Adjuncts and Minimalist Grammars*

Meaghan Fowlie

UCLA Linguistics,
3125 Campbell Hall, UCLA, Los Angeles, California, USA
`http://mfowlie.bol.ucla.edu`

Abstract. The behaviour of adverbs and adjectives has qualities of both ordinary selection and something else, something unique to modifiers. This makes them difficult to model. Modifiers are generally optional and transparent to selection while arguments are required and driven by selection. Cinque [4] 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 optionality and transparency to selection. I propose a model of adjunction with a separate Adjoin function that 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.

Keywords: adjoin, minimalist grammars, adjectives, adverbs, functional projections, ordering, optionality

1 Introduction

Adjuncts are optional, meaning the sentence is grammatical without them. For example, in (1-a), *red* is optional. They are transparent to selection in that the selector seems to select for the features of the head, not those of the intervening adjunct. For example, in (1-b), the gender of *boek* 'book' is neuter. The intervening adjective does not have gender agreement, so *het* selects *boek* for its gender, regardless of the intervening adjunct.

(1) a. The (red) rose **Optionality**

b. Het mooi-e boek
the.NEU beautiful-DET book
'The beautiful book' (Dutch) **Transparency**

Many languages have a default order for adjuncts, with unmarked intonation and without special scopal meaning. For example, English has ordered adjectives.

* Many thanks to Ed Stabler, my dissertation committee chair, as well as to the rest of my committee (Ed Keenan, Martin Monti, and Carson Schutze). Thank you also to Thomas Graf for our MG discussions, UCLA syntax/semantics seminar, audiences at MoL13 and NWLC 2013, and of course to three very helpful anonymous reviewers.

(2) a. Wear the enormous ugly green hat
Wear the hat that is enormous, ugly, and green
b. #Wear the ugly enormous green hat
Of your enormous green hats, wear the ugly one.

This paper proposes a minimalist model of ordered adjuncts, using a new function adjoin that has access to sets of adjuncts for each category and hierarchy levels of adjuncts.

2 Minimalist Grammars

I formulate my model as a variant of *Minimalist Grammars* (MGs), which are Stabler's [14] formalisation of Chomsky's [3] 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.

At the heart of MGs is a function that takes two structures and puts them together. I will give derived structures as strings as Keenan & Stabler's [10] grammar would generate them.[1]

Definition 1. *A **Minimalist Grammar** is a five-tuple $G = \langle \Sigma, \mathbf{sel}, \mathbf{lic}, Lex, M \rangle$. Σ is the alphabet.* **sel**∪**lic** *are the base features. Let F={`+f`, $-$`f`,`=X`, `X`|`f` ∈ **lic**, `X` ∈ **sel**} be the features. $Lex \subseteq \Sigma \times F^*$, 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 finite sets of features, ***selectional*** features **sel** which drive the operation **Merge** and ***licensing*** features **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. Positive **sel** and **lic** features are `=X` and `+f` respectively, and negatives are `X` and `-f`. Intuitively, negative **sel** features are the categories of lexical items. Merge and Move are defined over ***expressions***: sequences of pairs ⟨derived structure, feature stack⟩. The first pair can be thought of as the "main" structure being built; the remaining are waiting to move.

An MG essentially works as follows: **Merge** takes two expressions and combines them into one if the first structure displays `=X` and the second `X` for some `X` ∈ **sel**. The `X` features are deleted, after which the second structure may still

[1] 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 really relevant here, as our interest is in spec/adjunct placement, which will always be on the left. For convenience of English reading, I will give sentences in head-spec-complement order, but the formal definition I give here always puts the selected on the left and the selector on the right.

have features remaining, meaning the second structure is going to move. It is stored separately by the derivation until the matching positive licensing feature comes up later in the derivation, when the moving structure is combined again; this is **Move**. **Move** also carries the requirement that for each f∈**lic** there be at most one structure waiting to move. This is the ***shortest move constraint (SMC)***.[2]

Definition 2 (*Merge*). *For α, β sequences of negative* **lic** *features, s, t derived structures, $mvrs_{s,t}$ expressions:*[3]

$$\mathbf{Merge}(s : \texttt{=X}\alpha :: mvrs_s,\ t : \texttt{X}\beta :: mvrs_t) = \begin{cases} ts : \alpha :: mvrs_s \cdot mvrs_t & \text{if } \beta = \epsilon \\ (s : \alpha) :: (t : \beta) :: mvrs_s \cdot mvrs_t & \text{if } \beta \neq \epsilon \end{cases}$$

Definition 3 (*Move*). *For α, β, γ sequences of negative* **lic** *features, s, t derived structures, mvrs an expression, suppose $\exists!\langle t, \beta\rangle \in mvrs$ such that $\beta = \texttt{-}f\gamma$. Then:*

$$\mathbf{Move}(s : \texttt{+f}\alpha :: mvrs) = \begin{cases} ts : \alpha :: mvrs & \text{if } \gamma = \epsilon \\ s : \alpha :: t : \gamma :: mvrs) & \text{if } \gamma \neq \epsilon \end{cases}$$

In this article I will make use of *derivation trees*, which are trees describing the derivation. They may also be annotated: in addition to the name of function, I (redundantly) include for clarity the derived expressions in the form of strings and features. For example, figure 1 shows derivation trees (annotated and unannotated) of *the wolf* with feature D.

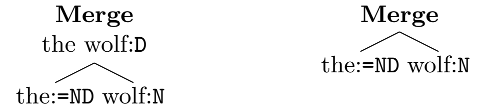


Fig. 1. Annotated and unannotated derivation trees

[2] The SMC is based on economy arguments in the linguistic literature [3], but it is also crucial for a type of finiteness: the valid derivation trees of an MG form a regular tree language [11]. 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∈**lic**.

[3] :: adds an element to a list; · appends two lists.

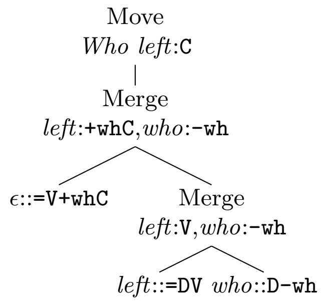


Fig. 2. Example: *Who left?*

3 Cartography

Despite their optionality, linguists, most famously Cinque [4], argue that certain adjuncts have a default order that is consistent across languages. The phenomena this model is designed to account for are modifiers and other apparently optional projections such as the following.

