Minimal C-structure: Rethinking Projection in Phrase Structure

Joseph Lovestrand
University of Oxford

John J. Lowe
University of Oxford

Proceedings of the LFG'17 Conference

University of Konstanz

Miriam Butt, Tracy Holloway King (Editors)

2017

CSLI Publications

pages 285–305

http://csli-publications.stanford.edu/LFG/2017

Keywords: constituent-structure, phrase structure, projecting nodes, non-branching nodes, XLE

Lovestrand, Joseph, & Lowe, John J. (2017). Minimal C-structure: Rethinking Projection in Phrase Structure. In Butt, Miriam, & King, Tracy Holloway (Eds.): *Proceedings of the LFG'17 Conference, University of Konstanz* (pp. 285–305). Stanford, CA: CSLI Publications.

Abstract

This paper addresses the formal properties of constituent structure (c-structure). We demonstrate inadequacies in the formalization of traditional X′ theory by Bresnan (2001) and Bresnan et al. (2016), and in the alternative proposal of Marcotte (2014). We propose "minimal c-structure" as a new approach to phrase structure within Lexical-Functional Grammar, which almost entirely eliminates non-branching nodes, and neatly captures the distinction between projecting and non-projecting words. Our proposal is fully formalized, and has been successfully tested by an XLE implementation.

1 Introduction

In Lexical-Functional Grammar (LFG), constraints on surface phrasal constituency are expressed in terms of phrase structure rules and represented by tree diagrams: "A commonly used representation of c-structure is the context-free phrase structure tree, defined by context-free phrase structure rules augmented by regular expressions" (Bresnan 2001: 44). The primitive elements in c-structure are nodes, labelled according to syntactic category and projection level. Our concern in this paper is the representation of projection level.[1] In this paper we show that existing formalizations of phrase structure in LFG are both inadequate and license superfluous structure. We propose a new formalization of phrase structure which eliminates redundancy while also capturing the full variety of phrase structures assumed in LFG.

Section 2 summarizes the formalization of X′-theoretic phrase structure proposed in Bresnan (2001), unmodified in Bresnan et al. (2016). Section 3 reviews a recent proposal by Marcotte (2014) to remove the notion of levels of structure from the formal properties of c-structure. Section 4 proposes a new formalization of phrase structure. Section 5 discusses further implications of our approach. A partial grammar for English based on our XLE (Crouch et al. 2011) implementation is provided in the appendix.

2 Current assumptions in LFG

We take Bresnan (2001; unmodified in Bresnan et al. 2016) as representative of standard assumptions regarding the formal properties of c-structure. Bresnan (2001: 100) describes the formal properties of c-structure nodes: "Formally, X′ categories

[†]Thanks to Mary Dalrymple and the University of Oxford Syntax Working Group (June 8, 2016) for feedback on these ideas in their early stages, to attendees of SE-LFG23 (13 May 2017), in particular Adam Przepiórkowski, for further constructive criticism, and also to the audience at LFG17, Konstanz (25 July 2017), in particular Ron Kaplan, Doug Arnold, John Payne and Peter Sells.

[1] Syntactic category labels, such as N(oun), V(erb), P(reposition), D(eterminer), etc., can be treated as atomic or can be decomposed into features (Bresnan 2001, Dalrymple 2017); syntactic categories are not the focus of this paper, and we will assume an atomic approach for simplicity.

can be analyzed as triples consisting of a categorical feature matrix, a level of structure, and a third, privative feature F, which flags a category as 'function' (F) or unspecified as to function (lexical)." The "level of structure" feature, which we call BAR following Andrews & Manning (1999), has three values: 0, 1, 2. These digits each correspond to a level of structure which is represented notationally using the traditional X-bar symbols: X^0 for [BAR 0], X' for [BAR 1], and XP for [BAR 2]. The use of integers in this context implies that, in an endocentric projection, a mother must have a BAR value higher than its daughter.[2] The question of dominance is not discussed formally by Bresnan, but the familiar templatic description of X-bar principles (1) makes it clear that some additional mechanism is intended to enforce the dominance sequence.

(1) a. **Specifier phrase structure rule**
$XP \rightarrow X'$, YP

b. **Complement phrase structure rule**
$X' \rightarrow X^0$, ZP

Marcotte (2014) criticizes this lack of formalization of dominance: "This dominance sequence is implied by the use of integers in the feature nomenclature, but does not constitute a formal requirement." That is, we intuitively know that 2 is above 1, but this does not come for free in the formal system.

The model put forward by Bresnan (2001) seems straightforward and intuitive, but it does not capture the diversity of structures used in descriptive analyses, and results in the postulation of unnecessary (and undesirable) structure, which can only be eliminated by the postulation of additional complications to the system. In the following subsections we discuss the main shortcomings of Bresnan's proposal.

2.1 Non-projecting words

Bresnan et al. (2016) differ in their theory of c-structure from Bresnan (2001) by incorporating the concept of non-projecting categories, as developed primarily by Toivonen (2003). Non-projecting categories differ from projecting categories in that they do not project any levels of structure, but are similar in being (potentially) preterminal nodes. They are technically not endocentric, and do not license any complement or specifier positions. In the current conception of non-projecting words, they only interact with the phrase structure through adjunction. Toivonen (2003) limits adjunction of non-projecting words to projecting heads (X^0). However, others have proposed allowing such adjunction to another non-projecting word or to a maximal phrase level (XP) (Spencer 2005, Arnold & Sadler 2013).

The close relation between non-projecting categories and zero-level projecting categories is evidenced by the notation: Toivonen (2003) proposes that X^0 be

[2] The formal properties of exocentric structures are not discussed by Bresnan (2001). We provide a formal analysis of exocentric structures in §5.2.

used to represent a preterminal (i.e. BAR 0) projecting node, $\hat{X}$ to represent a non-projecting node, and X as a cover for all zero level categories. Bresnan et al. (2016) incorporate the concept and notation of non-projecting categories into the theory of c-structure, but do not revise the formal analysis of c-structure nodes to account for this new node type. What is the BAR value of a non-projecting node? Since non-projecting nodes are preterminal, they might be considered to have a BAR value of 0 (as implied by their grouping with X^0 under the cover X). But if the BAR of a non-projecting node is 0, how is it distinguished formally from a projecting head? Bresnan et al. (2016) provide no formal account.

