

# Order and Optionality: Minimalist Grammars with Adjunction

Meaghan Fowlie

UCLA Linguistics

Los Angeles, California

mfowlie@ucla.edu

## Abstract

Adjuncts are characteristically optional, but many, such as adverbs and adjectives, are strictly ordered. In Minimalist Grammars (MGs), it is straightforward to account for optionality or ordering, but not both. I present an extension of MGs, MGs with Adjunction, which accounts for optionality and ordering simply by keeping track of two pieces of information at once: the original category of the adjoined-to phrase, and the category of the adjunct most recently adjoined. By imposing a partial order on the categories, the Adjoin operation can require that higher adjuncts precede lower adjuncts, but not vice versa, deriving order.

## 1 Introduction

The behaviour of adverbs and adjectives has qualities of both ordinary selection and something else, something unique to that of modifiers. This makes them difficult to model. Modifiers are generally optional and transparent to selection while arguments are required and driven by selection. In languages with relatively strict word order, arguments are strictly ordered, while modifiers may or may not be. In particular, (Cinque, 1999) proposes that adverbs, functional heads, and descriptive adjectives are underlyingly uniformly ordered across languages and models them by ordinary Merge or selection. Such a model captures only the ordering restrictions on these morphemes; it fails to capture their apparent optionality and transparency to selection. I propose a model of these ordered yet optional and transparent morphemes that introduces a function *Adjoin* which operates on pairs of categories: the original category of the modified phrase together with the category of the most recently adjoined modifier. This allows the derivation to keep track of both the true head of the

phrase and the place in the Cinque hierarchy of the modifier, preventing inverted modifier orders in the absence of Move.

## 2 Minimalist Grammars

I formulate my model as a variant of *Minimalist Grammars* (MGs), which are Stabler (1997)'s formalisation of Chomsky's (1995) notion of feature-driven derivations using the functions Merge and Move. MGs are mildly context-sensitive, putting them in the right general class for human language grammars. They are also simple and intuitive to work with. Another useful property is that the properties of well-formed *derivations* are easily separated from the properties of *derived structures* (Kobele et al., 2007). Minimalist Grammars have been proposed in a number of variants, with the same set of well-formed derivations, such as the string-generating grammar in Keenan & Stabler (2003), the tree-generating grammars in Stabler (1997) and Kobele et al (2007), and the multidominant graph-generating grammar in Fowlie (2011).

At the heart of each of these grammars is a function that takes two derived structures and puts them together, such as string concatenation or tree/graph building. To make this presentation as general as possible, I will simply call these functions **Com**. I will give derived structures as strings as (2003)'s grammar would generate them,<sup>1</sup> but this is just a place-holder for any derived structure the grammar might be defined to generate.

**Definition 2.1.** A *Minimalist Grammar* is a five-tuple  $G = \langle \Sigma, \text{sel}, \text{lic}, \text{Lex}, M \rangle$ .  $\Sigma$  is a finite set of symbols called the *alphabet*.  $\text{sel} \cup \text{lic}$  are finite sets of *base features*. Let  $F = \{+\text{f}, -\text{f}, =\text{X}, \text{X} \mid \text{f} \in$

<sup>1</sup>Keenan & Stabler's grammar also incorporates an additional element: lexical items are triples of string, features, and lexical status, which allows derivation of Spec-Head-Complement order. I will leave this out for simplicity, as it is not relevant here.

$\mathbf{lic}, X \in \mathbf{sel}\}$  be the *features*. For  $\epsilon$  the empty string,  $Lex \subseteq \Sigma \cup \{\epsilon\} \times F^*$  is the *lexicon*, and  $M$  is the set of operations **Merge** and **Move**. The language  $L_G$  is the closure of  $Lex$  under  $M$ . A set  $C \subseteq F$  of designated features can be added; these are the types of complete sentences.

Minimalist Grammars are *feature-driven*, meaning features of lexical items determine which operations can occur and when. There are two disjoint finite sets of features, *selectional* features  $\mathbf{sel}$  which drive the operation Merge and *licensing* features  $\mathbf{lic}$  which drive Move. Merge puts two derived structures together; Move operates on the already built structure. Each feature has a positive and negative version, and these features with their polarities make the set  $F$  from which the feature stacks for Lexical Items are drawn. In the course of the derivation the features will be *checked*, or deleted, by the operations Merge and Move.