(3) 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. [Il premio Nobel]$_{\text{top}}$, [a chi]$_{\text{wh}}$ lo daranno?
[the prize Nobel]$_{\text{top}}$, [to whom]$_{\text{wh}}$ it give.fut
The Nobel Prize, to whom will they give it? **Left periphery**
f. [$_{\text{DP}}$ zhe [$_{\text{NumP}}$ yi [$_{\text{ClP}}$ zhi [$_{\text{NP}}$ bi]]]
[$_{\text{DP}}$ this [$_{\text{NumP}}$ one [$_{\text{ClP}}$ CL [$_{\text{NP}}$ pen]]]
'this pen' **Functional projections**

These three phenomena 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 the noun even if adjectives intervene.

The hypothesis that despite their optionality these projections are strictly ordered is part of *syntactic cartography* [12]. Cinque [4], [5] 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, Bosnian/Serbo-Croatian, Mandarin Chinese, and more show strong preferences for *frankly* to precede *unfortunately*. These arguments continue for a great deal more adverbs.[4]

[4] Data from [4]

(4) Italian

a. **Francamente** ho *purtroppo* una pessima opinione di voi.
Frankly have *unfortunately* a bad opinion of you
'Frankly I unfortunately have a very bad opinion of you.'

b. *Purtroppo ho francamente una pessima opinione di voi.
Unfortuately have **frankly** a bad opinion of you

(5) English

a. **Frankly**, I *unfortuately* have a very bad opinion of you

b. ?*Unfortunately* I **frankly** have a very bad opinion of you

(6) Bosnian/Serbo-Croatian

a. **lskreno**, ja *naialost* imam jako lose misljenje o vama
Frankly, I *unfortunately* have very bad opinion of you.
Frankly, I unfortunately have a very bad opinion of you.'

b. **Naialost*, ja **iskreno** imam jako lose misljenje o varna.
unfortunately I **frankly** have very bad opinion of you.

(7) Mandarin Chinese

a. **laoshi-shuo** wo *buxing* dui tamen you pian-jian.
Frankly, I *unfortunately* to them have prejudice
'Honestly I unfortunately have prejudice against them.'

b. **buxing* wo **laoshi-shuo** dui tamen you pian-jian.
unfortunately I **Frankly** to them have prejudice

Supposing these hierarchies are indeed universal, the grammar should account for it.

4 Desiderata

In addition to these three main properties, an account of adjuncts should ideally also account for the following: selectability of adjunct categories, adjuncts of adjuncts, unordered adjuncts, so-called obligatory adjuncts, and adjunct islands.

(8) Mary is **tall** *tall* is selected by *is*

(9) The **surprisingly short** basketball player *surprisingly* modifies *short*

(10) 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**. English PP adjuncts are unordered

(11) a. He makes a **good** father *good* is an adjunct but is not optional
b. *He makes a father
c. She worded the letter **carelessly**.
d. ...and Marc did so **carefully**. *carefully* is an adjunct
e. *She worded the letter. yet it is not optional

(12) a. He left [because she arrived]$_{\text{adjunct}}$.
b. *Who did he leave because __ arrived? (some) adjuncts are islands

c. He thinks [she arrived]$_{\text{object}}$.
d. Who does he think __ arrived? Embedded object CPs are not islands; islandhood is a property of adjuncts, not embedded clauses in general.

In sum, an account of adjuncts in minimalist grammars should ideally have the following properties:

1. **Optionality**: sentences should be grammatical with or without adjuncts
2. **Transparency to selection**: If a phrase P is normally selected by head Q, when P has adjuncts Q should still select P.
3. **Order**: there should be a mechanism for forcing an order on adjuncts
4. **Selectability** (8)
   (a) **Efficiency:** All adjectives are possible arguments of the same predicates, so there should be a way to select for any adjective, rather than cross-listing the selector for each adjective category.
5. **Adjuncts of adjuncts** (9)
   (a) **Efficiency:** Similarly to selection, there should be a way to say that, say, an adverbs can adjoin to all adjective, rather than having a homophonous form of the adverb for each adjective category.
6. **Unordered** (10)
7. **Obligatory adjuncts** (11)
8. **Islands** (12)

5 Previous Approaches to Adjunction

This section provides a brief overview of three 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 last is an MG formalisation 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.

5.1 Traditional MG/CG solution

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. Traditionally in MGs, an `X`-modifier has features `=XX`: it selects an `X` and the resulting structure has category feature `X`. Similarly, in categorial grammars, an X-modifier has category X/X or X\X. As such, the properties of traditional MG and CG models of adjunction are the same.[5]

[5] This is not the only possible solution using the MG architecture, but rather the traditional solution. Section 5.2 gives a model within MGs that accounts for order.

An anonymous reviewer suggested a different solution, with a set of silent, meaningless heads that turn categories into selectors of their adjuncts, for example ϵ::`=N =Adj =N`. Such a solution does much better on desiderata 4 and 5 than the one given here, but shares with the cartographic solution given in section 5.2 the problem of linguistic undesirability of silent, meaningless elements.

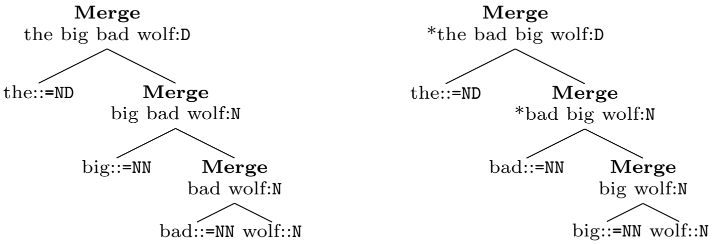
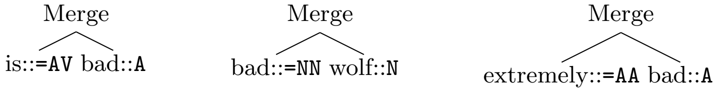


Fig. 3. Traditional MG approach

1. **Optionality**: ✓the original category is kept
2. **Transparency to selection**: **Sort of:** in Fig. 3, *the* selects N, but the N it checks is the one introduced by *bad*, not the one on *wolf.*
3. **Order**: **No**, the original category is kept so any adjunct may adjoin at any time.
4. **Selectability** Adjuncts need two versions, one for being adjuncts and the other for being selected. For example, *bad*::=NN cannot be selected by anything until it has itself selected an N.
   turn an =NN into an N by being selected; however, such a solution predicts the general existence of silent Ns, and zero-derivation of adjectives from nouns; indeed, silent, meaningless versions of any modifiable category and zero-derivation of any modifiable category to its modifiers' categories.

