Notes on the Complexity of Complex Heads in a Minimalist Grammar

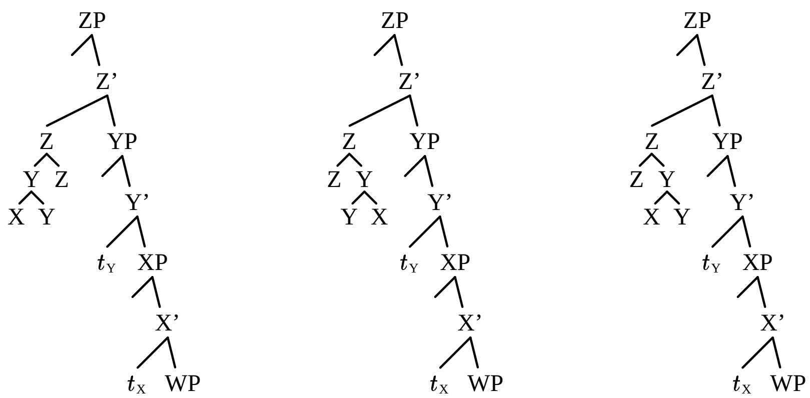
Jens Michaelis
Universität Potsdam

Abstract

The type of a *minimalist grammar (MG)* introduced in Stabler (1997) provides a simple algebraic formalization of the perspectives as they arise from Chomsky (1995b) within the linguistic framework of transformational grammar. As known (cf. Michaelis 2001a; 2001b; Harkema, 2001), this MG–type defines the same class of derivable string languages as, e.g., *linear context–free (string) rewriting systems (LCFRSs)* (Vijay–Shanker, Weir and Joshi, 1987; Weir, 1988). In this paper we show that, in terms of weak equivalence, the subclass of MGs which allow *(overt) head movement* but no *phrasal movement* in the sense of Stabler (1997), constitutes a proper subclass of *linear indexed grammars (LIGs)*. and thus *tree adjoining grammars (TAGs)*. We also examine the "inner hierarchic complexity" of this embedding in some more detail by looking at the subclasses canonically resulting from a differentiation between left adjunction of the moved head to the attracting one, and right adjunction of this kind. Furthermore, we show that adding the possibility of phrasal movement by allowing just one "indistinguishable" licensee to trigger such movement has no effect on the weak generative capacity of the corresponding MG–subclasses. On the other hand however, MGs which do not employ head movement but whose licensee set consists of at most two elements, are shown to derive, a.o., languages not derivable by any LIG. In this sense our results contribute to sheding some light on the complexity as it arises from the interplay of two different structural transformation types whose common existence is often argued to be linguistically motivated.

1. Introduction

The type of a *minimalist grammar (MG)* introduced in Stabler (1997) provides a simple algebraic formalization of the perspectives as they arise from Chomsky (1995b) within the linguistic framework of a principles and parameter–approach to transformational grammar. As known (cf. Michaelis 2001a; 2001b; Harkema, 2001), this MG–type constitutes a *mildly context–sensitive formalism* in the sense that it defines the same class of derivable string languages as *linear context–free (string) rewriting systems (LCFRSs)* (Vijay–Shanker, Weir and Joshi, 1987; Weir, 1988).[1] In particular, the MG–definition permits a type of *(overt) head movement* which rather directly reflects the derivation mode of *(successive) head(–to–head) adjunction*—which, in the minimalist approach, takes place due to the necessity of feature checking—(successively) creating more complex heads (cf. Figure 1). Nev-



. . . successive left head adjunction . . . successive right head adjunction . . . (mixed) successive head adjunction

Figure 1: Complex head "Z" [2] resulting from . . .

ertheless, there is a further notable property of MGs which—in connection with Michaelis (2001a)—follows from Harkema (2001) as well as Michaelis (2001b): each MG can be transformed into a weakly equivalent MG that

* This work has been funded by DFG–grant STA 519/1–1.

1. Hence, MGs as defined in Stabler (1997) join to a series of weakly equivalent formalism classes among which, beside LCFRSs, there is, e.g., the class of set–local *multicomponent tree adjoining grammars (MCTAGs)* (cf. Weir, 1988). For a list of some further of such classes of generating devices, beside MCTAGs, see e.g. Rambow and Satta (1999).