<i>Polarity</i> →	Pos	Neg	
for <b>Merge</b>	=X	X	$X \in \mathbf{sel}$
for <b>Move</b>	+f	-f	$f \in \mathbf{lic}$

Table 1: Features

In order for a derivation to succeed, LIs must be in the following form:

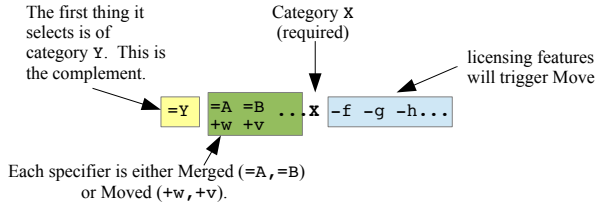


Figure 1: LI template

For example,  $\langle \text{kick}, =D=DV \rangle$  takes a complement of category D, a specifier of category D, and is itself a V.  $\langle \text{which}, =ND-wh \rangle$  takes an N as complement forming a D phrase, which will move because of feature *wh*.

Merge and Move are defined over *expressions*: sequences of pairs  $\langle$ derived structure, feature stack $\rangle$ . The first pair in the sequence can be thought of as the “main” structure being built; the remaining are waiting to move. An expression *displays* feature  $f$  just in case that feature is the first feature in the feature stack of the first pair.

An MG essentially works as follows: Merge is a binary operation driven by  $\mathbf{sel}$ . It takes two expres-

sions and combines them into one just in case the first expression displays =X and the second displays X for some  $X \in \mathbf{sel}$ . Once the second expression is selected, it may still have features remaining; these are always negative licensing features and mean that the second structure is going to move. As such it is stored separately by the derivation. When the matching positive licensing feature comes up later in the derivation, the moving structure is combined again. This is Move.

Move also carries the requirement that for each  $f \in \mathbf{lic}$  there be at most one structure waiting to move. This is the *shortest move constraint (SMC)*.<sup>2</sup>

**Definition 2.2 (Merge).** For  $\alpha, \beta$  sequences of negative  $\mathbf{lic}$  features,  $s, t$  derived structures:<sup>3</sup>

$$\mathbf{Merge}(\langle s, =X\alpha \rangle :: \text{movers}_s, \langle t, X\beta \rangle :: \text{movers}_t) = \begin{cases} \langle \mathbf{Com}(s, t), \alpha \rangle :: \text{movers}_s \cdot \text{movers}_t & \text{if } \beta = \epsilon \\ \langle s, \alpha \rangle :: \langle t, \beta \rangle :: \text{movers}_s \cdot \text{movers}_t & \text{if } \beta \neq \epsilon \end{cases}$$

**Definition 2.3 (Move).** For  $\alpha, \beta, \gamma$  sequences of negative  $\mathbf{lic}$  features,  $s, t$  derived structures, suppose  $\exists! \langle t, \beta \rangle \in \text{movers}$  such that  $\beta = -f\gamma$ . Then:  $\mathbf{Move}(\langle s, +f\alpha \rangle :: \text{movers}) = \begin{cases} \langle \mathbf{Com}(s, t), \alpha \rangle :: \text{movers} - \langle t, \beta \rangle & \text{if } \gamma = \epsilon \\ \langle s, \alpha \rangle :: \langle t, \gamma \rangle :: \text{movers} - \langle t, \beta \rangle & \text{if } \gamma \neq \epsilon \end{cases}$

In this article I will make use of *annotated derivation trees*, which are trees describing the derivation. In addition to the name of the function, I (redundantly) include for clarity the derived expressions in the form of strings and features, and sometimes an explanation of why the function applied. For example, Figure 2 shows derivations (unannotated and annotated) of *the wolf* with feature D.

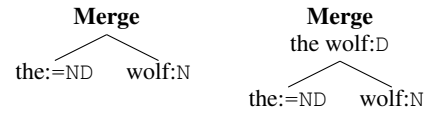


Figure 2: Unannotated and annotated derivation trees

<sup>2</sup>The SMC is based on economy arguments in the linguistic literature (Chomsky, 1995), but it is also crucial for a type of finiteness: the valid derivation trees of an MG form a regular tree language (Kobele et al., 2007). The number of possible movers must be finite for the automaton to be finite-state. The SMC could also be modified to allow up to a particular (finite) number of movers for each  $f \in \mathbf{lic}$ .

<sup>3</sup>::: adds an element to a list;  $\cdot$  appends two lists;  $-$  removes an element from a list.