   (a) **Efficiency: No.** Adjuncts have two versions, or else we permit new silent categories and zero-derivation.
5. **Adjuncts of adjuncts** Since adjunction is selection in this model, we have the same problem, but with the same solution: the feature for selection is also the feature for being adjoined to.

   (a) **Efficiency:** The homophony for selection covers adjunction too.
6. **Unordered** ✓ All adjuncts are unordered in this model
7. **Obligatory adjuncts: No**, there is no way to distinguish between an phrase with an adjunct and one without.
8. **Islands** Not without additional constraints. See Graf [8] for an account that will work with the present approach.

Frey & Gärtner Frey & Gärtner [7] propose an improved version of the categorial grammar approach, one which keeps the modified element the head, giving true transparency to selection. They do this by asymmetric feature checking. To the basic MG formalism a third polarity is added for **sel**, ≈X. This polarity drives the added function Adjoin. Adjoin behaves just like Merge except that instead of cancelling both ≈X and X, it cancels only ≈X, leaving the original X in tact. 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, but it is not designed to capture ordered adjuncts. Also, since adjuncts don't have categories of their own (just $\approx$`X`), it is not clear how to model selection of and adjunction to adjuncts.[6]

5.2 Selectional approach

A third approach is to treat adjuncts just like any other selector. This is the approach implicitly taken by syntactic cartography in mainstream linguistics.[7] 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.* I will briefly outline one solution, in which the full structure is always present.

We give each modifier and functional head a silent, meaningless version that serves only to tie the higher modifier to the lower, like syntactic glue. For example, we add to the lexicon a silent, meaningless "size" modifier that goes where *big* and *small* and other LIs of category `S` go.

the::`=S D`	ϵ::`=S D`	wolf::`N`	
big::`=G S`	ϵ::`=G S`	bad::`=N G`	ϵ::`=N G`

This solution doubles substantial portions of the lexicon. Doubling is not computationally significant, but it does indicate a missing generalisation: somehow, it just happens that each of these modifiers has a silent, meaningless doppelganger. Relatedly, the ordering facts are epiphenomenal. There is no universal principle predicting the fairly robust cross-linguistic regularity. Moreover, normally when something silent is in the derivation, we want to say it is contributing something semantically. Here these morphemes are nothing more than a trick to hold the syntax together.

1. **Optionality**: ✓Choose the right version of an LI. Note this is **inefficient**.
2. **Transparency to selection**: No, selection is of the adjunct, not the head. For example, in the lexicon above, *the* selects the (possibly empty) adjunct with features `=G S`, not the noun.
3. **Order**: ✓This is Merge, so order is determined by the particular lexical items' feature stacks

[6] This is not to say that it cannot be done. [7] has examples with selectional features `=`$\approx$`X`, though there is little discussion.

[7] It is not clear whether we should take mainstream syntax approaches to mean that there are always-present, silent, meaningless, functional heads. Another interpretation of their models is that each functional head on the hierarchy has a set of homophones, one for each level down in the hierarchy.

4. **Selectability** ✓This is ordinary Merge, so selection proceeds as usual.
   (a) **Efficiency:** ✓No homophony added for selection
5. **Adjuncts of adjuncts** Adjuncts of adjuncts are simply selectors of adjuncts.
   (a) **Efficiency:** ✓No homophony added for adjuncts of adjuncts
6. **Unordered** Very difficult, but possible with enough homophony.
7. **Obligatory adjuncts: No**, the same thing that allows optionality of adjuncts prevents us from requiring that an adjunct be present.
8. **Islands: No**, not without additional constraints. Again, see [8] for constraints that may work here.

6 Proposal

I propose a solution, which I will call Minimalist Grammars with Adjunction (MGAs),[8] which accounts for ordering by indexing phrases according to the hierarchy level of the last adjunct adjoined to them.

A given adjunct phrase P needs four pieces of information: P's category, what P is an adjunct of, what level adjunct P is, and what level the last adjunct that adjoin to P was. We need to know what a category is an adjunct of because that will determine whether, say, an adjective can adjoin to a noun phrase. I include in the grammar a set of adjuncts for each category. The hierarchy level of the adjunct is needed for when it acts as an adjunct. If the phrase it is adjoining to already has a adjunct, we need to check that the new adjunct is higher in the hierarchy. For this purpose, every phrase carries with it an additional number, indexing the level of its last adjunct. The two numbers are kept separate so that adjuncts can have adjuncts, as in *bright blue*. *Bright blue* has an adjunct *bright*, which may affect what further adjuncts can adjoin to it, but which does not affect what the phrase *bright blue* can adjoin to.

To track hierarchy level, each category feature is expanded into a triple consisting of the category feature, the level of the hierarchy of adjuncts the head is at, and the level of hierarchy the whole phrase is at. Hierarchy levels are encoded as natural numbers,[9] and $\leq$ is the usual order on $\mathbb{N}$.[10] These numbers are lexically specified; for example *wolf*::`[N,0,0]` would be in the lexicon.

By splitting the category into its category and its level as adjunct, we can allow all, say, adjectives, to have the same category. This extends the efficiency gains in [6] to selection of adjuncts and adjuncts of adjuncts.

When adjunct `[Y,n,m]` adjoins to something of category `[X,i,j]`, the resulting phrase is of category `[X,i,m]`, $\mathtt{i},\mathtt{j},\mathtt{n},\mathtt{m} \in \mathbb{N}$. The second number is what

[8] My earlier paper [6] used this name as well; this model is designed to improve on it.

[9] This is a similar approach to that taken by Adger [1], who proposes a second version of Merge that models the functional heads in a hierarchy. Our approaches differ in that in MGAs the original category is kept under Adjoin. A general discussion of the use of explicit hierarchy in grammars can be found in [2].