2.2 One-level projections

Some versions of X-bar theory assume that all categories project the same number of levels of structure. Kornai & Pullum (1990) call this “Uniformity”. Unless otherwise stated, it is normally assumed that all categories have two levels of projection.[3] In practice, Uniformity is not part of the LFG version of X-bar theory. For example, Bresnan (2001: 127) mentions in passing that the “specifier of IP is taken to be null (omitted) in Welsh, as a parametric choice.” The structure given is represented in (2). In this structure, a head X^0 is immediately dominated by a maximal projection node, XP.[4]

(2) Single-level endocentric structure (Bresnan 2001: 127):

IP

I^0 S

This “parametric choice” of allowing one or two levels of structure in an endocentric projection has not been formalized. The mother node of a one-level endocentric structure must share some property with the mother node of a two-level endocentric structure; this is represented, but not formalized, by the ‘-P’ element of the node label. But in formal terms, what is the BAR of the XP in a one-level structure? If its BAR is 1, it would be formally impossible to select the class of maximal projections. Presumably, since it is represented by XP, its BAR is 2. But this would have interesting consequences in regard to the question of dominance sequence discussed above. In this system, a node whose BAR is 0 can apparently be immediately dominated by a node whose BAR is 2. So the dominance sequence implied by the integer values of BAR does not have to be sequential.[5] Just as there

[3]The assumption that all syntactic heads must project two levels of structure appears to have become the X-bar standard assumption with the publication of Chomsky (1986). There were a variety of earlier proposals, e.g., three levels (Jackendoff 1977) or one level for lexical categories and two levels for functional categories (Fukui 1986). See Muysken (1982) for an overview.

[4]In other places, Bresnan allows similar structures as an effect of Economy of Expression (§2.3). The case under discussion here involves a structure that will never have a mid-level node that could be subject to the constraints of Economy of Expression.

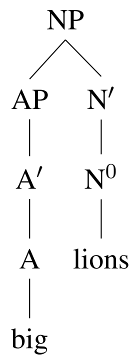
[5]Kornai & Pullum (1990: 28) note that many linguists have been inconsistent on this point violating what they call “Succession”.

is a need for formal constraints on the dominance sequence (such that [BAR 1] cannot dominate [BAR 2]), there is also a need for formal constraints that allow the non-sequential dominance sequence (such that [BAR 2] can dominate [BAR 0]). No such constraints are proposed or formalized by Bresnan (2001).

2.3 Non-branching structures

In the standard approach to X′ theory, it is assumed that if a head can project structure, the spine of that structure is present even when non-head elements of the structure are not present. Kornai & Pullum (1990) call this assumption "Maximality". In LFG, Maximality is enforced by phrase structure rules that only select maximal phrases (XP) as non-head daughters. The result of Maximality is that when a head which can potentially co-occur with a complement and specifier appears alone in a constituent, its structure must be analyzed with at least two non-branching nodes, like the AP in (3). Note that the NP in (3) also has one non-branching node since there is no complement phrase.

(3) Non-branching nodes (Bresnan et al. 2016: 90):



Given the principled objection to empty categories in LFG,[6] Maximality requires the assumption that all complement and specifier positions can be optional. Dalrymple et al. (2015) call this type of optionality "Daughter Omission". Bresnan generalizes Daughter Omission over all phrase structure rules. Dalrymple et al. (2015: 386-388) point out that it is problematic to generalize Daughter Omission since there are cases when a daughter node is best analyzed as obligatory. In addition, Kornai & Pullum (1990: 31, 46) state that there is essentially no analytical value in assuming Maximality if those levels can have non-branching (or empty) nodes. The assumption of Maximality is satisfied with no corresponding empirically-observable content. In other words, it is a non-falsifiable assumption.

The non-branching nodes required by Maximality are in tension with a general principle of Economy of Expression. In the context of c-structure, Economy of Expression is a principle which states that, all else being equal, the smallest licit c-structure is the optimal one. This tension is noted by Jackendoff (1977: 36)

[6]Except perhaps as a "last resort" (Bresnan et al. 2016: 205).

who writes that Maximality comes "at the expense of some otherwise superfluous structure." The intuition is that the c-structure in (3) would be better if it did not have non-branching nodes that do not contain any unique grammatical information other than their BAR.

Bresnan (2001: 91) develops this intuition into what Dalrymple et al. (2015) call "X′ Elision". X′ Elision is a process of pruning all unnecessary nodes from a well-formed c-structure so that it is as small as possible. X′ Elision is therefore a complication of the grammar: it is a derivational process that, given an initial production of c-structures which are well-formed according to the phrase structure rules, then modifies those c-structures in ways that cannot be derived via the phrase structure rules. Given this complication, the position of Dalrymple et al. (2015) is that a general principle of Economy of Expression is not satisfactory motivation for incorporating X′ Elision in the grammar. Our proposed theory of c-structure improves on both of these positions by rejecting Maximality. It allows phrase structure rules to produce trees that satisfy the Economy of Expression principle without requiring a secondary operation like X′ Elision. It also removes the need to assume a generalized version of Daughter Omission. (See §6 for discussion of optionality in our approach.)

2.4 Conclusion

Some of the issues in the proposals of Bresnan (2001) could technically be resolved without fundamental changes to the system, e.g. by adding a feature to indicate whether a node is projecting or not, and adding a metaconstraint to allow non-sequential dominance sequences. However, such a solution would fail to capture the generalizations and subtleties that will be represented in the formal analysis proposed in §4. For example, a feature indicating non-projecting status would also occur on all nodes of level 2, without any explanation for why non-projecting words and XP nodes form this class. In addition to providing an elegant account of the issues in §2.1 and §2.2, the formal analysis of §4 also avoids the tension between Maximality and Economy of Expression highlighted in §2.3.

3 Marcotte (2014)

Marcotte (2014) reviews three formal approaches to c-structure and syntactic categories with a particular interest in the passing of information between nodes, such as the shared lexical (or functional) category features shared by a head and its mother. That particular problem is not addressed in this paper, but Marcotte's proposal is of interest because it also revises Bresnan's formal analysis of c-structure nodes. Marcotte attempts to simplify the formal properties of the system by removing the BAR feature and thus "allowing a reformulation of its insights with a reduced number of theoretical primitives..." (Marcotte 2014: 426). However, this simplification results in a system which cannot account for some basic syntactic structures.