### 3 Cartography

The phenomena this model is designed to account for are modifiers and other apparently optional projections such as the following:

- (1) a. The small ancient triangular green Irish pagan metal artifact was lost.  
 b. \*The metal green small artifact was lost. **Adjectives**  
 c. Frankly, John probably once usually arrived early.  
 d. \*Usually, John early frankly once arrived probably. **Adverbs**  
 e. [DP zhe [NumP yi [CIP zhi [NP bi]]]  
 [DP this [NumP one [CIP CL [NP pen]]]  
 'this pen' **Functional projections**

These three phenomena can all display *optionality*, *transparency to selection*, and *strict ordering*. By transparency I mean that despite the intervening modifiers, properties of the selected head are relevant to selection. For example, in a classifier language, the correct classifier selects a noun even if adjectives intervene.

The hypothesis that despite their optionality these projections are strictly ordered is part of *syntactic cartography* (Rizzi, 2004). Cinque (1999, 2010) in particular proposes a universal hierarchy of functional heads that select adverbs in their specifiers, yielding an order on both the heads and the adverbs. He proposes a parallel hierarchy of adjectives modifying nouns. These hierarchies are very deep. The adverbs and functional heads incorporate 30 heads and 30 adverbs.

Cinque argues that the surprising universality of adverb order calls for explanation. For example, Italian, English, Norwegian, Bosnian/Serbo-Croatian, Mandarin Chinese, and more show strong preferences for *frankly* to precede (*un*)*fortunately*. These arguments continue for a great deal more adverbs.<sup>4</sup>

- (2) Italian  
 a. **Francamente** ho *putroppo* una **Frankly** have *unfortunately* a pessima opinione di voi.  
 bad opinion of you  
 'Frankly I unfortunately have a very bad opinion of you.'  
 b. \**Putroppo* ho francamente una pessima *Unfortunately* have **frankly** a bad opinione di voi.  
 opinion of you
- (3) English  
 a. **Frankly**, I *unfortunately* have a very bad opinion of you

- b. ?*Unfortunately* I **frankly** have a very bad opinion of you
- (4) Norwegian  
 a. Per forlater [**rerlig talt**] [*heldigvis*]  
 Peter leaves [**honestly spoken**] [*fortunately*]  
 [nil] selskapet.  
 [now] the.party.  
 'Frankly, Peter is fortunately leaving the party now.'  
 b. \*Per forlater [*heldigvis*] [**rerlig talt**]  
 Peter leaves [*fortunately*] [**honestly spoken**]  
 [nil] selskapet.  
 [now] the.party.
- (5) Bosnian/Serbo-Croatian  
 a. **Iskreno**, ja *naialost* imam jako lose **Frankly**, I *unfortunately* have very bad misljenje o vama  
 opinion of you.  
 Frankly, I unfortunately have a very bad opinion of you.'  
 b. \**Naialost*, ja **iskreno** imam jako lose *unfortunately* I **frankly** have very bad misljenje o varna.  
 opinion of you.
- (6) Mandarin Chinese  
 a. **laoshi-shuo** wo *buxing* dui tamen you **Frankly**, I *unfortunately* to them have pian-jian.  
 prejudice  
 'Honestly I unfortunately have prejudice against them.'  
 b. \**buxing* wo **laoshi-shuo** dui tamen you *unfortunately* I **Frankly** to them have pian-jian.  
 prejudice

Supposing these hierarchies are indeed universal, the grammar should account for it. Moreover, in addition to strictly ordered adjuncts, ideally a model of adjunction should account for unordered adjuncts as well. For example, English PPs are unordered:

- (7) a. The alliance officer shot Kaeli **in the cargo hold** *with a gun*.  
 b. The alliance officer shot Kaeli *with a gun* **in the cargo hold**.

It is not unusual to see this kind of asymmetry, where right adjuncts are unordered but left adjuncts are ordered.

### 4 Previous approaches to adjunction

This section provides a brief overview of four approaches to adjunction. The first two are from a categorial grammar perspective and account for the optionality and, more or less, transparency to selection; however, they are designed to model unordered adjuncts. The other two are MG formal-

<sup>4</sup>Data from Cinque (1999)

isations of the cartographic approach. Since the cartographic approach takes adjuncts to be regular selectors, unsurprisingly they account for order, but not easily for optionality or transparency to selection.

#### 4.1 Categorical Grammar solutions

To account for the optionality and transparency, a common solution is for a modifier to combine with its modified phrase, and give the result the same category as the original phrase. In traditional categorial grammars, a nominal modifier has category  $N \setminus N$  or  $N/N$ , meaning it combines with an  $N$  and the result is an  $N$ .

Similarly, in MGs, an  $X$ -modifier has features  $=XX$ : it selects an  $X$  and the resulting structure has category feature  $X$ .