2. In terms of an X-Bar representation.

neither employs head movement nor *covert (phrasal) movement* as allowed by the general MG–definition.[3] In fact it is this MG–subtype which, e.g., is covered in terms of the succinct MG–reformulation in Stabler and Keenan (2000) (reducing MGs to their "bare essentials"), and which the MG–recognizer in Harkema (2000) (working in polynomial time depending on the length of the input string) is defined for. Moreover, the "MG–internal" equivalence result can be seen as providing some technical support to more recent linguistic work which, in particular, tries to completely dispense with any type of movement different from overt phrasal movement (e.g. Koopman and Szabolcsi, 2000; Mahajan, 2000).

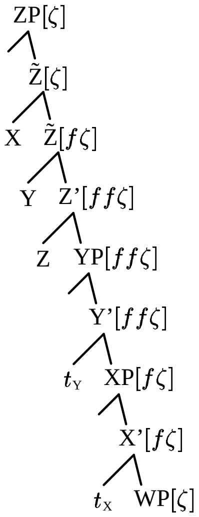


Figure 2: Complex head "Z" resulting from successive left head adjunction as representable in an RLIG.

Many linguists working within the transformational tradition, however, believe head movement to be indispensable for an adequate explanation of natural language syntax.[4] How the kind of head movement originally allowed in an MG can be (re)integrated into the succinct MG–definition is shown in Stabler (2001). As indicated in there, the recognition complexity w.r.t. such an MG and an input string increases—adapting the methods of Harkema (2000)—at most as much as in the case when adding two new distinct *licensees*, i.e. two new distinct features potentially triggering phrasal movement, to the grammar.

Concentrating on questions concerning the increase of generative complexity, we show in this paper that, in terms of derivable string languages, the subclass of MGs allowing head movement but no phrasal movement in the sense of Stabler (1997), constitutes a proper subclass of LIGs, and thus TAGs. This is done by embedding MGs weakly equivalently into *extended left LIGs (ELLIGs)* in the sense of Michaelis and Wartena (1999).[5] Examining the "inner hierarchic complexity" of this embedding in some more detail, it can be shown that MGs which allow only left head adjunction define the same class of derivable languages as RLIGs,[6] and thus *context free grammars (CFGs)* (cf. Michaelis and Wartena, 1999). MGs which allow only right head adjunction derive more languages then CFGs, and MGs allowing right as well as left head adjunction seem to provide a further proper extension. Furthermore, adding the possibility of phrasal movement by allowing the MG's licensee set to consist of at most one feature[7] has no effect on the weak generative capacity of the considered MG–subclasses. On the other hand, it can be shown that MGs which do not employ head movement but whose licensee set consists of at most two elements, derive, a.o., languages not derivable by any LIG.[8]

The presented results are of interest in at least two respects: first, they contribute in a more general sense to one of the central issues mathematical approaches to linguistics are concerned with, namely, how much strong generative capacity can be squeezed out of a formalism without increasing the weak generative power.[9] Second, since the presented results provide a first narrow step towards an answer to the question of how the specific types of head movement and phrasal movement as defined in MGs are related to each other in terms of generative capacity, they may not only be a promising starting point, when seeking for a lower upper bound on the parsing complexity of MGs, but also shed some light on the structural complexity as it arises from the interplay of two different structural transformation types whose common existence is often argued to be linguistically motivated.

For illustrative purposes we demonstrate how MGs allowing head movement but no phrasal movement can be weakly equivalently embedded into a subclass of TAGs, instead of LIGs, which in its turn is weakly equivalent

3. The only movement possibly used is *overt phrasal movement*.
4. Even current accounts which argue in favour of overt vs. covert movement do not completely exclude overt head movement (e.g. Kayne 1998; 1999).
5. In Michaelis and Wartena (1999), ELLIGs were defined as the formal counterpart of *extended right LIGs (ERLIGs)*. An ELLIG (respectively, ERLIG) is an LIG, G, such that for each nonterminating production r, the distinguished symbol on the righthand side (rhs) is the leftmost (respectively, rightmost) nonterminal, i.e., r is of the form $A[\zeta \ldots] \to wB[\eta \ldots]\alpha$ (respectively, $r = A[\zeta \ldots] \to \alpha B[\eta \ldots]w$), where w is a string of terminals. Thus, applying such an r to a corresponding object $A[\zeta\theta]$, the stack associated with the nonterminal A is passed on to the leftmost (respectively, rightmost) nonterminal child replacing ζ, the upper part of the stack, by η. If, in addition, for each such nonterminating rule r, no terminal symbol appears to the left (respectively, right) of the distinguished nonterminal symbol of the rhs, i.e., if w is always the empty string, then G is simply referred to as an *LLIG (respectively, RLIG)*.
6. A corresponding weakly equivalent RLIG can be defined such that it represents complex heads, created by successive head adjunction, as indicated in Figure 2.
7. Thus, at most one "indistinguishable" type of phrasal movement is available.
8. A schematic overview is given in Figure 3.
9. See e.g. Joshi (2000) and references therein for a more recent discussion.