[10] $\mathbb{N}$ is simply acting as an index set, and that the maximal depth of hierarchies in a language bounds the actual index set for the grammar.

tracks the level of the hierarchy the phrase is at; it is the only thing that can change.

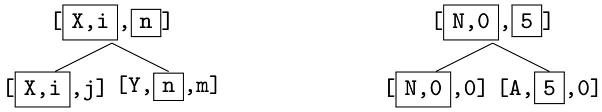


Fig. 4. Adjoin. The category feature of the new phrase is the first two elements of the adjoined-to phrase followed by the second element of the adjunct

6.1 Example

Before I give the full formal definition I will present an example. Suppose we have a grammar in which the adjunct sets are defined as follows:

Ad(`N`)={`Adj, P, C`}, Ad(`Adj`)={`Adv, Int`}, Ad(`Adv`)={`Int`}, Ad(`V`)={`Adv,T`}

We can derive *Apparently, John very often sang* as in figure 5. *very* adjoins to *often* since *often* is at level 0 and *very* is at level 3, and $3 \geq 0$. The whole phrase adjoins to *sang* since it's at level 18 and *sang* is at 0. `T` Merges to the VP, yielding a phrase at level 25. *Apparently* is at level 26, so it can adjoin.

To get order, we require that the first number of the adjunct be at least as high as the second number of the adjoined-to phrase. For example, in Figure 6, the derivation of *the big bad wolf* works because `Adj`∈**Ad**(`N`), and $6 > 4 > 0$. The derivation of *the bad big wolf* fails because the category of *big wolf* is `[N,0,6]`. *bad*::`[Adj,4,0]` can't adjoin to it because *bad* is a level-4 adjunct, but *big wolf* is already at level 6, and $4 < 6$.

6.2 Definition

Merge must be trivially redefined for categories as triples. **Merge** only cares about category, so it looks to match the positive selectional feature with the first element of the triple. (**Move** is unchanged.)

Definition 4 (Merge). *For $\alpha, \beta \in F^*$; s, t strings:*

$$\textbf{Merge}(\langle s, \texttt{=X}\alpha\rangle :: mvrs_s, \langle t, [\texttt{X},\texttt{i},\texttt{j}]\beta\rangle :: mvrs_t) = \begin{cases} \langle st, \alpha\rangle :: mvrs_s \cdot mvrs_t & \textit{if } \beta = \epsilon \\ \langle s, \alpha\rangle :: \langle t, \beta\rangle :: mvrs_s \cdot mvrs_t & \textit{if } \beta \neq \epsilon \end{cases}$$

Adjoin applies when the category of the adjunct is an adjunct of the category it is adjoining to, and if the adjunct is a k-level adjunct then the level of the phrase it is adjoining to is no higher than k. **Move** works as expected: the adjunct has negative licensing features left after it has had its category feature checked by **Adjoin**, it is added to the list of movers.

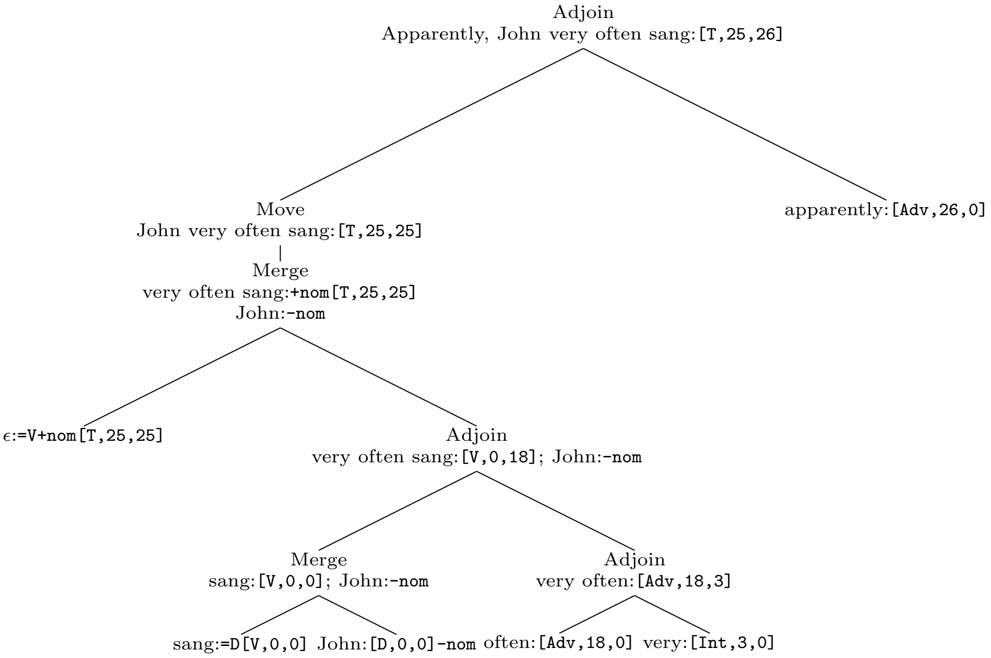


Fig. 5. Adjunct of adjunct; functional head merge; adjunction after functional head merge

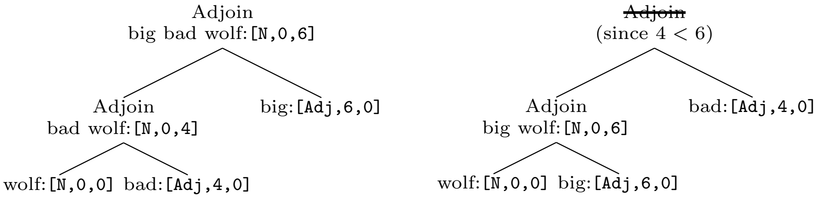


Fig. 6. Adjunct ordering: valid and invalid derivations

Definition 5 (Adjoin). *Let $s, t \in \Sigma$ be strings, $\mathtt{Y,X}$∈**sel** be categories, $i, j, n, m \in \mathbb{N}$, $mvrs \in (\Sigma^* \times F)^*$ be a mover list, and $\alpha, \beta \in F^*$.*