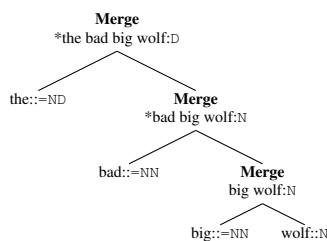


Figure 3: Traditional MG derivation of *\*the bad big wolf*

What this approach cannot account for is ordering. This is because the category of the new phrase is the same regardless of the modifier's place in the hierarchy. That is, the very thing that accounts for the optionality and the transparency of modifiers (that the category does not change) is what makes strict ordering impossible. Moreover, the modifier is not truly transparent to selection: the modifier in fact becomes the new head; it just happens to share a category with the original head. This can be seen in tree-generating grammars such as Stabler (1997) (Figure 4).

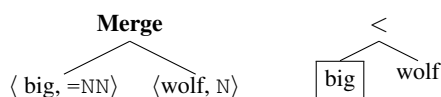


Figure 4: Derivation tree and derived *bare tree*. The  $<$  points to the head, *big*.

##### 4.1.1 Frey & Gärtner

Frey & Gärtner (2002) propose an improved version of the categorial grammar approach, one which keeps the modified element the head, giv-

ing true transparency to selection. They do this by asymmetric feature checking.

To the basic MG formalism a third polarity is added for **sel**,  $\approx X$ . This polarity drives the added function **Adjoin**. **Adjoin** behaves just like **Merge** except that instead of cancelling both  $\approx X$  and  $X$ , it cancels only  $\approx X$ , leaving the original  $X$  intact. This allows the phrase to be selected or adjoined to again by anything that selects or adjoins to  $X$ . This model accounts for optionality and true transparency: the modified element remains the head (Figure 4.1.1).

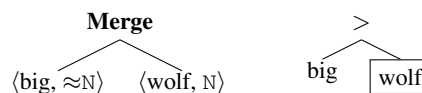


Figure 5: Frey & Gärtner: derivation tree and derived *bare tree*. The  $>$  points to the head, *wolf*.

Since this grammar is designed to model unordered modifiers, illicit orders are also derivable (Figure 6).

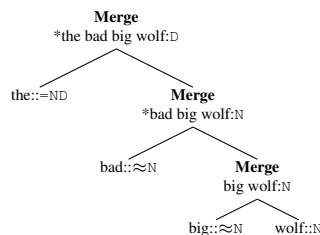


Figure 6: F & G derivation of *\*the bad big wolf*

#### 4.2 Selectional approach

A third approach is to treat adjuncts just like any other selector. This is the approach taken by syntactic cartography. Such an approach accounts straightforwardly for order, but not for optionality or transparency; this is unsurprising since the phenomena I am modelling share only ordering restrictions with ordinary selection.

The idea is to take the full hierarchy of modifiers and functional heads, and have each select the one below it; for example, *big* selects *bad* but not vice versa, and *bad* selects *wolf*. However, here we are left with the question of what to do when *bad* is not present, and the phrase is just *the big wolf*. *big* does not select *wolf*.

##### 4.2.1 Silent, meaningless heads

The first solution is to give each modifier and functional head a silent, meaningless version that serves only to tie the higher modifier to the lower.



For  $\langle A, \geq \rangle$  a partial order,  $a, b \in A$  are *incomparable*, written  $a \parallel b$ , iff  $a \not\geq b$  and  $b \not\geq a$ .

To shorten the definition of Adjoin, I define a function  $f_{\text{adj}}$  which determines the output features under Adjoin. If the adjunct belongs to the hierarchy of adjuncts being tracked by the second element of the feature pair, that second element changes. If not, the feature pair is unchanged.

**Definition 5.2.** For  $W, X, Y, Z \in \text{sel}$ ,  $W \in \text{ad}(Y)$  :

$$f_{\text{adj}}([W, X], [Y, Z]) = \begin{cases} [Y, W] & \text{if } W \geq Z \\ [Y, Z] & \text{if } W \parallel Z \\ \text{undefined} & \text{otherwise} \end{cases}$$

Notice that if  $Z$  and  $W$  are incomparable, no record is kept of the feature ( $W$ ) of the adjunct. This is just like Frey & Gärtner's asymmetric feature checking, and derives adjuncts that are unordered with respect to each other. In Definition 5.3, I model languages like English in which generally unordered adjuncts, like PPs, appear to the right, while ordered adjuncts, like adjectives, appear to the left. The rules could be easily modified for different orderings. See Section 6 for further discussion.