$$
\begin{array}{rl}
 & \text{2-ML}^{\text{HM:none}} \\
\{a_1^n b_1^n a_2^m b_2^m c_1^n d_1^n c_2^m d_2^m \mid m,n \geq 0\} & \in \quad \not\in \\
\text{Seki } et\ al.\text{, 1991} \longrightarrow & \text{LIL} = \text{TAL} \\
\{a_1^n a_2^m b_2^m c_2^m b_1^n c_1^n d_1^n \mid m,n \geq 0\} & \in \quad \supsetneq \quad \not\in \\
\text{Wartena, 1999} \longrightarrow & \text{ELLIL} \\
 & \supseteq \\
 & \text{0-ML}^{\text{HM:arbit-adj}} = \text{1-ML}^{\text{HM:arbit-adj}} \\
\{a^n b^n d^n \mid n \geq 0\} & \in \quad \supseteq \quad \not\in \\
\text{[conjecture]} \longrightarrow & \text{0-ML}^{\text{HM:right-adj}} = \text{1-ML}^{\text{HM:right-adj}} \\
\text{Cornell, 1996; Stabler, 1997} \longrightarrow \{ww \mid w \in \{a,b\}^*\} & \in \quad \supsetneq \quad \not\in \\
 & \text{0-ML}^{\text{HM:left-adj}} = \text{1-ML}^{\text{HM:left-adj}} \\
 & \| \\
 & \text{0-ML}^{\text{HM:none}} = \text{1-ML}^{\text{HM:none}} \\
 & \| \\
 & \text{CFL} = \text{RLIL}
\end{array}
$$

Figure 3: Schematic overview of our results.[10,11]

to ELLIGs (Section 3). Largely skipping formal details afterwards, we subsequently emphasize the crucial points concerning the hierarchy of the corresponding MG–subclasses resulting from the different types of head movement as available in the MG–formalism (Section 3.1–3.4). Then, we turn to simple phrasal movement as allowed by the MG–definition (Section 4) and, finally, present an example of an MG which does not employ any head movement, but phrasal movement "slightly beyond" the simple type, thereby deriving a language not derivable by any LIG (Section 5). But first, since the reader might be less familiar with MGs, we briefly introduce them in their aspects relevant here.

2. Minimalist Grammars

An MG is a formal device which specifies a countable set of *expressions (over $\mathcal{S} \cup \mathcal{P} \cup \mathcal{I}$)*,[12] i.e., a countable set of finite, binary (ordered) trees each equipped with a leaf–labeling function assigning a string from $\mathcal{S}^*\mathcal{P}^*\mathcal{I}^*$ to each leaf, and with an additional binary relation, the asymmetric relation of *(immediate) projection*, defined on the set of pairs of siblings (cf. Figure 4).

A *maximal projection* within such an expression τ is a subtree of τ which is either identical to τ, or its root is projected over by its root's sibling. The *specifiers*, the *complement* and the *head* of (a maximal projection in) τ are determined in the canonical way by means of the projection relation (cf. Figure 5). τ is *complete* if its head–label is in $\{\mathsf{c}\}\mathcal{P}^*\mathcal{I}^*$, and each other leaf–label in $\mathcal{P}^*\mathcal{I}^*$. The *phonetic yield* of τ is the string which results from concatenating the leaf–labels in "left–to–right–manner" ignoring all instances of non–phonetic features.

The base of an MG, G, is formed by a *lexicon* (a finite set of simple expressions, i.e. single node trees in the above sense, also called *heads*) and two structure building functions: *merge* (combining two trees) and *move*

10. Here, for $n \geq 0$ and $x \in \{\text{none}, \text{left-adj}, \text{right-adj}, \text{arbit-adj}\}$, $n\text{-ML}^{\text{HM:}x}$ denotes the class of all languages derivable by any MG whose licensee set consists of at most n elements, and which permits only head(–to–head) adjunction of the type indicated by x.

