_N \sqrt{\text{colour}}] \dots [_{PP} \text{ of the room}]]$

The answer will be as follows: PP is the i-specifier of an element of $R\text{Lex} \cap C\text{Lex}$ (that is, a category which is in both the EP of N, and is an LI). Rather than the N s-selecting the PP complement, a functional category notated by the Hebrew letter Qof, \P , which is a member of both the root and the non-root ‘lexica’, relates its two arguments.¹⁰ \P is then the semantic ‘head’ of this structure, although the lexical content is given by N. I will show how a straightforward extension of Katzer’s (1996) semantics for voice gives us the right semantic technology for the syntactic structures I motivate.

We further have an answer to the Ordering Question: The PP can come to the right of its N because the N can be generated as a higher left daughter, so that the PP is still a left daughter and i-specifier of a lower layer of labeling. So rather than (90), we have:



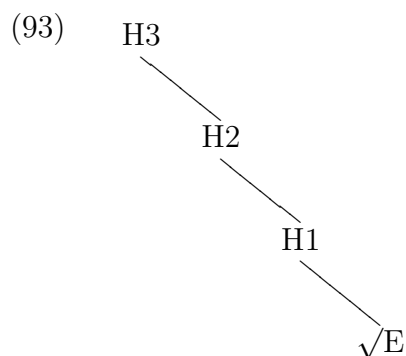
2.4 Spellout of Functional Categories

The denial of the existence of functional heads forces us into a position which takes the phonology and morphology of functional morphemes to be read off of labeled structures, rather than the standard position, where they are functional heads Merged as independent pieces of structure. There are two cases to consider:

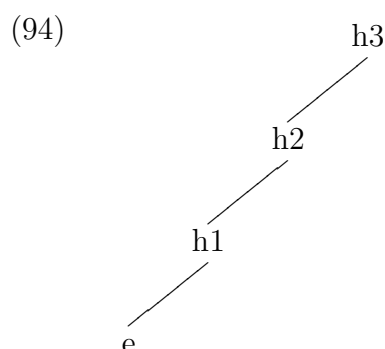
- (92)
- a. bound functional morphemes (bfms)
 - b. free functional morphemes (ffms)

For bfms, I adopt a version of the approach advocated by Brody (2000a). Take a structure like the following, where each H is in the same extended projection with E as the root (I give the tree here slanted towards the right, as Brody does. In the current system, of course, no such slant is necessary for unary structures):

¹⁰Thanks to Daniel Harbour for the suggestion to use \P , which sounds like a K but looks like a P, thus neatly capturing the prepositional but also case (K) marking properties of the head.

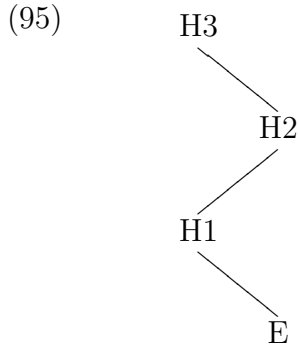


One of the insights of Brody's system is that the syntactic complement line (our i-complement line) corresponds to a morphological structure. In (94) each h is the morpheme corresponding to the category H:



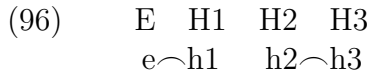
For Brody, (94) is a morphological specifier structure, with a general principle, that specifiers precede heads, ensuring the linear order of the affixes. I do not adopt Brody's Mirror Axiom here, so I will assume that the linear order is stipulated for elements in a complement line (and, in fact, may be dependent on the linearization properties of particular morphemes as in Bye and Svenonius 2010). However, the scope order of the labels of syntactic structure is, at heart, the source of the sequential order of affixes.