**Definition 5.3 (Adjoin).** For  $s, t$  derived structures,  $\gamma, \beta \in \{-f \mid f \in \text{lic}\}^*$ ,  $\alpha \in \{+f, =X \mid f \in \text{lic}, X \in \text{sel}\}^*$ ,  $W, X, Y, Z \in \text{sel}$ ,  $W \in \text{ad}(Y)$ ,  $C = f_{\text{adj}}([W, X], [Y, Z])$ :

$$\text{Adjoin}(\langle s, [W, X] \alpha \gamma \rangle :: \text{mvr}_s, \langle t, [Y, Z] \beta \rangle :: \text{mvr}_t) = \begin{cases} \langle \text{Com}(s, t), \alpha C \rangle :: \text{mvr}_s \cdot \text{mvr}_t & \text{if } \gamma, \beta = \epsilon \ \& \ W \geq Z \\ \langle \text{Com}(t, s), \alpha C \rangle :: \text{mvr}_s \cdot \text{mvr}_t & \text{if } \gamma, \beta = \epsilon \ \& \ W \parallel Z \\ \langle s, \alpha C \rangle :: \langle t, \beta \rangle :: \text{mvr}_s \cdot \text{mvr}_t & \text{if } \gamma = \epsilon, \beta \neq \epsilon \ \& \ W \not\geq Z \\ \langle t, \alpha C \rangle :: \langle s, \gamma \rangle :: \text{mvr}_s \cdot \text{mvr}_t & \text{if } \gamma \neq \epsilon, \beta = \epsilon \ \& \ W \not\geq Z \\ \langle \epsilon, \alpha C \rangle :: \langle s, \gamma \rangle :: \langle t, \beta \rangle :: \text{mvr}_s \cdot \text{mvr}_t & \text{if } \gamma, \beta \neq \epsilon \ \& \ W \not\geq Z \end{cases}$$

The first case is for ordered adjuncts where neither the adjunct nor the adjoined-to phrase will move (encoded in empty  $\gamma, \beta$ ). The second is the same but for unordered adjuncts, which will appear on the right. The last three cases are for moving adjunct, moving adjoined-to phrase, and both moving, respectively.  $\alpha$  is a sequence of positive licensing features, which allows adjuncts to take

specifiers.

Merge needs a slight modification, to incorporate the paired categories. Notice that Merge is interested only in the first element of the pair, the “real” category.

**Definition 5.4 (Merge).** For  $\alpha, \beta \in F^*$ ,  $s, t$  derived structures,  $X, Y \in \text{sel}$ :

$$\text{Merge}(\langle s, =X \alpha \rangle :: \text{mvr}_s, \langle t, [X, Y] \beta \rangle :: \text{mvr}_t) = \begin{cases} \langle \text{Com}(s, t), \alpha \rangle :: \text{mvr}_s \cdot \text{mvr}_t & \text{if } \beta = \epsilon \\ \langle s, \alpha \rangle :: \langle t, \beta \rangle :: \text{mvr}_s \cdot \text{mvr}_t & \text{if } \beta \neq \epsilon \end{cases}$$

Move remains as in definition 2.3 above.

## 5.1 Examples

MGAs are most easily understood by example. This first example demonstrates straightforward applications of Adjoin that derive strictly-ordered prenominal adjectives. *The big bad wolf* is derivable because the derivation remembers that an N-adjunct at level  $G$  in the hierarchy,  $\langle \text{bad}, [G, G] \rangle$ , adjoined to the noun. It encodes this fact in the second element of the pair  $[N, G]$ . *Big* is then able to adjoin because it too is an N-adjunct and it is higher in the hierarchy than *bad* ( $S > G$ ). Finally, *the* can be defined to select *wolf* directly.

Let  $\text{sel} = \{D, G, M, N, P, C, T, V\}$  and the partial order  $\geq$  on  $\text{sel}$  be such that  $D \geq S \geq G \geq M \geq N$  and  $C \geq T \geq V$

$$\text{adjuncts} = \{\langle N, \{S, G, M, P, C\} \rangle\}$$

$$\text{Lex} = \{\langle \text{bad}, [G, G] \rangle, \langle \text{big}, [S, S] \rangle, \langle \text{the}, =N[D, D] \rangle, \langle \text{wolf}, [N, N] \rangle, \langle \text{woods}, [N, N] \rangle, \langle \text{in}, =D[P, P] \rangle\}$$