11. Note that Wartena (1999) actually provides a formal proof of the fact that the language $\{a_1^n b_1^n c_1^n a_2^m b_2^m c_2^m d_1^n \mid m,n \geq 0\}$, although derivable by some LIG, is not derivable by any ERLIG. For reasons of symmetry however, it becomes immediately clear from the corresponding proof details that the language $\{a_1^n a_2^m b_2^m c_2^m b_1^n c_1^n d_1^n \mid m,n \geq 0\}$, although derivable by some LIG, is not derivable by any ELLIG.

12. $\mathcal{S}$, $\mathcal{P}$ and $\mathcal{I}$ are assumed to be pairwise disjoint sets, namely, a set of *syntactic*, *phonetic* and *interpretable features*, respectively. $\mathcal{S}$ is partitioned into *basic categories*, *selectors*, *licensees*, and *licensors*. There is at least the basic category c.

13. Here, "<" (respectively, ">") as "label" of a non–leaf node means "my left (respectively, right) child projects over my right (respectively, left) child."

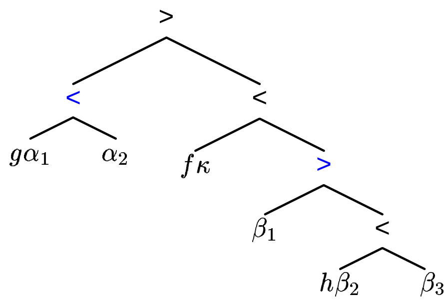


Figure 4: A typical (minimalist) expression.[13]

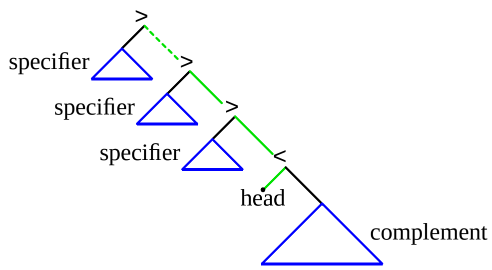


Figure 5: The typical (minimalist) expression structure.

(transforming a single tree). Both functions build structure by canceling two particular matching instances of syntactic features within the leaf–labels of the trees to which they are applied. The closure of the lexicon under these two functions is the set of trees specified by G. The *(string) language* derivable by G is a particular subset of $\mathcal{P}^*$, namely, the set of the phonetic yields of the complete expressions within this closure.

The function *merge* is applicable to $\langle \upsilon, \phi \rangle$, a pair of expressions, if ϕ's head–label starts with an instance of some basic category x, and υ's head–label with an instance of $^{=}\mathsf{x}$, the corresponding *weak* selector of x. Depending on whether υ is simple or not, ϕ is selected as the complement or the highest specifier, respectively. Within the resulting tree, *merge*(υ, ϕ), the corresponding instances of $^{=}\mathsf{x}$ and x are cancelled (cf. Figure 6). In

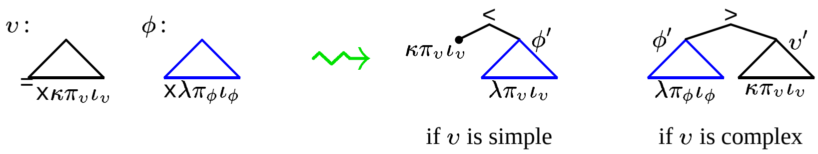


Figure 6: *merge*(υ, ϕ) — weak selection.

case υ is a head, its label may likewise start with an instance of a corresponding *strong* selector of x, $^{=}\mathsf{X}$ or $\mathsf{X}^{=}$, both additionally triggering *(overt) head movement*, i.e., *merge*(υ, ϕ) is defined as before, but in addition π_ϕ, the string of phonetic head–features of the selected ϕ, is incorporated into the label of the selecting head υ, either immediately to the right (triggered by $^{=}\mathsf{X}$) or immediately to the left (triggered by $\mathsf{X}^{=}$) of π_υ, the string of phonetic features within υ's (unique) label (cf. Figure 7).[14,15]

The function *move* is applicable to an expression υ, if there is exactly one maximal projection ϕ in υ whose head–label starts with in instance of some licensee $\text{-}\mathsf{x}$ such that υ's head–label starts with an instance of a cor-