Marcotte's proposal is to remove the BAR feature, and to instead define the relationships between nodes in terms of dominance relations and shared category features, which he defines in terms of "x-structure". There are three basic definitions that define types of nodes in c-structure:

(4) a. PROJECTING NODE: A node projects iff its x-structure is identical with its mother's x-structure.
$Proj(*) \iff \chi(*) = \chi(\mathcal{M}(*))$

b. MAXIMAL PROJECTION: A node is a maximal projection iff it is not a projecting node.
$Max(*) \iff \neg Proj(*)$

c. TERMINAL: A node is a terminal iff no node has it as a mother.[7]
$Term(*) \iff \neg\exists n.\mathcal{M}(n) = *$

In (4), χ is a function from nodes to x-structures. $\mathcal{M}$ is the mother function from a node to its mother; $*$ represents the current node, and n represents any other node. In this system, there are four types of nodes, roughly equivalent to X^0, X', XP and $\hat{X}$. A projecting head ($\approx X^0$) is a node that meets the definitions of PROJECTING NODE (it has the same category as its mother) and TERMINAL (it is not the mother of any node).[8] A maximal projection ($\approx$ XP) meets the definition of MAXIMAL PROJECTION (it does not have a mother with identical features), and is not TERMINAL. Intermediate nodes ($\approx X'$) meet the definition of PROJECTING NODE, but not of TERMINAL. A non-projecting node ($\approx \hat{X}$) is both a MAXIMAL PROJECTION and a TERMINAL.[9]

Marcotte applies his approach to c-structure to the structure-function mapping principles, defining default positions for subjects, objects, heads, etc. For example (slightly simplified):

(5) **Marcotte (2014) "Endocentric c- to f-structure mappings"**

a. A projecting node shares the f-structure of its mother:
$Proj(*) \Longrightarrow \uparrow = \downarrow$

b. A SUBJ is a DP daughter of IP:
$$\begin{array}{ll} Max(*) & Max(\mathcal{M}(*)) \\ \chi = \mathrm{D} & \chi(\mathcal{M}(*)) = \mathrm{I} \end{array} \Longrightarrow (\uparrow \text{SUBJ}) = \downarrow$$

[7] Presumably this is an accurate correction of a typographical error in the original which reads "A node is a terminal iff it [sic] node has it as a mother."

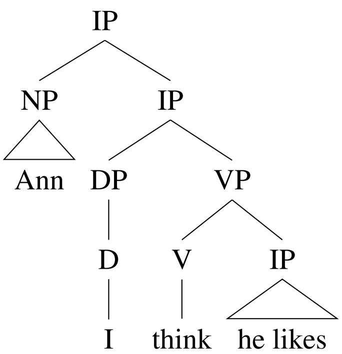
[8] Marcotte uses terminal to refer to what Bresnan calls a preterminal node. The lexical information at the bottom of the tree is not considered a node.

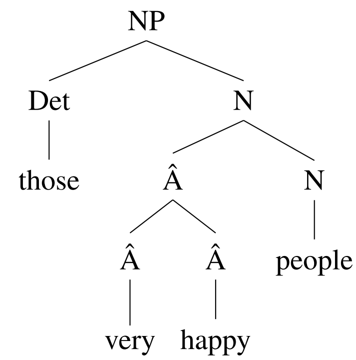
[9] This proposal is strikingly similar to that of Muysken (1982). Muyksen proposes eliminating bar levels and instead defines relationships in an endocentric projection in terms of two binary features: [±projection], [±maximal]. Muysken's [+maximal] is analogous to Marcotte's MAXIMAL PROJECTION, and [−maximal] parallels PROJECTING NODE. Muysken's [−projection] is analogous to TERMINAL, and [+projection] is similar to the non-Terminals. Muysken also discusses "non-projecting minor elements". He defines these in the same way as X^0: [−projection,+maximal].

Notably absent, however, is any definition of adjunction. In adjunction structures, the head and the mother have identical category features, just as in endocentric projection structures. The crucial difference between adjunction and projection is that in the former the head and its mother also share the same level of structure, while in the latter the mother has a higher level of structure. This distinction cannot be captured in Marcotte's (2014) proposal, because he has no equivalent to Bresnan's "level of structure" feature.

In addition, XP adjunction structures (as illustrated in (6)) pose a practical problem for Marcotte's definition of maximal phrases. In XP adjunction, a maximal phrase (XP) is the mother of another maximal phrase of the same category features. Such a structure is not possible in Marcotte's proposed system because, by definition, if a node's mother has the same category features, that node cannot be a MAXIMAL PROJECTION. It is a PROJECTING NODE.

The adjunction of non-projecting categories also poses difficulties for Marcotte's approach. For example, Arnold & Sadler (2013) propose that a non-projecting node can adjoin to another non-projecting node, a possibility which permits an insightful account of prenominal modification in English (7). However, in Marcotte's system, the modifiers *very* and *happy* would not count as non-projecting words. For Marcotte, a non-projecting word is both TERMINAL and MAXIMAL. A MAXIMAL PROJECTION is defined as a node that does not have the same category features as its mother, but in (7) both modifiers have the same category as their mother, so, by definition, they are PROJECTING, not MAXIMAL.[10] In addition, the mother of the two modifiers is a MAXIMAL PROJECTION in Marcotte's terms, but it is not TERMINAL since it has a daughter node that shares the same features. Thus this node does not meet the definition of a non-projecting node either.

(6)


(7)


These are just two examples of a number of analytical roadblocks that arise when applying Marcotte (2014)'s system to adjunction structures. It might be possible to address some of these issues by modifying his system, but it is hard to see how this could be done without compromising the formal elegance that motivates

[10] This example would not be a problem if, following e.g. Payne et al. (2010), adjectives and adverbs were treated as part of separate c-structure categories. But the point remains valid, e.g. in the phrase *those really very happy people* the same problem would apply to the relationship between *really* and *very*.

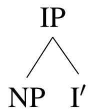
Marcotte's proposal. At the current time, it seems doubtful that any such attempt to entirely remove reference to levels of structure from the theory of projection is likely to be successful.

4 Minimal c-structure

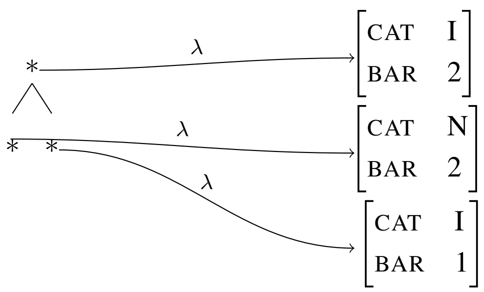
The facts and generalizations missed by both Bresnan's and Marcotte's approaches can be captured in relatively simple terms. Rather than abandoning reference to level of structure, as Marcotte (2014) does, we propose to split the "level of structure" feature of Bresnan (2001) into two features, the interaction of which permits us to provide an insightful and efficient model of c-structure. We call this approach "minimal c-structure" since it generates c-structures with the minimal number of levels of structure required to model constituency.

Following Kaplan (1989), syntactic category information and projection level are not directly encoded in c-structure, but are projected from c-structure nodes via a projection λ. That is, the traditional representation of Bresnan (2001) in (8) must be understood as a shorthand for something like (9).