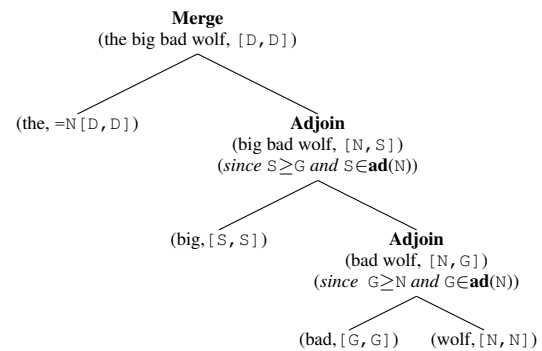


Figure 7: Valid derivation of *the big bad wolf*

\**Bad big wolf*, on the other hand, is not derivable without movement since the derivation remembers that *big*, which is at level  $S$  in the hierarchy, has already been adjoined. *bad*, being lower in the hierarchy, cannot adjoin.

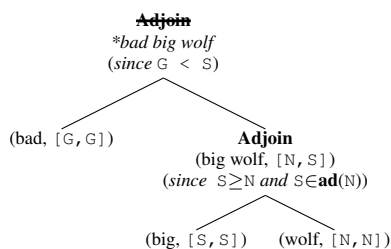


Figure 8: Invalid derivation of *\*bad big wolf*

This next example shows a right adjunct, a PP, being adjoined to an NP. Since  $P||N$  – that is, no hierarchical order is defined between  $N$  and  $P$  – the PP adjoins to the right, but does not alter the category of the noun.

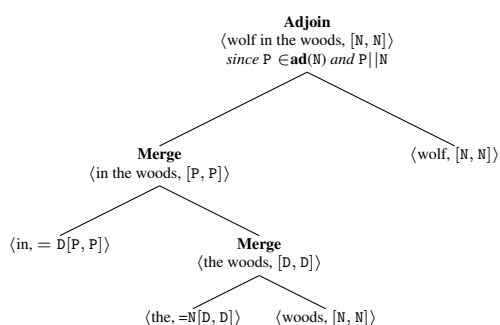


Figure 9: Right adjunction

## 6 Discussion and extensions

This model captures both the strict ordering of the merge-only models and the optionality and transparency to selection of the categorial approaches. Cinque’s observation that there is a hierarchy of functional heads and adverbs is modelled directly by defining a hierarchy in the grammar itself. The strict linear order falls out of the order imposed on the selectional features and the definition of Adjoin: adjunction is only defined when the hierarchy is respected. Optionality is the result of the transitivity of orders: intervening adjuncts are not necessary for a higher one to be adjoined. Transparency to selection is modelled by the pairing of the selectional features: the original category of the modified element is preserved, and Merge can see only that feature. The adjuncts are literally ignored.

The cross-linguistic consistency of the orders is accounted for by the claim that all human languages have the same partial order on **sel**. As such, it does not have to be learned, but rather comes with the grammar.

Computationally, this approach has an advantage over the merge-only model with homophony

as the latter increases the size of the lexicon by a polynomial function in the depths of the hierarchies of adjuncts, but the former does not.

### 6.1 Left and right adjuncts

As mentioned, I defined Adjoin to derive the asymmetry observed between left and right adjuncts in many languages: left adjuncts such as adverbs and descriptive adjectives are strictly ordered, while right adjuncts such as PPs and clauses are not. This fact is derived by letting the presence or absence of an ordering relation between the adjunct and modified category determine which case of Adjoin applies. If there is an order, the usual linear order will be calculated by Com, and the place in the hierarchy is tracked. Otherwise, the linear order is switched, and there is asymmetric feature checking.

If this is not the effect desired, there are alternatives. The simplest is to make the domain of the function  $ad\ sel \times \{right, left\}$ , specifying the sets of right and left adjuncts. This allows for much more flexibility, for good or ill. It does not derive the asymmetry, but does allow ordered and unordered adjuncts to appear on the same side of the head, if such a pattern is desired. This is an empirical question.

### 6.2 Selection and adjuncts

This model allows LIs that are in the set of adjuncts to be selected normally as arguments, since adjuncts have categories of their own. For example, *Red Ridinghood was small* is derivable by allowing *was* to select  $\langle small, [S, S] \rangle$ :  $\langle was, =S[V, V] \rangle$ . This is an improvement over models that do not give adjuncts categories of their own, such as Frey & Gärtner’s, but it is still lacking. In this model, there will have to be massive duplication in the lexicon so that *was* can select every adjective:  $\langle was, =S[V, V] \rangle$ ,  $\langle was, =G[V, V] \rangle$  etc.