14. In the minimalist approach suggested in Chomsky (1995b) the merge–operator applies freely, and head movement is a separate step following a corresponding application of this operator. As noted by Chomsky (1995a, p. 327), in a strong sense this can be seen as a violation of the "extension condition" on structure building functions. Stabler (1998, p. 78, fn. 5) argues that that the implementation of head movement within MGs not only avoids such a violation, but "it [also] explains the coincidence of the selection and head movement configurations." Note also that the implementation of head movement is in accordance with the *head movement constraint*, demanding that a moving head can never pass over the closest c–commanding head. To put it differently, whenever we are concerned with a case of successive head movement, i.e. recursive adjunction of a (complex) head to a higher head, it obeys *strict cyclicity*. The way in which MGs reflect the "usual" linguistic notion of head adjunction arising from head movement is made explicit in Stabler (1998).

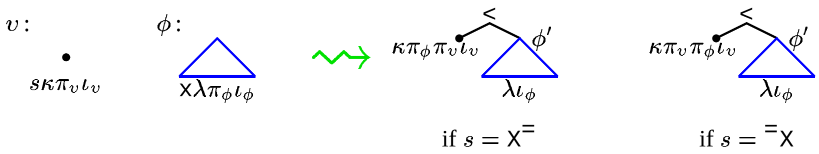


Figure 7: *merge*(υ, ϕ) — strong selection.

responding *strong* licensor $\mathtt{+X}$ or *weak* licensor $\mathtt{+x}$ triggering *overt* or *covert phrasal movement*, respectively.[16] If υ's head–label starts with a strong licensor then, within the resulting tree *move*(υ), ϕ is moved into the new created, highest specifier position, while the triggering instances of $\mathtt{+X}$ and $\mathtt{-x}$ are cancelled, and the "original" position of ϕ's root becomes a single node labeled with the empty string (cf. Figure 8). If υ's head–label starts with

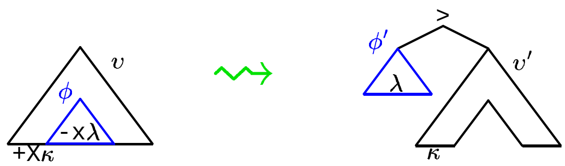


Figure 8: *move*(υ) — overt phrasal movement.

a weak licensor then, within the resulting tree *move*(υ), the triggering instance of $\mathtt{+x}$ is cancelled, while a copy of ϕ in which the triggering instance of $\mathtt{-x}$ as well as all instances of phonetic features are cancelled, is moved into the new created, highest specifier position, and while another copy of ϕ in which all instances of non–phonetic features are cancelled is "left behind."[17]

3. MG–Head Movement in Terms of TAGs

Let G be an MG which allows head movement but no phrasal movement.[18] A nonterminal in our weakly equivalent TAG, G', is either the start symbol, S, or a pair $\langle \mathtt{y}, t \rangle$ with $\mathtt{y}$ being a basic category from G, and with $t \in \{\mathtt{weak}, \mathtt{strong}\}$, where $\mathtt{weak}$ and $\mathtt{strong}$ are two new, distinct symbols.

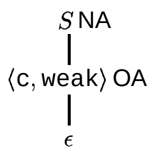


Figure 9: The unique initial tree of G'.

There is a single initial tree (cf. Figure 9), and for each lexical MG–item α, there are two elementary auxiliary trees depending on the form of the (unique) label of α: we generally distinguish the cases $\mathtt{y}\pi\iota$ (cf. Figure 10) and $s\,{}^=\mathtt{x}_1 \cdots {}^=\mathtt{x}_n \mathtt{y}\pi\iota$, s being any selector, ${}^=\mathtt{x}_1, \ldots, {}^=\mathtt{x}_n$ weak selectors for an $n \geq 0$, $\mathtt{y}$ a basic category, $\pi \in \mathcal{P}^*$, and $\iota \in \mathcal{I}^*$. The latter case divides into three subcases depending on whether s is of the form ${}^=\mathtt{x}$, $\mathtt{X}^=$, or ${}^=\mathtt{X}$ (cf. Figure 11–13). Hence, G' only uses auxiliary elementary trees which may be called *extended right auxiliary*, i.e., auxiliary trees in which the foot node is the leftmost nonterminal node on the frontier, and all interior nodes left of the spine are marked for null adjunction.[19,20]

16. The uniqueness of ϕ provides us with a strict version of the *shortest move constraint (SMC)*.
17. For more details on the definition of *merge* and *move* see Stabler (1997). Particular examples of MGs are given below in Section 3–5.
18. That is, G does not employ the function *move* to derive any expression from some instances of the lexical items. Therefore, we may assume that no (label of any) lexical item contains an instance of some licensee or licensor feature.
19. This TAG–subtype may also be seen as a straightforward extension of a particular subtype of a *tree insertion grammar* (Schabes and Waters, 1995).
20. With the intend of simplifying our presentation, G' also fits in with the "classical" TAG–definition allowing selective adjunction, but not substitution (see e.g. Vijay–Shanker and Weir, 1994).