(8)

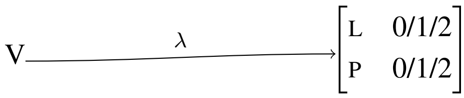


(9)



We refer to the projection of λ as the l-structure. Instead of one BAR feature, we assume two features: LEVEL or L, which represents the projection level of a particular node in a particular structure; and PROJECTION or P, which represents the total number of levels of structure in the projection of the node. The values of L and P are integers, i.e. 0, 1, 2 etc. The value 2 is a sufficient maximum for English, but higher values may be required for other languages. We distinguish non-projecting words from projecting words by defining the former as having no P feature.[11] We are not concerned with encoding category information via λ, only projection level information (see fn. 1). We therefore assume the following type of structures:

(10)



[11]Unlike previous works on non-projecting words, we do not restrict non-projecting words to adjunction structures. A node with no P value can either be adjoined or selected as the specifier complement of a phrase structure rule. In this way our analysis provides a systematic account of the c-structure of what are often called "minor categories".

Although we understand the features L and P as features within the 'l-structure' projected from a node, for ease of representation we utilize an alternative notation whereby L and P values are shown as superscripts on c-structure nodes, separated by a slash. For example, $\mathrm{V}^{0/1}$ refers to a node of category V whose l-structure includes the features <L,0> and <P,1>.

4.1 Annotated phrase structure rules and templates

The L and P values for any node are determined by a combination of constraints which appear as annotations on c-structure nodes or in lexical entries. We assume that all words define their mother (preterminal node) as L = 0. P values are not (normally) determined lexically: it is not the case that a projecting head always projects the same number of levels of structure. Phrase structure rules can, but do not need to refer to L and P values directly. They can also refer to a relationship between L and P, for example by requiring that the L and P values for a particular node are identical (our definition of a maximal projection). They can also specify relations between mother and daughter nodes, for example that the P value for a particular node is identical to the mother node's P value (i.e. the head of an endocentric projection).

There are a fixed set of constraints required to model projection in c-structure, and for convenience we define these as templates, called as appropriate in the c-structure rules:

(11) Basic templates:

a.	LP	$\equiv$	$(*_\lambda\ \mathrm{L}) = (*_\lambda\ \mathrm{P})$	g.	LO	$\equiv$	$(*_\lambda\ \mathrm{L}) = 0$
b.	LPM	$\equiv$	$(\hat{*}_\lambda\ \mathrm{L}) = (\hat{*}_\lambda\ \mathrm{P})$	h.	LOM	$\equiv$	$(\hat{*}_\lambda\ \mathrm{L}) = 0$
c.	LUD	$\equiv$	$(\hat{*}_\lambda\ \mathrm{L}) = (*_\lambda\ \mathrm{L})$	i.	PXM	$\equiv$	$(\hat{*}_\lambda\ \mathrm{P})$
d.	PUD	$\equiv$	$(\hat{*}_\lambda\ \mathrm{P}) = (*_\lambda\ \mathrm{P})$	j.	PNX	$\equiv$	$\neg(*_\lambda\ \mathrm{P})$
e.	LIM	$\equiv$	$(\hat{*}_\lambda\ \mathrm{L}) = 1$	k.	PNXM	$\equiv$	$\neg(\hat{*}_\lambda\ \mathrm{P})$

f. LDOWN $\equiv \{(*_\lambda\ \mathrm{L}) = 0 \wedge (\hat{*}_\lambda\ \mathrm{L}) = 1 \mid (*_\lambda\ \mathrm{L}) = 1 \wedge (\hat{*}_\lambda\ \mathrm{L}) = 2\ \}$

The constraint LP requires that the node in question have identical values for L and P. LPM states the same constraint of the mother of the current node. LUD and PUD ensure that a daughter and its mother have identical L or P values respectively. LIM specifies that the L value of the node's mother has the value 1. LDOWN models the change in projection level through an endocentric structure, ensuring that the L value of the node in question is one lower than the L value of its mother. LO and LOM specify the value 0 for the feature L respectively of a node or its mother. PXM is an existential constraint, requiring the mother of a node to have a P feature. PNX and PNXM are negative existential constraints, requiring that a node or its mother respectively do not have a P feature. These last three constraints are relevant to the analysis of non-projecting words, since we define non-projecting words as lacking P features.

The definition of LDOWN (11f) requires a brief comment. Intuitively, we treat L and P values as integers, with the expected ordering relation, such that the principle to be captured by LDOWN is the following:

(12) LDOWN $\equiv$ ($*_\lambda$ L) = ($\hat{*}_\lambda$ L) -1

However, l-structures are feature structures, and values such as 0, 1 and 2 in a feature structure are symbols, not natural numbers. There are various possibilities for effecting ordering relations between symbols; for the purposes of our XLE implementation, however, we adopted the simplest approach, replicating the arithmetic in (12) by stipulation, assuming that there are only three possible values for L.[12]

We also intuitively want a system where the L value is never "greater than" the P value such as: $\forall *_\lambda$, P $\geq$ L. Again, since our values are not natural numbers, this is not possible. Nor is it necessary: given the constraints in (11) and the assumption that the L value of all words is 0, our system will never create a c-structure that violates this intuitive general constraint.

Particular combinations of the constraints in (11) are associated with standard phrase structure notions, such as head or specifier positions. We therefore define further templates which call relevant combinations of templates from (11):

(13) Complex templates:

a.	HEADX	$\equiv$	@LDOWN $\wedge$ @PUD
b.	HEADA	$\equiv$	@LUD $\wedge$ @PUD
c.	EXT	$\equiv$	@LPM $\wedge$ @LP
d.	INT	$\equiv$	@LIM $\wedge$ @LP
e.	NONPRJ	$\equiv$	@LO $\wedge$ @PNX
f.	NONPRJM	$\equiv$	@LOM $\wedge$ @PNXM
g.	PRJM	$\equiv$	@LOM $\wedge$ @PXM
h.	NONPADJ	$\equiv$	@LOM $\wedge$ @NONPRJ

HEADX is the constraint set required for heads in traditional X′-theoretic structures, i.e. in specifier and complement structures: such a head has an L value one lower than that of its mother, and a P value identical to that of its mother. HEADA applies to heads in adjunction and coordination structures: such heads have the same L and P values as their mothers.[13] The constraints introduced by EXT apply to specifiers and adjuncts (unless nonprojecting): they are maximal projections (L=P) and are daughters of maximal projections (that this is true for adjuncts is discussed in §5.1). INT applies to complement positions (unless nonprojecting): they are maximal projections which are daughters of a node with an L value 1 (and therefore sisters of a head with <L,0>). NONPRJ requires that a node be nonprojecting: its L value is 0 and it has no P feature. NONPRJM states the same requirement

[12]It is of course unproblematic to license higher values for L by simply adding additional disjunctions to the definition of LDOWN. Given our formalism, licensing higher values for L and P does not mean those values necessarily occur, since values are determined bottom-up.