To solve this problem, we can take advantage of the function **ad**, and define *was* to select anything from a particular image under **ad**. Such a model expands the definition of Merge to operate not only on categories, but also on sets of categories. The model would look something like this:

**Merge** $\langle was, =ad(N)[V, V] \rangle$ ,  $\langle small, [S, S] \rangle$  is defined iff  $S \in ad(N)$

Because the set of features  $F$  is finite, allowing Merge to be defined over subsets of  $F$  does not change the finite properties of MGs. Merge could in fact be allowed to be defined over any subset

of  $F$ . I suggest this model because it is restricted: only sets that exist for other reasons already can be quantified over.

MGAs also allow adjuncts to select arguments and license Move. For example, a preposition can select a complement before becoming an adjunct PP. Moreover, a functional projection such as Focus can Move a focused phrase into its specifier from the main tree, or Topic can Merge a specifier. The latter is a result of allowing positive polarity features to follow the category pair. Recall that in traditional MGs, an LI must be of the following form for the derivation to succeed, where each  $p_i$  is a positive polarity feature,  $X, Y \in \text{sel}$  and each  $f_i \in \text{lic}$ :

$$(\text{= } Y(p_1 p_2 \dots p_n)) X(-f_1 -f_2 \dots -f_m)$$

However, in MGAs, LIs of the following form are possible if the LI will Adjoin, the crucial difference being the presence of  $p_{n+1} \dots p_k$ :

$$(\text{= } Y(p_1 p_2 \dots p_n)) [X, Y](p_{n+1} \dots p_k) (-f_1 -f_2 \dots -f_m)$$

Figure 10 shows the end of a derivation in which the mover *briefly* is an adjunct, and so the licenser, the null  $F_{OC}$  head. Its positive licensing feature  $+f_{OC}$  moves to the front of the stack of the derived structure's features.

Suppose  $F_{OC} \in \text{ad}(T)$  and  $F_{OC} \geq T$ .

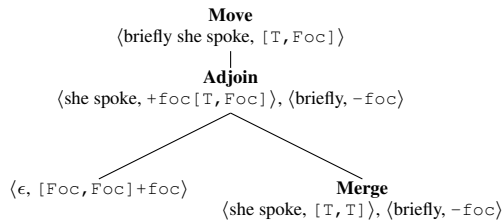


Figure 10: Adjunct FocP with moved specifier.

### 6.3 Adjuncts of adjuncts

In natural language, adjuncts can also be adjoined to, for example as in *the very bad wolf*. The function **ad** maps single categories to their adjuncts, but it is not generally the case that, say, an adverb, can only adjoin to certain adjectives. In order to capture this fact without duplication in the lexicon, Adjoin, like Merge, can be extended to allow subsets of  $F$ . Similarly to the Merge case, we can restrict these subsets by requiring that they be the image of a category under **ad**. For example:

$\langle \text{frankly}, [F_r, F_r] \rangle, \langle \text{unfortunately}, [F_o, F_o] \rangle, \langle \text{allegedly}, [A_l, A_l] \rangle, \langle \text{bad}, [G, G] \rangle, \langle \text{wolf}, [N, N] \rangle \in \text{Lex}$   
 $F_r \geq F_o \geq A_l \geq V, S \geq G \geq N, P$   
 $\text{ad}(N) = \{S, G, P\}$   
 $\text{ad}(V) = \text{ad}(S) = \text{ad}(G) = \{F_r, F_o, A_l\}$

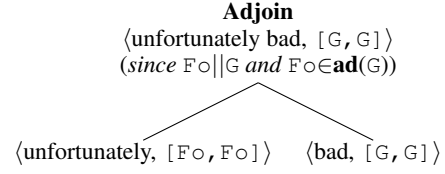


Figure 11: Adjoining to an adjunct

Notice however that we are still missing a generalisation:  $S, G$ , and indeed all adjectives have the same adjuncts. Now, this can be modelled by calling this set  $\text{ad}(\text{ad}(N))$ . However, such a solution assumes a special status for  $N$  over many other categories such as  $G$ : why  $\text{ad}(\text{ad}(N))$  rather than  $\text{ad}(\text{ad}(G))$ ? I would argue that such a status would reflect the reality of natural language. We can see  $N$  and  $V$  behaving in special ways: both are at the bottom of hierarchies, for example. However, as far as I am aware, no such status exists in any MGs. Formalising these observations is a matter for further research.

### 6.4 Islandhood

Adjuncts have another classic property: islandhood. Movement is not possible out of certain types of adjuncts.