Turning to ffms, in Brody's system (e.g Brody 2000a, Brody 2000b), if x is the complement of y then y is suffixed to x, that is, the syntactic complement line corresponds to a morphological specifier relation (this is the Mirror Axiom of Brody's theory). It then follows that if y is not suffixed to x, then x cannot be the complement of y. This leads Brody to take separate morphological words in the same extended projection to involve a 'wiggly' complement line. For example, if $e \frown h1$ is a word, and $h2 \frown h3$ is a word, then the structure will look as follows:



Here, H1 is a specifier of H2, which means that H1 and H2 do not correspond to a single morphological word. The morphological words in this structure are those that correspond to (H2,H3) and (E,H1).

There is an alternative to the ‘wiggly word’ approach, sketched in Williams (2003). Williams does not assume that there is a Mirror Axiom. Rather he takes morphological words to ‘span’ sections of an extended projection, as follows:



Here the complement line of functional categories above the root is H1-H2-H3-E but the free lexical word is the bimorphemic $e \curvearrowright h1$, which ‘spans’ the structure H1-E. The free functional word $h2 \curvearrowright h3$ ‘spans’ the structure H3-H2.

This approach makes immediate sense of fusional morphology, in that a single morpheme can correspond to a number of functional category labels. Rather than the bimorphemic $h2 \curvearrowright h3$, we could have the single ‘fusional’ morpheme $h5$ spanning H3-H2. I will adopt this approach to ffms in what follows.

Following Brody, we can assume that it is a property of the label of a SO that is responsible for where a sequence of morphemes is spelled out, so that while one language spells out $h2 \curvearrowright h3$ at H2, another might spellout $h2 \curvearrowright h3$ at H3, with concomitant ordering effects if H2 and H3 have specifiers. This will be a second order feature of the label, acquired during the acquisition of Λ .

In the system developed here, then, there are no functional categories qua lexical items, which means that free functional morphemes must either be spellouts of fragments of structure, or, that they must actually have lexical roots in them (for example, this is probably the case for auxiliaries as briefly discussed above). As already mentioned, at least some prepositions (the argument introducing ones) are ‘spans’ of a case category K in the EP of the nominal and a relational category (\mathfrak{P} , POSS, v^* etc) in another EP.

3 Conclusion

I have argued in this chapter for a new view of how structures are built and labeled. The resulting system provides a unified solution to the problem of labeling, and also, once augmented with definitions of interpretive specifier and complement, a new perspective on argument introduction, which correlates different styles of argument introduction with linear

order and, as we will see, prepositional morphology. The theory also disallows direct Merge of roots with complements, effectively forcing arguments to be generated as specifiers.

References

- Abels, Klaus and Neeleman, Ad. 2007. Left right asymmetries and the LCA, lingbuzz <http://ling.auf.net/lingBuzz/000279>.
- Abney, S. P. 1987. The English noun phrase in its sentential aspect. Ph.D. thesis, MIT, Cambridge MA.
- Adger, David. 2003. *Core Syntax*. Oxford: Oxford University Press.
- Adger, David. 2010a. A minimalist theory of feature structure. In Anna Kibort and Greville Corbett, eds., *Features: Perspectives on a Key Notion in Linguistics*, 185–218, Oxford: Oxford University Press.
- Adger, David. 2010b. Variability and grammatical architecture, lingBuzz/001176.
- Adger, David, Harbour, Daniel, and Watkins, Laurel. 2009. *Mirrors and Microparameters: Phrase Structure beyond Free Word Order*. Cambridge: Cambridge University Press.
- Adger, David and Ramchand, Gillian. 2005. Move and Merge: *Wh*-dependencies revisited. *Linguistic Inquiry* 36:161–194.
- Adger, David and Svenonius, Peter. 2011. Features in minimalist syntax. In Cedric Boeckx, ed., *Oxford Handbook of Linguistic Minimalism*, 27–51, Oxford, UK: OUP.
- Baker, Mark. 2003. *Lexical Categories*. Cambridge, U.K.: Cambridge University Press.
- Boeckx, Cedric. 2008. *Bare Syntax*. Oxford, UK: Oxford University Press.
- Boeckx, Cedric. 2010. Defeating lexicocentrism, lingbuzz/001130.
- Borer, Hagit. 2005a. *In Name Only: Structuring Sense*. Oxford: Oxford University Press.
- Borer, Hagit. 2005b. *The Normal Course of Events: Structuring Sense*. Oxford University Press.
- Bowers, John. 1993. The syntax of predication. *Linguistic Inquiry* 24:591–656.
- Brody, Michael. 2000a. Mirror theory: syntactic representation in perfect syntax. *Linguistic Inquiry* 31:29–56.
- Brody, Michael. 2000b. Word order, restructuring and mirror theory. In Peter Svenonius, ed., *The Derivation of VO and OV*, volume 31 of *Linguistik Aktuell*, 27–43, John Benjamins.
- Brody, Michael and Szabolcsi, Anna. 2003. Overt scope in Hungarian. *Syntax* 6:19–51.
- Bye, Patrik and Svenonius, Peter. 2010. Exponence, phonology, and non-concatenative morphology, lingBuzz/001099.