[13]The definition of HEADA in (13b) works unproblematically in XLE, but would need to be complicated to work within the formal description of LFG given by Kaplan & Bresnan (1982); this is discussed further in §5.2.

of the mother node; this is used in the lexical entries of nonprojecting words. PRJM states that the mother node has an L value of 0 and has a P feature; this is used in the lexical entries of projecting words. NONPADJ applies to nonprojecting adjuncts, which we assume can adjoin, in English, only to nodes with an L value of 0. We can therefore represent the major types of phrase structure in the following way:

(14) a. Specifier rule: $\mathrm{X} \rightarrow \underset{@\text{EXT}}{\mathrm{Y}} \quad \underset{@\text{HEADX}}{\mathrm{X}}$

b. Complement rule: $\mathrm{X} \rightarrow \underset{@\text{HEADX}}{\mathrm{X}} \quad \underset{@\text{INT}}{\mathrm{Y}}$

c. 'XP' adjunction: $\mathrm{X} \rightarrow \underset{@\text{HEADA}}{\mathrm{X}} \quad \underset{@\text{EXT}}{\mathrm{Y}}$

d. Conjunction: $\mathrm{X} \rightarrow \underset{@\text{HEADA}}{\mathrm{X}} \quad \underset{@\text{NONPRJ}}{\mathrm{Cnj}} \quad \underset{@\text{HEADA}}{\mathrm{X}}$

e. Non-proj. adj.: $\mathrm{X} \rightarrow \underset{@\text{NONPADJ}}{\mathrm{Y}} \quad \underset{@\text{HEADA}}{\mathrm{X}}$

These complex templates also provide a mechanism for retaining the structure-function association principles of Bresnan (2001). Nodes annotated with @HEADX and @HEADA are functional heads (↑=↓). Complements are annotated with @INT and can be associated with the appropriate grammatical functions. Specifier and adjunct nodes are annotated with @EXT, and can e.g. be associated with the grammaticalized discourse functions or adjunct functions.

4.2 Illustration

As an illustration of minimal c-structure, we give in (15) the necessary phrase structure rules to derive the English sentence *The small dog eats biscuits*. The lexical entries supply category information and the template call @PRJM, for projecting words, or @NONPRJM, for nonprojecting words (see the appendix). The result is the phrase structure in (16).

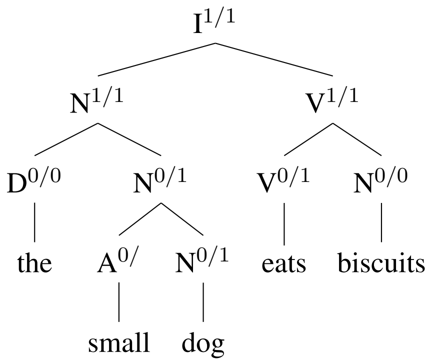
(15) a. $\mathrm{I} \rightarrow \underset{\substack{(\uparrow \text{SUBJ}) = \downarrow \\ @\text{EXT}}}{\mathrm{N}} \left\{ \underset{\substack{\uparrow=\downarrow \\ @\text{HEADX}}}{\mathrm{I}} \;\middle|\; \underset{\substack{\uparrow=\downarrow \\ @\text{INT}}}{\mathrm{V}} \right\}$

b. $\mathrm{V} \rightarrow \underset{\substack{\uparrow=\downarrow \\ @\text{HEADX}}}{\mathrm{V}} \quad \underset{\substack{(\uparrow\text{OBJ})=\downarrow \\ @\text{INT}}}{\mathrm{N}}$

c. $\mathrm{N} \rightarrow \underset{\substack{\uparrow=\downarrow \\ @\text{EXT}}}{\mathrm{D}} \quad \underset{\substack{\uparrow=\downarrow \\ @\text{HEADX}}}{\mathrm{N}}$

$$\text{d.}\quad \text{N} \;\rightarrow\; \begin{array}{cc} \text{A} & \text{N} \\ \downarrow\in(\uparrow\text{ADJ}) & \uparrow=\downarrow \\ @\text{NONPADJ} & @\text{HEADA} \end{array}$$

(16)



This tree has only nine nodes, and no non-branching nodes, compared with a standard X′-theoretic tree following the pattern in (3), which would have at least 14 nodes including at least 5 non-branching nodes.

The LEVEL value is determined from the bottom up, with all words specifying L=0 of their preterminal node. The PROJECTION value is determined by the number of projection levels in the phrase. A projecting head is permitted to have two, one or even zero levels of structure in any given c-structure. Here, *dog* is immediately dominated by a node $N^{0/1}$, while biscuits is immediately dominated by a node $N^{0/0}$: the difference is that *dog* heads a complex phrase which includes a specifier, while *biscuits* is the only word in its phrase. There are no superfluous levels of structure: the single node dominating *biscuits* is both the maximal projection and the preterminal node of this phrase.

Non-projecting words are distinguished from projecting words by not having a PROJECTION feature. The PROJECTION value for all other nodes is determined by annotations on phrase structure rules. The ‘dominance constraint’ (e.g., that a node with LEVEL value 1 cannot be the mother of a head with a value of 2) is formally constrained in the phrase structure rules.

The traditional notion of a ‘maximal projection’ remains, defined as L=P; the traditional notion of a (pre)terminal node remains, as L=0. As with *biscuits* in (16), these notions are not mutually exclusive. Note that despite our radically reduced structure, our proposal also retains the notions of specifier, complement and adjunct, licensing the standard structure-function mapping principles.

Interestingly, specifier, complement and adjunct structures are not necessarily mutually exclusive: the top node in (16) has both a specifier daughter and a complement daughter. Only maximal projections (L=P) can have specifier daughters; only nodes with the feature L = 1 can have complement daughters; the node $I^{1/1}$ satisfies both constraints simultaneously.[14]

[14]Note that such possibilities have to be specifically licensed, as in (15a): they do not fall out directly from the phrase structure templates in (14).

4.3 Nonprojecting categories

A key requirement of our model is that it appropriately capture the difference between projecting and non-projecting categories, something which both Bresnan (2001) and Marcotte (2014) fail to achieve. We have modelled this with respect to adjectives, assuming that prenominal adjectives in English are non-projecting, but that in other positions adjectives can project phrases. We assume that adjectives which can appear in both prenominal and other positions have both projecting and non-projecting variants, but that adjectives such as *former* are non-projecting only (since they can only appear prenominally) and that adjectives such as *asleep* are projecting only (since they cannot appear prenominally). Our model correctly captures the grammaticality/ungrammaticality of the following examples:

(17) a. The small dog eats biscuits.
b. The dog is small.
c. The former president eats biscuits.
d. *The president is former.
e. *The asleep dog eats biscuits.
f. The dog is asleep.