Figure 10: Elementary auxiliary trees of G' resulting from the lexical MG–item $\mathtt{y}\pi\iota$.

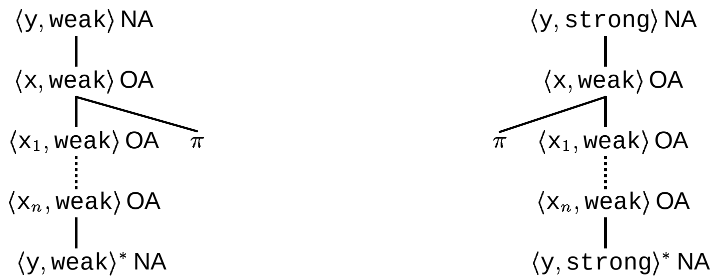


Figure 11: Elementary auxiliary trees of G' resulting from the lexical MG–item $^{=}\mathtt{x}^{=}\mathtt{x}_1\cdots{}^{=}\mathtt{x}_n\mathtt{y}\pi\iota$.

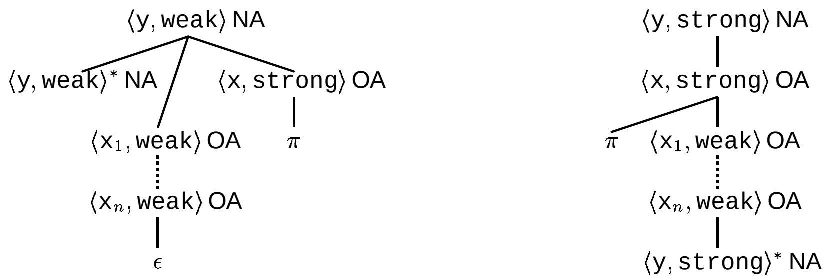


Figure 12: Elementary auxiliary trees of G' resulting from the lexical MG–item $\mathtt{X}^{==}\mathtt{x}_1\cdots{}^{=}\mathtt{x}_n\mathtt{y}\pi\iota$.

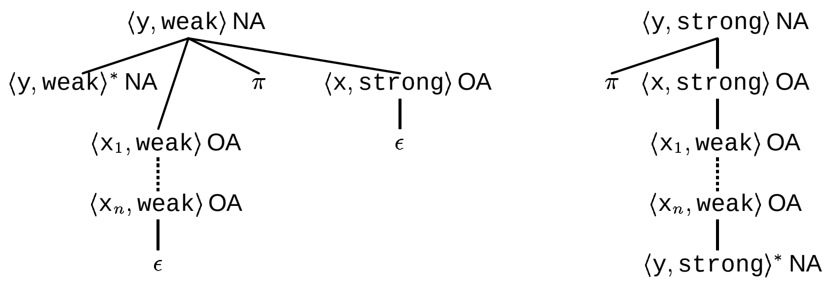


Figure 13: Elementary auxiliary trees of G' resulting from the lexical MG–item $^{=}\mathtt{X}^{=}\mathtt{x}_1\cdots{}^{=}\mathtt{x}_n\mathtt{y}\pi\iota$.