- Cecchetto, Carlo and Donati, Caterina. 2010. On labeling: Principle C and head movement. *Syntax* 241–278.
- Cecchetto, Carlo and Donati, Caterina. to appear. Relabeling heads: A unified account for relativization structures. *Linguistic Inquiry*.
- Chomsky, Noam. 1961. On the notion “rule of grammar”. In Roman Jakobson, ed., *Structure of Language and its Mathematical Aspects*, volume XII of *Proceedings of the Twelfth Symposium in Applied Mathematics*, 6–24, Providence, Rhode Island: American Mathematical Society.
- Chomsky, Noam. 1970. Remarks on nominalization. In R. A. Jacobs and P. S. Rosenbaum, eds., *Readings in English Transformational Grammar*, Waltham, MA: Ginn-Blaisdell.
- Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.
- Chomsky, Noam. 2000. Minimalist inquiries: the framework. In R. Martin, D. Michaels, and Juan Uriagereka, eds., *Step by Step: Essays on Minimalist syntax in Honour of Howard Lasnik*, 89–115, Cambridge, MA: MIT Press.
- Chomsky, Noam. 2004. Beyond explanatory adequacy. In Adriana Belletti, ed., *Structures and Beyond*, 104–131, New York: Oxford University Press.
- Chomsky, Noam. 2007. Approaching UG from below. In Hans-Martin Gärtner Uli Sauerland, ed., *Interfaces + Recursion = Language?*, Berlin and New York: Mouton de Gruyter.
- Chomsky, Noam. 2008. On phases. In Robert Freidin, Carlos P. Otero, and Maria Luisa Zubizarreta, eds., *Foundational Issues in Linguistic Theory*, 133–166, Cambridge, MA: MIT Press.
- Cinque, Guglielmo. 1999. *The Syntax of Adverbs*. Oxford: Oxford University Press.
- Citko, Barbara. 2008. Missing labels. *Lingua* 118:907–944.
- Collins, Chris. 2002. Eliminating labels. In Samuel David Epstein and T. Daniel Seely, eds., *Derivation and Explanation in the Minimalist Program*, 42–64, Blackwell Publishing.
- Collins, Chris. 2005. A smuggling approach to the passive in English. *Syntax* 8:81–120.
- Collins, Chris and Stabler, Edward. 2009. A formalization of minimalist syntax, ms NYU and UCLA.
- Donati, Caterina. 2006. On wh-head-movement. In Lisa Lai Shen Cheng and Norbert Corver, eds., *Wh-Movement on the Move*, 21–46, Cambridge MA: MIT Press.
- Fillmore, C. 1968. The case for case. In C. Bach and R. Harms, eds., *Universals in Linguistic Theory*, 1–81, New York: Holt, Rhinehart and Winston.
- Folli, Raffaella and Harley, Heidi. 2007. Causation, obligation, and argument structure: On the nature of little v. *Linguistic Inquiry* 38:197–218.
- Frampton, John and Gutmann, Sam. 2002. Crash-proof syntax. In Samuel David Epstein and T. Daniel Seely, eds., *Derivation and Explanation in the Minimalist Program*, 90–105, Malden, Massachusetts: Blackwell Publishing.