Note that in our model, the inability of nonprojecting words to take specifiers, phrasal adjuncts and complements does not have to be stipulated, but falls out of the templates given in (13). Specifiers and phrasal adjuncts require their mother to have identical L and P values, while complements require their mother to have L=1; nonprojecting nodes are necessarily L=0 and lack a P value, so cannot take daughters with such requirements.

5 Two further implications of minimal c-structure

In addition to providing an improved formalization of issues that have been dealt with in previous work, minimal c-structure also provides a natural explanation of why X′ adjunction is not possible, and allows a formalization of the exocentric category S.

5.1 X′ adjunction

Adjunction is a perennial problem for X′ theoretic approaches to phrase structure. We can distinguish three different types of adjunction in traditional X′ notation: adjunction to XP, adjunction to X′, and adjunction to X^0. We assume adjunction to X^0 is restricted to non-projecting categories.

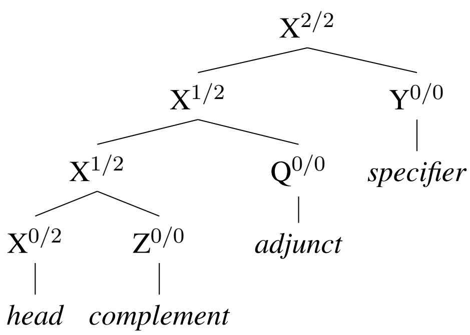
Let us see how to translate this three-way distinction into our model. XP adjunction means adjunction to nodes with the L/P values 2/2, 1/1 or 0/0, i.e. L=P.[15]

[15] And, of course, 3/3 etc., if more than two levels of projection are admitted.

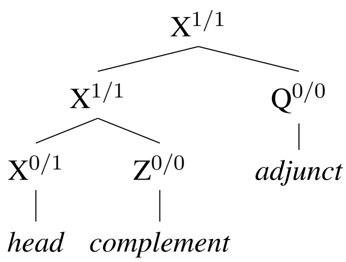
Adjunction of non-projecting categories to zero-level or non-projecting nodes is adjunction to 0/0, 0/1, 0/2 or 0/, i.e. L=0. Both of these are unproblematic, and are formalized in (14) above making use of independently required templates.

X′ adjunction, however, is impossible to formalize in minimal c-structure. Intuitively, X′ adjunction prevents an adjunct from occurring closer to the head than its complement, or further from the head than its specifier. It is impossible to select a set of L and P values which uniquely identifies such a position. In the tree in (18), the adjunct *Q* is adjoined in the equivalent of an X′ position; the node it is adjoined to has the projection features 1/2. However, this is not the only possibility. In (19), an adjunct *Q* is adjoined above the complement, but no specifier is present. Since the L and P values are not fixed, in this structure the adjunct is adjoined to a node with the features 1/1. The tree in (20) is also a type of X′ adjunction, only in this case there is no complement present; here the adjunct attaches to a node with the features 0/1. The problem is that adjunction to 1/1, which is required for (19), would be XP adjunction in (20), since the adjunct would appear higher than the specifier, and at the same time adjunction to 0/1, which is required for (20), would license adjunction between head and complement in (19). Thus it is not possible to define a set of L/P values which would capture the intuition of 'X′ adjunction'.

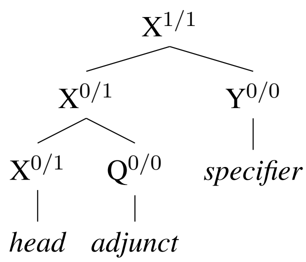
(18)



(19)



(20)



We take the impossibility of 'X′ adjunction' as a positive result, since a wealth of research in the last 30 years has shown that there is little or no evidence for processes which make specific reference to intermediate levels.[16] We therefore follow Toivonen (2003) in assuming the principle of "adjunction identity", that in adjunction "same adjoins to same".

[16]Early arguments in Travis (1984), see also Carnie (2000, 2010).

5.2 Exocentric ‘S’

Thus far we have discussed only endocentric projections. However, exocentric categories such as S are widely utilized in LFG. Our proposal not only admits exocentric categories, but even provides an insightful account of them. For illustrative purposes, consider a language similar to Welsh (Sadler 1997), with top-level clausal structure as in (2). In our system, S can be introduced as a complement daughter, just like any other complement:

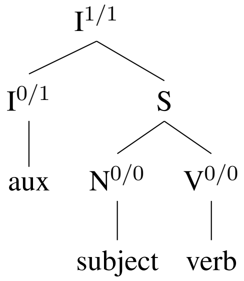
(21) $$\begin{array}{lllll} \mathrm{I} & \rightarrow & \mathrm{I} & \mathrm{S} \\ & & \uparrow{=}\downarrow & \uparrow{=}\downarrow \\ & & @\textsc{headx} & @\textsc{int} \end{array}$$

Since no daughter of S is the head, or a specifier, or a complement, or indeed an adjunct, none of the templates above apply to any of the daughter nodes. We can assume that daughters of S may be specified as necessarily projecting, for example, but the point is no daughter will make any specification about the L/P values of S.

(22) $$\begin{array}{llllll} \mathrm{S} & \rightarrow & \mathrm{N} & \mathrm{V} & \textit{or just} & \mathrm{X}^* \\ & & @\textsc{lp} & @\textsc{lp} & & @\textsc{lp} \end{array}$$

In XLE, the specification L=P is satisfied if $\neg$L $\wedge$ $\neg$P. This results in:

(23)



Thus as an exocentric category, S lacks L/P values. We take this to be the definition of an exocentric category in our system.[17] Since the features L and P are features to do with endocentric projection, this is intuitively satisfying.

This works in XLE, but given the definition of LP in (11a), it does not work in the theory. According to Kaplan & Bresnan (1982), a specification like L=P requires those features to exist, and if values for those features are not found, the derivation will fail. Under those assumptions, a practical solution is simply to redefine the template LP for languages which license exocentric structures as follows:[18]

(24) $\textsc{lp} \equiv \{ (*_\lambda\ \textsc{l}) = (*_\lambda\ \textsc{p}) \mid \neg(*_\lambda\ \textsc{l}) \wedge \neg(*_\lambda\ \textsc{p}) \}$

[17] Note that multiple exocentric categories are possible, distinguished by category label (cf. fn. 1).