- (8) a. You left [because your ex showed up]<sub>Adj</sub>  
 b. \*Who did you leave [because \_\_\_\_ showed up]<sub>Adj</sub>?

Any approach that keeps Adjoin separate from Merge introduces the option of stipulating the Adjunct Island Constraint (AIC), either as a separate constraint on Adjoin, as Frey & Gärtner do, or by simply not including movers<sub>s</sub> in the definition of Adjoin, making the function undefined when the adjunct carries movers. This is not very satisfying, though: better perhaps would be to derive it, as Graf (2013) does. On the other hand, perhaps not all adjuncts are islands. If *beside* \_\_\_\_ is an adjunct in (9), it is not an adjunct island.

- (9) Who are you sitting [beside \_\_\_\_]<sub>Adjunct</sub>?

As always, islands must remain a matter for further research.

## 7 Conclusion

I have presented a model of adjunction that accounts for both the optionality and the strict or-



dering of many adjuncts. MGAs accomplish this by the simple expedience of keeping track of two pieces of information at once: the original category of the projecting phrase, and the category of the most recent adjunct to adjoin. This allows Adjoin to be defined to only apply when the next adjunct is not lower in a hierarchy than the last. At the same time, Merge can see the original category, and ignores the adjunct's category.

I have also suggested some extensions of MGAs to more efficiently account for adjuncts as the second argument of Merge and Adjoin. These involved quantification over categories, with the added suggestion that the sets of categories in question be restricted by the sets of adjuncts already defined.

Future directions for this research include not only matters internal to the model, such as how best to model adjuncts of adjuncts, but also larger questions of the mathematical properties of MGAs. MGAs are weakly equivalent to MGs, since MGAs merely take existing ways to derive certain strings and seek more efficient ways, which capture more generalisations. If every adjunct in the lexicon is replaced with the right set of selectors, Adjoin does not need to be used. For example, the adjectives in the MGA lexicon used in the examples in Section 5.1 can be replaced by the adjectives in either grammar from the selectional approaches in Section 4.2, and the same string set can be generated.

Clearly MGs and MGAs are not strongly equivalent: the derivation trees differ in that MGAs have a function that is not present in MGs.

Because the possible configurations of features remains finite, the derivation tree languages of MGAs should prove to be regular, following Kobele et al (2007)'s presentation: transition rules for Adjoin need merely be added.

Also of interest are the subregular properties of the derivation tree language. Although to my knowledge such notions as *tierwise strictly local* (Heinz et al., 2011) have not yet been formally defined for tree languages, I conjecture that in MGAs, Merge is tierwise strictly  $k$ -local, and Adjoin is strictly  $k$ -local.

## References

- Noam Chomsky. 1995. *The Minimalist Program*. MIT Press, Cambridge, MA.
- Guglielmo Cinque. 1999. *Adverbs and functional heads: a cross-linguistic perspective*. Oxford studies in comparative syntax. Oxford University Press, Oxford.
- Guglielmo Cinque. 2010. *The syntax of adjectives: a comparative study*. Linguistic Inquiry monographs. MIT Press, Cambridge, MA.
- Meaghan Fowlie. 2011. Multidominant minimalist grammars. Master's thesis, University of California, Los Angeles.
- Werner Frey and Hans-Martin Gärtner. 2002. On the treatment of scrambling and adjunction in minimalist grammars. In *Proceedings of the Conference on Formal Grammar (FGTrento)*, pages 41–52, Trento.
- Thomas Graf. 2013. The price of freedom: Why adjuncts are islands. Slides of a talk given at the Deutsche Gesellschaft für Sprachwissenschaft 2013, March 12–15, University of Potsdam, Potsdam, Germany.
- Jeffrey Heinz, Chetan Rawal, and Herbert Tanner. 2011. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Edward L. Keenan and Edward P. Stabler. 2003. *Bare Grammar*. CSLI Publications, Stanford.
- Gregory M. Kobele, Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to minimalism. In J. Rogers and S. Kepser, editors, *Model Theoretic Syntax at ESSLLI '07*. ESSLLI.
- Luigi Rizzi. 2004. Locality and left periphery. In Adriana Belletti, editor, *Structures and Beyond: The Cartography of Syntactic Structures*, volume 3, pages 223–251. Oxford University Press, Oxford.
- Edward Stabler. 1997. Derivational minimalism. *Logical Aspects of Computational Linguistics*, pages 68–95.
- Edward Stabler. 2013. Bracketing paradoxes and opacity: Parsing late adjunction in copy constructions. Talk given at UCLA Mathematical Linguistics Circle, April.