$$\begin{aligned}&\mathbf{Adjoin}(\langle s, [\mathtt{X}, \mathtt{i}, \mathtt{j}]\alpha :: mvrs\rangle, \langle t, [\mathtt{Y}, \mathtt{n}, \mathtt{m}]\beta\rangle)\\ &= \begin{cases}\langle ts, [\mathtt{X}, \mathtt{i}, \mathtt{n}]\alpha\rangle :: mvrs & \textit{if}\ \mathtt{n} \geq \mathtt{j}\ \mathit{\&}\ \mathtt{Y} \in \mathbf{Ad}(\mathtt{X})\ \mathit{\&}\ \beta = \epsilon\\ \langle s, [\mathtt{X}, \mathtt{i}, \mathtt{n}]\rangle :: \langle t, \beta\rangle :: mvrs & \textit{if}\ \mathtt{n} \geq \mathtt{j}\ \mathit{\&}\ \mathtt{Y} \in \mathbf{Ad}(\mathtt{X})\ \mathit{\&}\ \beta \neq \epsilon\end{cases}\end{aligned}$$

Notice that for Merge, there may be a mover list with both arguments ($movers_s$ and $movers_t$). Island constraints for adjoin are implemented by simply leaving out the mover list that would come with the adjunct. Adjoin is not defined when the adjunct has a mover.[11] This is not necessarily as stipulative as it sounds: [8] puts forth that for adjuncts to be truly optional they cannot have movers, or else the derivation tree without the adjunct would have an unchecked positive licensing feature. My definition of Adjoin is simply a way of conforming to this constraint.

Definition 6 (MGA). *A **Minimalist Grammar with Adjunction** is a six-tuple*

$G = \langle \Sigma, \mathbf{sel}, \mathbf{lic}, \mathbf{Ad}, Lex, M\rangle$. Σ is the alphabet. **sel**∪**lic** *are the base features. Let $F = \{+\mathtt{f}, -\mathtt{f}, \mathtt{=X}, [\mathtt{X}, \mathtt{n}, \mathtt{m}] | \mathtt{f} \in \mathbf{lic}; \mathtt{X}, \mathtt{Y} \in \mathbf{sel}; \mathtt{m}, \mathtt{n}\mathbb{N}\}$. $\mathbf{Ad} : \mathbf{sel} \rightarrow \mathcal{P}(\mathbf{sel})$ maps categories to their adjuncts. $Lex \subseteq_{fin} \Sigma \times F^*$, and M is the set of operations **Merge, Move,** and **Adjoin**. The language L_G is the closure of Lex under M. A set $C \subseteq \mathbf{sel}$ of designated features can be added; $\{[\mathtt{c}, \mathtt{i}, \mathtt{j}] | \mathtt{c} \in C; i, j \in \mathbb{N}\}$ are the types of complete sentences.*

6.3 Adverbs and Functional Heads

Contra Cinque [4], I model adverbs as separate from functional heads. Adverbs and adjectives differ from functional heads in two ways. First, they are not themselves adjoined to, while adverbs and adjectives are (*very blue*). Second, functional heads are sometimes required and sometimes optional. For example, English requires T, but not, perhaps, $\mathtt{Mod}_{\text{epistemic}}$ in every sentence. To model this, I give adjectives and adverbs category triples with their second number set to 0. This allows adjuncts to adjoin to them, starting at the bottom of that hierarchy. Functional heads, on the other hand, will start with their second number equal to their first number. This means that when they Merge, the resulting phrase is at the right level in the hierarchy, preventing low adjuncts from adjoining after the merger of a high functional head.[12]

[11] This is possible only because Adjoin and Merge are separate operations, as they are in [7]. A close look at the definitions of Merge and Adjoin reveals that there is nothing formally stopping Adjoin from being a case of Merge, one defined when both phrases display a category feature. I have chosen to keep them as separate operations so that Adjoin may have different properties from Merge, such as island effects, and to maintain a certain type of locality for Merge, discussed in section 7.

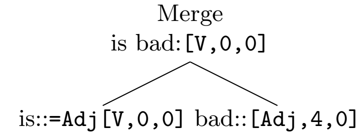
[12] There is nothing in this formalism that prevents adjunction to a functional head. If the function **Ad** assigns adjuncts to a functional head, then it has adjuncts. They

For example, in Figure 5, *very* adjoins to *often*, which is possible since the second number of *often* is 0. Later, functional head T Merges to the VP. Its second number is 25. This is important because we want to say that *apparently* can only adjoin here because its first number is 26, which is higher than 25. A low adverb such as *again*::`[Adv,3,0]` cannot adjoin to T.

6.4 Properties

Let us consider the desiderata laid out in section 4.

1. **Optionality**: ✓the original category is kept as the first element of the category triple
2. **Transparency to selection**: ✓the original category is kept
3. **Order**: ✓The third element of the category is the level of the last adjunct adjoined. The Adjoin rule requires that the adjunct be higher in the order than that third element of the category triple.
4. **Selectability** ✓Adjuncts have regular categories.



   (a) **Efficiency:** ✓Many adjuncts have the same category, so they have the same adjuncts. For example, **Ad**(`Adj`) = {`Adv,Int`}

5. **Adjuncts of adjuncts** ✓Adjunct categories are ordinary categories so they can have adjuncts too (Figure 5).

   (a) **Efficiency:**✓Many adjuncts have the same category, so they are selected by the same LI. For example, in the derivation of *is bad* above, *is* selects anything of category `Adj`.

6. **Unordered** ✓See section 6.5 below.
7. **Obligatory adjuncts: Maybe.** See Section 6.6
8. **Islands** ✓Since Adjoin is a separate operation, it can be defined so that there is no case for adjuncts with movers.

6.5 Unordered Adjuncts

As it stands, adjuncts such as PPs can be modelled as adjuncts, but they must all adjoin at the same level of the hierarchy, or else be cross-classified for each level of the hierarchy you want them to adjoin at. The former allows them to be freely ordered with respect to each other; the latter gives them freedom with respect to all adjuncts.

just behave a little oddly: e.g. `[F,3,3]` requires adjuncts above level 3. Note also a shortcoming in the present model: while Merge of a high functional head will prevent later adjunction of a low adverb, nothing prevents a low functional head that selects, say, `V`, from merging after the adjunction of a high adverb.