[18] It is for the same reason that HEADA may have to be complicated, as discussed in fn. 13. Since HEADA calls PUD, and we want this to apply even if there is no P value, we would need: HEADA $\equiv$ @LUD $\wedge$ { @PUD | [@PNX $\wedge$ @PNXM]}.

6 Conclusion

Minimal c-structure is a more theoretically and formally precise approach to writing phrase structure rules in LFG. By writing l-structure templates below the nodes in phrase structure rules and the slash notation on nodes in c-structure trees, we can avoid the use of the outdated and formally ambiguous diacritics of previous versions of X-bar theory.

Further implementation of minimal c-structure will test the coverage of the templates that have been proposed in this paper. If certain templates turn out to be universal, while others are fairly rare or language-specific, this may give us more insight into the nature of c-structure cross-linguistically.

More study is still needed on the nature of optionality in minimal c-structure. Most optionality is trivial: if the complement of a verb is absent, for example, the phrase structure rule introducing it is simply not used; there is therefore no need for the complement to be optional in the rule itself. In a binary structure of a head and one sister, this is unproblematic. In a ternary (or larger) structure where each sister of the head can be omitted, it is still relatively simple to avoid a non-branching structure where a head has no sisters, by means of a disjunction (see fn. 19).

Optional heads and discontinuous constituents provide a further avenue for future investigation. One way to analyze optional heads is by means of disjunction, as in (15a) in §4.2. While this may work well in the case of extended projections, it cannot be used to account for the type of optional heads found in discontinuous constituents. For example, Snijders (2012) analyzes an adjective separated from the noun it modifies as the sole daughter of an N mother node. Here, the non-branching structure is necessary for indicating the category of the mother node, not for level of structure information. In such restricted cases, a non-branching node may be necessary, even under a minimal c-structure analysis.

References

Andrews, Avery D. & Christopher D. Manning. 1999. *Complex predicates and information spreading in LFG*. Stanford, CA: CSLI Publications.

Arnold, Doug & Louisa Sadler. 2013. Displaced dependent constructions. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG13 Conference*, 48–68. Stanford, CA: CSLI Publications.

Bresnan, Joan. 2001. *Lexical-functional syntax*. Oxford: Blackwell.

Bresnan, Joan, Ash Asudeh, Ida Toivonen & Stephen Wechsler. 2016. *Lexical-functional syntax*. Oxford: Wiley-Blackwell. Second edition.

Carnie, Andrew. 2000. On the definition of X^0 and XP. *Syntax* 3(2). 59–106.

Carnie, Andrew. 2010. *Constituent structure*. Oxford: Oxford University Press 2nd edn.

Chomsky, Noam. 1986. *Barriers*. Cambridge, MA: MIT Press.

Crouch, Dick, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. Maxwell III & Paula Newman. 2011. *XLE documentation*. Palo Alto Research Center, CA. http://www2.parc.com/isl/groups/nltt/xle/doc/xle_toc.html.

Dalrymple, Mary. 2017. Unlike phrase structure category coordination. To appear.

Dalrymple, Mary, Ronald M. Kaplan & Tracy Holloway King. 2015. Economy of expression as a principle of syntax. *Journal of Language Modelling* 3(2). 377–412.

Falk, Yehuda N. 2001. *Lexical-Functional Grammar: an introduction to parallel constraint-based syntax*. Stanford, CA: CSLI Publications.

Fukui, N. 1986. *A theory of category projection and its applications*: MIT dissertation.

Jackendoff, Ray. 1977. *$\bar{X}$ Syntax: A Study of Phrase Structure*. Cambridge, MA: MIT Press.

Kaplan, Ronald M. 1989. The Formal Architecture of Lexical-Functional Grammar. In Chu-Ren Huang & Keh-Jiann Chen (eds.), *ROCLING II: Proceedings of the Computational Linguistics Conference*, 3–18. Tapei: The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Kaplan, Ronald M. & Joan Bresnan. 1982. Lexical-Functional Grammar: A Formal System for Grammatical Representation. In Joan Bresnan (ed.), *The Mental Representation of Grammatical Relations*, 173–281. Cambridge, MA: MIT Press.

Kornai, András & Geoffrey K. Pullum. 1990. The X-bar theory of phrase structure. *Language* 66. 24–50.

Marcotte, Jean-Philippe. 2014. Syntactic categories in the correspondence architecture. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG14 Conference*, 408–428. Stanford, CA: CSLI Publications.

Muysken, Pieter. 1982. Parametrizing the notion "head". *Journal of Linguistic Research* 2(3). 57–75.

Payne, John, Rodney Huddleston & Geoffrey K. Pullum. 2010. The distribution and category status of adjectives and adverbs. *Word Structure* 3(1). 31–81.

Sadler, Louisa. 1997. Clitics and the Structure-Function Mapping. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG97 Conference*, Stanford, CA: CSLI Publications.

Snijders, Liselotte. 2012. Issues Concerning Constraints on Discontinuous NPs in Latin. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG12 Conference*, 565–581. Stanford, CA: CSLI Publications.

Spencer, Andrew. 2005. Case in Hindi. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG05 Conference*, 429–446. Stanford, CA: CSLI Publications.

Toivonen, Ida. 2003. *Non-projecting words: A case study of Swedish verbal particles*. Dordrecht: Kluwer.

Travis, Lisa deMena. 1984. *Parameters and effects of word order variation*: MIT dissertation.

Appendix

To establish the viability of our "minimal" approach to c-structure, we implemented a mini-grammar of English in XLE. The implementation makes use of the ability to define additional projections in XLE: the projection $*_\lambda$ we define as `l::*`, with $\hat{*}_\lambda$ defined as `l::M*`. The implementation covers most of the basic grammatical structures of English, including embedded clauses, wh-question formation, copular clauses, coordination, and all kinds of modification. It is loosely based on the English grammar in Falk (2001), but assumes up to two levels of projection for lexical categories. We give here a brief grammar for English based on our XLE implementation, and two additional c-structures, for illustration.