G' simulates the MG–derivation of an expression τ whose head–label starts with a basic category $\mathtt{y}$ by "reversing the top–down order," i.e., the complement becomes the highest constituent, and the specifiers are successively attached top–down in the sense indicated in Figure 14. Such a derivation, indeed, is simulated by G' exactly twice in the two outline ways. Vice versa, each TAG–derivable auxiliary tree T which does not permit (further) adjunction to any of his nodes, corresponds to an MG–derivable expression τ whose head–label starts with a basic category $\mathtt{y}$, in exactly one of the two outlined ways. Thus—ignoring the label of T's foot node—T's yield either equals τ's phonetic yield, or this is true up to the fact that the substring of τ's phonetic yield contributed by τ's head, $\mathit{yield}_{\mathcal{P}}(\mathit{head}_\tau)$, is "shifted" to the front within T's yield. If the latter, it is just $\mathit{yield}_{\mathcal{P}}(\mathit{head}_\tau)$ which constitutes T's yield left of its spine, and in MG–terms, T is connected with the expectation that the represented τ is inevitable selected strongly in a further derivation step (expressed by the second component of the label of T's root node, and thus foot node). Otherwise, T's yield left of its spine is empty, and the represented τ is connected with the demand of being selected weakly in a further derivation step (again coded by means of the second component of the label of T's root/foot node).

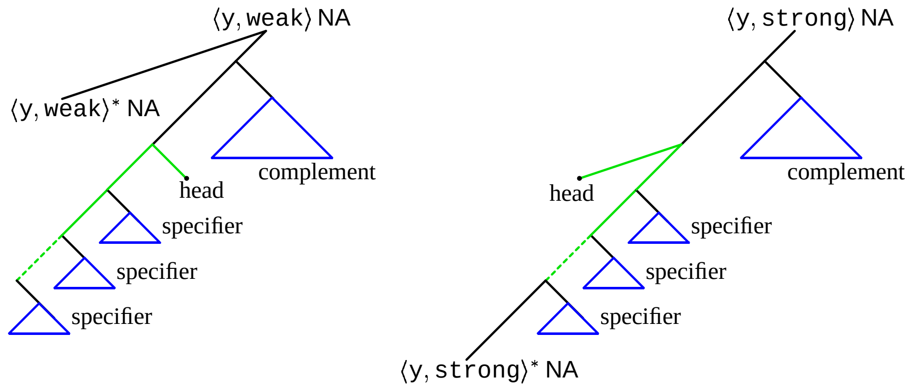


Figure 14: The MG–expression structure simulated by G'.

3.1. Null Head Adjunction

As an immediate consequence of our construction we observe that in case G does not use any strong selectors (i.e., no head movement takes place deriving an expression belonging to the closure of the lexicon of G), only (strictly) right auxiliary trees are effectively used in order to derive a tree in the weakly equivalent TAG G'. Thus, in this case, we are in fact concerned with a particular type of *tree insertion grammar* in the sense of Schabes and Waters (1995) additionally allowing adjunction constraints in the sense of the usual TAG–definition. Since adding the possibility of such constraints to the TIG–type does not increase the weak generative power, and since TIGs are known to constitute a weakly context–free formalism, this yields another proof of the well–known fact that MGs which do neither employ head movement nor phrasal movement only derive context–free languages (CFLs).[21]

3.2. Left Head Adjunction

As mentioned in the introduction, it can be shown that MGs which—beside the simple merge–operation—permit only left head adjunction triggered by a corresponding strong selection feature, do only derive CFLs as well. Skipping any formal details here, we just mention that, as far as a complex head of a corresponding MG is concerned, the dependencies between the (phonetic yields of the) single lexical heads incorporated into the (phonetic yield of the) complex head and their respective “traces” are nested. This allows us to use a single stack in order to “correctly redefine” the concept of successive left head adjunction within the weakly equivalent RLIG.[22]

3.3. Right Head Adjunction

The crucial difference between successive right head adjunction and successive left head adjunction is constituted by the fact that—within a complex head created by the former derivation mode—the dependencies between the (phonetic yields of the) single lexical heads incorporated into the (phonetic yield of the) complex head and their respective “traces” are cross–serial. This kind of dependencies can be made “visible” by means of a respective specifier being attached right beyond each “trace,” and containing some particular phonetic material; e.g., a copy of the lexical phonetic material of the head by which the specifier is selected as in the case of the MG G_{ww} deriving the copy language $\{ww \mid w \in \{a,b\}^*\}$, and consisting of the following 9 lexical items:[23]

$$\mathtt{{}^{=}C\,{}^{=}X}_a\ \mathtt{X}\ a \qquad \mathtt{X}_a\ a \qquad \mathtt{{}^{=}X}_a\ \mathtt{X}\ a \qquad \mathtt{{}^{=}X\ C} \qquad \mathtt{C}$$

$$\mathtt{{}^{=}C\,{}^{=}Y}_b\ \mathtt{Y}\ b \qquad \mathtt{Y}_b\ b \qquad \mathtt{{}^{=}Y}_b\ \mathtt{Y}\ b \qquad \mathtt{{}^{=}Y\ C}$$