An expansion of this model[13] could add a non-number to the set of possible indicies, call it $\emptyset$, and Adjoin could be defined to disregard the hierarchy and asymmetrically check the features for $\emptyset$-indexed adjuncts. Any distinct index also opens the door to adjoining on a different side of the head than other adjuncts; the definition I will give here models Engish PPs, which are post-head, unlike adjectives and many adverbs.

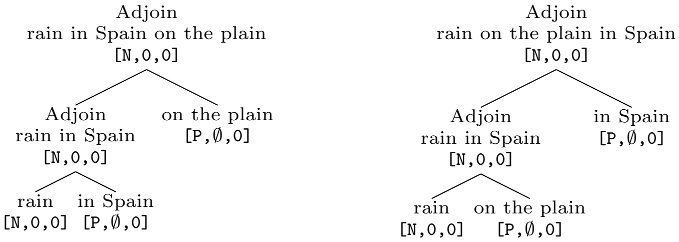


Fig. 7. Unordered English PPs

In definition 7, the first and third cases are for adjuncts with number indicies, and the second and fourth are for adjuncts with $\emptyset$ indicies.

Definition 7 (Adjoin 2). *Let $s, t \in \Sigma$ be strings, $\mathtt{Y,X} \in$ **sel** be categories, $i, j, n, m \in \mathbb{N}$, $mvrs \in (\Sigma^* \times F)^*$ be a mover list, and $\alpha, \beta \in F^*$.*

$$\mathbf{Adjoin}(\langle s, [\mathtt{X},\mathtt{i},\mathtt{j}]\alpha\rangle, \langle t, [\mathtt{Y},\mathtt{m},\mathtt{n}]\beta\rangle :: mvrs)$$
$$= \begin{cases} \langle ts, [\mathtt{X},\mathtt{i},\mathtt{m}]\alpha\rangle :: mvrs & \textit{if}\ \mathtt{m} \geq \mathtt{j}\ \textit{\&}\ \mathtt{Y} \in \mathbf{Ad}(\mathtt{X})\ \textit{\&}\ \beta = \epsilon \\ \langle st, [\mathtt{X},\mathtt{i},\mathtt{j}]\alpha\rangle :: mvrs & \textit{if}\ \mathtt{m} = \emptyset\ \textit{\&}\ \mathtt{Y} \in \mathbf{Ad}(\mathtt{X})\ \textit{\&}\ \beta = \epsilon \\ \langle s, [\mathtt{X},\mathtt{i},\mathtt{m}]\alpha\rangle :: \langle t, \beta\rangle :: mvrs & \textit{if}\ \mathtt{m} \geq \mathtt{j}\ \textit{\&}\ \mathtt{Y} \in \mathbf{Ad}(\mathtt{X})\ \textit{\&}\ \beta \neq \epsilon \\ \langle s, [\mathtt{X},\mathtt{i},\mathtt{j}]\alpha\rangle :: \langle t, \beta\rangle :: mvrs & \textit{if}\ \mathtt{m} = \emptyset\ \textit{\&}\ \mathtt{Y} \in \mathbf{Ad}(\mathtt{X})\ \textit{\&}\ \beta \neq \epsilon \end{cases}$$

6.6 Obligatory Adjuncts

Recall that some elements which really seem to be adjuncts are not optional, for example *He makes a *(good) father*. In MGAs there is a featural difference between nouns that have been modified and nouns that have not. For example, *father* is of category `[N,0,0]` and *good father* has category `[N,0,4]`. **Merge** is defined to ignore everything but the first element, `N`. However, the architecture is available to let **Merge** look at the whole category triple, by way of a positive selectional feature of the form `=[N,_,1]`, which selects anything of category `[N,i,j]` with $\mathtt{j} \geq \mathtt{1}$.

Definition 8 (Merge 2). *For α, β sequences of negative* ***lic*** *feature;, s, t strings; $\mathtt{X} \in$* **sel**; $\mathtt{i},\mathtt{j},\mathtt{m} \in \mathbb{N}$; $\mathbf{C} = \mathtt{X}$ *or* $\mathbf{C} = [\mathtt{X}, \_, \mathtt{m}]$ *& $\mathtt{j} \geq \mathtt{m}$:*

[13] I thank an anonymous reviewer for this suggestion.

$$\boldsymbol{Merge}(\langle s,\, \texttt{=}\mathbf{C}\alpha\rangle \; ::mvrs_s,\, \langle t, [\texttt{X},\texttt{i},\texttt{j}]\beta\rangle ::mvrs_t) = \begin{cases} \langle ts, \alpha\rangle :: mvrs_s \cdot mvrs_t & \textit{if } \beta = \epsilon \\ \langle s, \alpha\rangle :: \langle t, \beta\rangle :: mvrs_s \cdot mvrs_t & \textit{if } \beta \neq \epsilon \end{cases}$$

However, such an expansion of the definition of Merge is not of immediate help in all cases. In the case of *He makes a good father*, the NP *good father* is selected by D before the resulting DP is selected by *makes*, which is the verb that cares about whether the noun is modified. One solution is to cross-list *a* with a new determiner category only for modified NPs, and let *makes* select that category, as in Figure 8.

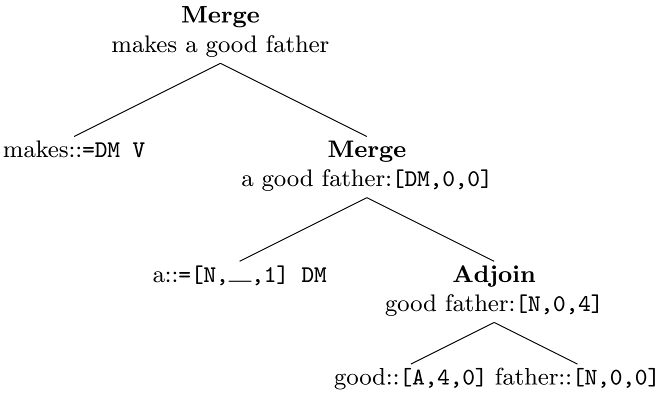


Fig. 8. Determiners of modified NPs could have their own category DM

Obligatory adjuncts are not the only reason to suspect that the tighter relationship is between the verb and the noun, not the verb and the determiner; i.e. that V should perhaps select N, not D. For one, it is well known that in terms of semantics, verbs select nouns. For example, *The man slept* makes sense, but *The table slept* does not, because men are the kinds of things that sleep and tables are not. Both DPs are headed by *the*, which does not carry the animacy information that the noun does. Another piece of evidence comes from noun incorporation. When a head is incorporated into a verb, normally it is the head that the noun selects that is incorporated, as in (13).