Rules:

$$\text{C} \rightarrow \begin{array}{cc} \text{N} & \text{C} \\ @\textsc{ext} & @\textsc{headx} \\ (\uparrow\textsc{udf}) = \downarrow & \uparrow=\downarrow \\ (\uparrow\textsc{udf}) = (\uparrow\textsc{udfpath}) & \end{array}$$

$$\text{C} \rightarrow \begin{array}{cc} \text{C} & \{\text{I} \mid \text{V}\} \\ @\textsc{headx} & @\textsc{int} \\ \uparrow=\downarrow & \uparrow=\downarrow \end{array}$$

$$\text{I} \rightarrow \begin{array}{cc} \{\text{N} \mid \text{C} \mid \text{P}\} & \{\text{I} \quad\mid\quad \text{V}\} \\ @\textsc{ext} & @\textsc{headx} \qquad @\textsc{int} \\ (\uparrow\textsc{subj}) = \downarrow & \uparrow=\downarrow \qquad\quad \uparrow=\downarrow \end{array}$$

$$\text{I} \rightarrow \begin{array}{cc} \text{I} & \text{V} \\ @\textsc{headx} & @\textsc{int} \\ \uparrow=\downarrow & \uparrow=\downarrow \end{array}$$

$$\text{V} \rightarrow \begin{array}{ccccc} \text{V} & \text{(N)}^{[19]} & \text{(N)} & \text{(P)} & \text{(C)} \\ @\textsc{headx} & @\textsc{int} & @\textsc{int} & @\textsc{int} & @\textsc{int} \\ \uparrow=\downarrow & (\uparrow\textsc{obj})=\downarrow & (\uparrow\textsc{obj2})=\downarrow & (\uparrow\textsc{obl})=\downarrow & (\uparrow\textsc{comp})=\downarrow \end{array}$$

$$\text{V} \rightarrow \begin{array}{cc} \text{V} & \{\text{N} \mid \text{A}\} \\ @\textsc{headx} & @\textsc{int} \\ \uparrow=\downarrow & (\uparrow\textsc{predlink})=\downarrow \end{array}$$

$$\text{V} \rightarrow \begin{array}{cc} \text{V} & \{\text{Adv} \mid \text{P}\} \\ @\textsc{heada} & @\textsc{ext} \\ \uparrow=\downarrow & \downarrow \in (\uparrow\textsc{adj}) \end{array}$$

$$\text{V} \rightarrow \begin{array}{cc} \text{Adv} & \text{V} \\ @\textsc{ext} & @\textsc{heada} \\ \downarrow \in (\uparrow\textsc{adj}) & \uparrow=\downarrow \end{array}$$

$$\text{N} \rightarrow \begin{array}{cc} \text{D} & \text{N} \\ @\textsc{ext} & @\textsc{headx} \\ \uparrow=\downarrow & \uparrow=\downarrow \end{array}$$

$$\text{N} \rightarrow \begin{array}{cc} \text{N} & \text{P} \\ @\textsc{heada} & @\textsc{ext} \\ \uparrow=\downarrow & \downarrow \in (\uparrow\textsc{adj}) \end{array}$$

$$\text{N} \rightarrow \begin{array}{cc} \text{Adj} & \text{N} \\ @\textsc{nonpadj} & @\textsc{heada} \\ \downarrow \in (\uparrow\textsc{adj}) & \uparrow=\downarrow \end{array}$$

$$\text{N} \rightarrow \begin{array}{ccc} \text{N} & \text{(P)} & \text{(C)} \\ @\textsc{headx} & @\textsc{int} & @\textsc{int} \\ \uparrow=\downarrow & (\uparrow\textsc{obl})=\downarrow & (\uparrow\textsc{comp})=\downarrow \end{array}$$

$$\text{X} \rightarrow \begin{array}{ccc} \text{X} & \text{Conj} & \text{X} \\ @\textsc{heada} & @\textsc{nonprj} & @\textsc{heada} \\ \downarrow\in\uparrow & \uparrow=\downarrow & \downarrow\in\uparrow \end{array}$$

$$\text{D} \rightarrow \begin{array}{cc} \text{N} & \text{D} \\ @\textsc{int} & @\textsc{headx} \\ (\uparrow\textsc{poss}) = \downarrow & \uparrow=\downarrow \\ & \neg(\uparrow\textsc{spec}) \end{array}$$

$$\text{Adj} \rightarrow \begin{array}{cc} \text{Adj} & \text{P} \\ @\textsc{headx} & @\textsc{int} \\ \uparrow=\downarrow & (\uparrow\textsc{obl}) = \downarrow \end{array}$$

[19]Note that the optionality specified here is constrained in the XLE version so that at least one non-head daughter must appear. This is done by formulating this as a set of disjunctions: { V N (N) (P) (C) | V N (P) (C) | V P (C) | V C }.

$$\text{Adj} \rightarrow \begin{array}{cc} \text{Adv} & \text{Adj} \\ @\textsc{nonpadj} & @\textsc{heada} \\ \downarrow\in(\uparrow\textsc{adj}) & \uparrow=\downarrow \end{array}$$

$$\text{Adv} \rightarrow \begin{array}{cc} \text{Adv} & \text{Adv} \\ @\textsc{nonpadj} & @\textsc{heada} \\ \downarrow\in(\uparrow\textsc{adj}) & \uparrow=\downarrow \end{array}$$

$$\text{P} \rightarrow \begin{array}{cc} \text{P} & \text{N} \\ @\textsc{headx} & @\textsc{int} \\ \uparrow=\downarrow & (\uparrow\textsc{obj})=\downarrow \end{array}$$

Lexicon (sample):

Verbs:

eat/eats	V	(↑ PRED)=‘eat<SUBJ,OBJ>’ @PRJM
eating	V	(↑ PRED)=‘eat<SUBJ,OBJ>’ (↑VFORM) = ING @PRJM

Copula:

is/are	I	(↑ TENSE) = PRES (↑ VFORM) =c ING @PRJM
	V	(↑ PRED) = ‘be<SUBJ,PLINK>’ @PRJM

Auxiliaries:

did	I	(↑ TENSE) = PAST @PRJM
	C	(↑ TENSE) = PAST (↑ QUEST) = + @PRJM

Prepositions:

to	P	(↑ PRED)=‘to<OBJ>’ @PRJM

Conjunctions/complementizers:

and	Conj	(↑CONJFM) = AND @NONPRJM
that	C	(↑ COMPFM) = THAT @PRJM

Nouns:

dog	N	(↑ PRED) = ‘dog’ @PRJM

Determiners/pronouns:

the	D	(↑ EXT) = ‘the’ @PRJM
’s	D	(↑ DEF) = + (↑ POSS) @PRJM.
what	N	(↑ PRED) = ‘what’ @PRJM
	D	(↑ EXT) = ‘what’ @PRJM

Adjectives:

small	A	(↑ PRED) = ‘small’ @PRJM
	A	(↑ PRED) = ‘small’ @NONPRJM
asleep	A	(↑ PRED) = ‘asleep’ @PRJM
former	A	(↑PRED) = ‘former’ @NONPRJM

Adverbs:

very	Adv	(↑ PRED) = ‘very’ @NONPRJM
quickly	Adv	(↑ PRED) = ‘quickly’ @PRJM

Figure 1: C-structures for further illustration