- Grimshaw, Jane. 1991. Extended projections, ms, Brandeis University.
- Guimaraes, Maximiliano. 2000. In defense of vacuous projections in bare phrase structure. *University of Maryland Working Papers in Linguistics* 9:90–115.
- Hendrick, Randall. 2007. Syntactic labels and their derivations. In Simin Karimi, Vida Samiiian, and Wendy K. Wilkins, eds., *Phrasal and Clausal Architecture: Syntactic derivation and interpretation.*, 93–107, Amsterdam: John Benjamins.
- Kayne, Richard. 1994. *The Antisymmetry of Syntax*. Cambridge, MA: MIT Press.
- Kayne, Richard. 2010. Antisymmetry and the lexicon. In Anna Maria Di Sciullo and Cedric Boeckx, eds., *The Biolinguistic Enterprise*, chapter 14, 329–353, Oxford, UK: Oxford University Press.
- Kratzer, Angelika. 1996. Severing the external argument from its verb. In Johan Rooryck and Laurie Zaring, eds., *Phrase Structure and the Lexicon*, 109–138, Dordrecht: Kluwer.
- Marantz, Alec. 1997. No escape from syntax: Don't try morphological analysis in the privacy of your own lexicon. In Alexis Dimitriadis, Laura Siegel, Clarissa Surek-Clark, and Alexander Williams, eds., *University of Pennsylvania Working Papers in Linguistics vol. 4.2*, 201–225, University of Pennsylvania.
- Marantz, Alex. 2006. Phases and words, ms NYU.
- Moro, Andrea. 2000. *Dynamic Antisymmetry*. Cambridge, MA.: MIT Press.
- Nilsen, Øystein. 2003. Eliminating positions. Ph.D. thesis, LOT, Utrecht Institute of Linguistics OTS.
- Perlmutter, David. 1978. Impersonal passives and the unaccusative hypothesis. In *Proceedings of the Fifth Meeting of the Berkley Linguistics Society*, UC Berkley.
- Pylkkänen, Liina. 2008. *Introducing Arguments*. Cambridge, Massachusetts: MIT Press.
- Radford, Andrew. 2000. NP shells. *Essex Research Reports in Linguistics* 2–20.
- Ramchand, Gillian. 2008. *Verb Meaning and the Lexicon: A First Phase Syntax*. Cambridge: Cambridge University Press.
- Rizzi, Luigi. 1997. The fine structure of the left periphery. In Liliane Haegeman, ed., *Elements of Grammar: Handbook of Generative Syntax*, 281–337, Dordrecht: Kluwer.
- Ross, John Robert. 1972. Act. In Donald Davidson and Gilbert Harman, eds., *Semantics of Natural Language*, 70–126, Dordrecht ; Boston: D. Reidel Publishing Company.
- 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. 2001. Move dissolves into merge. Ph.D. thesis, University of Geneva.
- Svenonius, Peter. 2004. On the edge. In David Adger, Cécile de Cat, and George Tsoulas, eds., *Peripheries: Syntactic Edges and their Effects*, 259–287, Dordrecht, The Netherlands: Kluwer Academic Publishers.

- Travis, Lisa. 2000. The L-syntax/S-syntax boundary: Evidence from Austronesian. In Ileana Paul, Vivianne Phillips, and Lisa Travis, eds., *Formal Issues in Austronesian Linguistics*, 167–194, Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Williams, Edwin. 2003. *Representation Theory*. Cambridge, Massachusetts: MIT Press.