(13) a. He [**stab**bed me [$_{\text{PP}}$ in [$_{\text{DP}}$ the [$_{\text{N}}$ **back**]]]]
b. back-stabbing
c. *back-in, *back-the, *back-the-stabbing, *back-in-the-stabbing

[13] proposes that verbs select NPs, and the NPs move to their Ds, which are functional heads on the spine.

For example we might have something like the partial derivation in Fig 9.

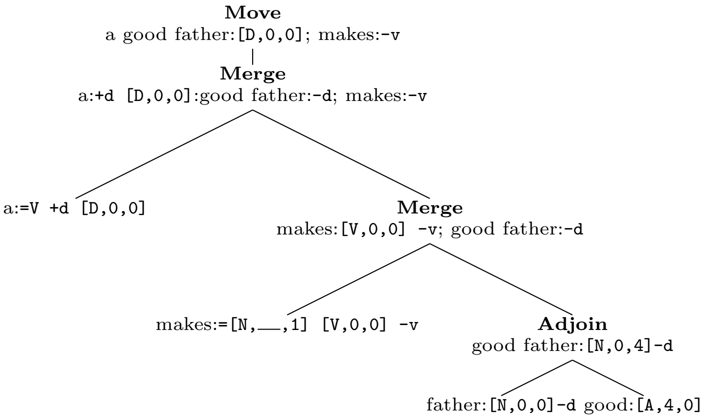

Move
a good father:[D,0,0]; makes:-v
Merge
a:+d [D,0,0]:good father:-d; makes:-v
a:=V +d [D,0,0]
Merge
makes:[V,0,0] -v; good father:-d
makes:=[N,__,1] [V,0,0] -v
Adjoin
good father:[N,0,4]-d
father:[N,0,0]-d
good:[A,4,0]

Fig. 9. Directly selecting N; moving NP up to functional projection D

7 Formal Properties

MGAs are clearly not strongly equivalent to traditional MGs, if we take *strong equivalence* to mean that the set of derivations trees are the same. This is of course is impossible since MGAs have an extra function, Adjoin. MGAs are, on the other hand, weakly equivalent to MGs, meaning that for every MGA, an MG can be defined that generates the same strings, and vice versa.

Lemma 1. *$L(MG) \subseteq L(MGA)$*

Proof. MGAs also include Merge and Move, and place no additional restrictions on their action. Any MGA language could have Adjoin stripped away and what remained would be an MG.

Lemma 2. *L(MGA) $\subseteq$ L(MG)*

Proof. MGs are weakly (and indeed strongly) equivalent to Multiple Context Free Grammars (MCFGs) so it suffices to show that $L(MGA) \subseteq L(MCFG)$.

We translate an MGA into an MCFG is the normal way, following [9]: the nonterminals of the MCFG are sequences of feature sequences from the MGA. This translation is based on the basic grammar given in Definition 6, but it is easy to see how it could be expanded to include the extentions suggested in later sections.

Given MGA $G = \langle \Sigma, F = \mathbf{sel} \cup \mathbf{lic}, Lex, M, S, Ad$, define an MCFG $MCFG(G) = \langle \Sigma, N, P, S \rangle$ defining the language

$N = \{\langle \delta_0, \delta_1, ..., \delta_j \rangle | 0 \leq j \leq |\mathbf{lic}|$, all $\delta_i \in$suffix$(Lex)\}$

Let $h =$Max$(\{i | \exists \mathtt{X} \in \mathbf{sel} : i = |\mathbf{Ad}(\mathtt{X})\}$

The rules P are defined as follows, $\forall \alpha, \beta, \delta_0, ..., \delta_i, \gamma_0, ..., \gamma_j \in$ suffix(Lex). $s_0, ..., s_i, t_0, ..., t_j$ are variables over strings.

Lexical rules: $\alpha(s)$ $\forall \langle s, \alpha \rangle \in Lex$

Merge-and-stay rules: Here is the first case of the Merge rule for MGAs.

$$\textbf{Merge}(\langle s,\ \texttt{=X}\alpha\rangle\ ::\text{mvrs}_s,\ \langle t, [\texttt{X},\texttt{m},\texttt{n}]\rangle::\text{mvrs}_t) = \langle st, \alpha\rangle :: \text{mvrs}_s \cdot \text{mvrs}_t$$

It becomes a set of MCFG rules as follows. In the rule set below, $s = s_0, t = t_0$,the tree parts of mvrs$_s$ and mvrs$_t$ are $s_1, ..., s_i$ and $t_1, ..., t_j$ respectively, and their features become $\delta_1, ..., \delta_i$ and $\gamma_1, ...\gamma_j$. One rule is made for each index less than the maximum possible index h for the grammar. (Any rule indicies that fall outside the set of indicies for that particular category simply go unused in practice.)
Here is the description of the MCFG rules corresponding to this Merge rule:

$\langle\alpha, \delta_1, ..., \delta_i, \gamma_1, ...\gamma_j\rangle(s_0t_0, s_1, ..., s_i, t_1, ..., t_j)$
:- $\langle = \texttt{X}\alpha, \delta_1, ..., \delta_i\rangle(s_0, ..., s_i)\ \langle[\texttt{X},\texttt{m},\texttt{n}], \gamma_1, ...\gamma_j\rangle(t_0, ..., t_j)$
$\forall\texttt{X} \in \textbf{sel}, \forall\texttt{n},\texttt{m} \leq h$

The rest of the MCFG rules are formed similarly.

Merge-and-move rules: $\forall\texttt{X} \in \textbf{sel}, \forall\texttt{n},\texttt{m} \leq h$
$\langle\alpha, \beta, \delta_1, ..., \delta_i, \gamma_1, ...\gamma_j\rangle(s_0, t_0, s_1, ..., s_i, t_1, ..., t_j)$
:- $\langle = \texttt{X}\alpha, \delta_1, ..., \delta_i\rangle(s_0, ..., s_i)\ \langle[\texttt{X},\texttt{m},\texttt{n}]\beta, \gamma_1, ...\gamma_j\rangle(t_0, ..., t_j)$

Adjoin-and-stay rules: $\forall\texttt{X},\texttt{Y} \in \textbf{sel}\ \ s.t.\ \texttt{Y} \in \textbf{Ad}(\texttt{X}), \forall\texttt{k},\texttt{l},\texttt{n},\texttt{m} \leq h\ \ s.t.\ \texttt{n} \geq \texttt{k}$
$\langle[\texttt{X},\texttt{m},\texttt{n}], \delta_1, ..., \delta_i, \gamma_1, ...\gamma_j\rangle(s_0t_0, s_1, ..., s_i, t_1, ..., t_j)$
:- $\langle[\texttt{X},\texttt{m},\texttt{k}], \delta_1, ..., \delta_i\rangle(s_0, ..., s_i)\ \langle[\texttt{Y},\texttt{n},\texttt{l}], \gamma_1, ...\gamma_j\rangle(t_0, ..., t_j)$

Adjoin-and-move rules: $\forall\texttt{X},\texttt{Y} \in \textbf{sel}\ \ s.t.\ \texttt{Y} \in \textbf{Ad}(\texttt{X}), \forall\texttt{k},\texttt{l},\texttt{n},\texttt{m} \leq h\ \ s.t.\ \texttt{n} \geq \texttt{k}$
$\langle[\texttt{X},\texttt{m},\texttt{n}], \beta, \delta_1, ..., \delta_i, \gamma_1, ...\gamma_j\rangle(s_0, t_0, s_1, ..., s_i, t_1, ..., t_j)$
:- $\langle[\texttt{X},\texttt{m},\texttt{k}], \delta_1, ..., \delta_i\rangle(s_0, ..., s_i)\ \langle[\texttt{Y},\texttt{n},\texttt{l}]\beta, \gamma_1, ...\gamma_j\rangle(t_0, ..., t_j)$

Move-and-stop rules: $\forall\texttt{f} \in \textbf{lic}$
$\langle\alpha, \delta_1, ..., \delta_{i-1}, \delta_{i+1}, ..., \delta_j\rangle(s_is_0, s_1, ..., s_{i-1}, s_{i+1}, ..., s_j)$
:- $\langle +\texttt{f}\alpha, \delta_1, ..., \delta_{i-1}, -\texttt{f}, \delta_{i+1}, ..., \delta_j\rangle(s_0, ..., s_j)$

Move-and-keep-moving rules: $\forall\texttt{f} \in \textbf{lic}$
$\langle\alpha, \delta_1, ..., \delta_{i-1}, \beta, \delta_{i+1}, ..., \delta_j\rangle(s_0, ..., s_j)$
:- $\langle +\texttt{f}\alpha, \delta_1, ..., \delta_{i-1}, -\texttt{f}\beta, \delta_{i+1}, ..., \delta_j\rangle(s_0, ..., s_j)$

These rule sets are finite since MGAs never add anything to feature sequences, but only either remove features or change just the indicies of `[X,i,j]` features. As such, the suffixes $\alpha, \beta, \delta, \gamma$ are limited in number. Since any given grammar has a maximal hierarchy depth h, the indicies `k,l,m,n` in the rules are defined to be limited by h.

Theorem 1 (Weak equivalence of MGAs and MGs). *For any MGA* $G = \langle\Sigma, \textbf{sel}, \textbf{lic}, \textbf{Ad}, \mathit{Lex}, \{\textbf{Merge, Move, Adjoin}\}\rangle$, *there is a weakly equivalent MG* $G' = \langle\Sigma, \textbf{sel}_{MG}, \textbf{lic}, \mathit{Lex}_{MG}, \{\textbf{Merge, Move}\}\rangle$.

Proof. By lemmas 1 and 2

References

1. Adger, D.: A minimalist theory of feature structure. Ms..¡ h ttp://ling. auf. net/lingBuzz/000583, as of 30, 06–08 (2007)
2. Bernardi, R., Szabolcsi, A.: Optionality, scope, and licensing: An application of partially ordered categories. Journal of Logic, Language and Information 17(3), 237–283 (2008)
3. Chomsky, N.: The Minimalist Program. MIT Press, Cambridge, MA (1995)
4. Cinque, G.: Adverbs and functional heads: a cross-linguistic perspective. Oxford studies in comparative syntax, Oxford University Press, Oxford (1999)
5. Cinque, G.: The syntax of adjectives: a comparative study. Linguistic Inquiry monographs, MIT Press, Cambridge, MA (2010)
6. Fowlie, M.: Order and optionality: Minimalist grammars with adjunction. In: The 13th Meeting on the Mathematics of Language. p. 12 (2013)
7. Frey, W., Gärtner, H.M.: On the treatment of scrambling and adjunction in minimalist grammars. In: Proceedings of the Conference on Formal Grammar (FGTrento). pp. 41–52. Trento (2002)
8. Graf, T.: The price of freedom: Why adjuncts are islands. Slides of a talk given at the Deutsche Gesellschaft für Sprachwissenschaft pp. 12–15 (2013)
9. Harkema, H.: A characterization of minimalist languages. In: Logical aspects of computational linguistics, pp. 193–211. Springer (2001)
10. Keenan, E.L., Stabler, E.P.: Bare Grammar. CSLI Publications, Stanford (2003)
11. Kobele, G.M., Retoré, C., Salvati, S.: An automata-theoretic approach to minimalism. In: Rogers, J., Kepser, S. (eds.) Model Theoretic Syntax at ESSLLI '07. ESSLLI (2007)
12. Rizzi, L.: Locality and left periphery. In: Belletti, A. (ed.) Structures and Beyond: The Cartography of Syntactic Structures, vol. 3, pp. 223–251. Oxford University Press, Oxford (2004)
13. Sportiche, D.: Division of labor between merge and move: Strict locality of selection and apparent reconstruction paradoxes. LingBuzz (2005)
14. Stabler, E.: Derivational minimalism. Logical Aspects of Computational Linguistics pp. 68–95 (1997)