21. Vice versa, it is not hard to verify that each CFG is weakly equivalent to some MG of this kind. This can be proven rather straightforwardly, e.g., by starting with a CFG in Chomsky normal form.
22. Note that the type of RLIG needed does use the stack in only the following “normalized” way: once an element has been popped from the stack, the stack has to be emptied before a new element can be pushed onto the stack. This, of course, is just a reflex of the successive cyclicity by which a complex complex head is created.
23. Since lexical items are always simple expression, we will usually identify each such item with its head–label. Note further that, referring to Cornell (1996), the example MG G_{ww} is also given in Stabler (1997).

3.4. Arbitrary Head Adjunction

An MG which derives the non–CFL $\{a^n b^n d^n \mid n \geq 0\}$ by means of mixed successive head adjunction is the MG $G_{a^n b^n d^n}$ from below. We see that, at the same time, while $G_{a^n b^n d^n}$ derives cross–serial dependencies between a's and d's by means of successive right head adjunction analogously to the way exploited by G_{ww}, $G_{a^n b^n d^n}$ additionally derives nested dependencies between a's and b'S as well as between b's and d's. Since these additional nested dependencies are derived by "stepwise intervening" left head adjunction this suggests that a language like $a^n b^n d^n$ is not derivable by an MG which uses only right head adjunction. The MG $G_{a^n b^n d^n}$ consists of the following 6 lexical items:

$\mathsf{X}^{=}\,{}^{=}\mathsf{z}\;\mathsf{y}\;b \qquad \mathsf{x}\;a \qquad \mathsf{z}\;d \qquad {}^{=}\mathsf{Y}\;\mathsf{x}\;a \qquad \mathsf{Y}^{=}\,\mathsf{c} \qquad \mathsf{c}$

4. Simple Phrasal Movement

Assume G_{MG} to be an MG such that each selection feature that occurs within the label of some lexical entry is weak, and such that the set of licensees is a singleton set. Thus, G_{MG} does not allow any kind of head movement, but an "indistinguishable" type of phrasal movement. Again, we will skip formal details, when arguing that the language derivable by G_{MG} is a CFL, and briefly sketch the crucial point here.

Suppose that an expression ϕ is selected by another expression υ, yielding an expression $\tau = \mathit{merge}(\upsilon, \phi)$ such that the head–label of ϕ', the subtree of τ resulting from ϕ,[24] starts with an (unchecked) licensee instance. We could additionally "store" this information in the head–label of τ with the intend of preventing τ from being merged with another expression such that the resulting tree would contain two different maximal projections with an unchecked licensee instance. More generally, we could additionally "store" the head–label of ϕ within the head–label of τ; and within the head–label of each expression subsequently derived from τ we could not only store the head–label of ϕ, but also the number of the still unchecked licensee instances introduced by this head–label as long as there is still such an unchecked licensee instance. This would enable us to postpone the "actual insertion of ϕ" until it has reached its final landing site.

The last considerations, indeed, provide us rather rather straightforwardly with a method of constructing a weakly equivalent CFG for G_{MG}. This, at least, is true if we take into account only expressions being derivable from the lexicon by means of *merge* and *move*, and serving to derive a complete: first, it should be mentioned that there is only a finite number of possibilities for an extended head–labeling in the outlined way.[25] But the crucial reason why such a construction becomes finally possible is that each expression τ derivable from the lexicon of G_{MG}, and serving to derive a complete expression contains at most one maximal projection with an unchecked instance of some licensee feature starting the head-label, since the cardinality of the licensee set is 1.[26] That is to say, whenever we predict, in virtual terms of our weakly equivalent CFG, a specifier position to be the landing site of some maximal projection τ_1, we do not have to worry about the possibility that τ_1 in its turn contains a "trace" which has arisen from extracting some maximal projection τ_1' out of τ_1. Such a configuration cannot appear under any circumstances.

Let us also note here that—in a second step—it is possible to directly combine each of the converting methods presented in Section 3 with the just mentioned one in order to prove that the weak generative capacity of none of the considered subclasses of MGs allowing different types of head movement is increased if, additionally, the set of licensees is allowed to contain a single element.

5. Beyond simple phrasal movement

We conclude by emphasizing that, on the other hand, phrasal movement in the sense of the MG–definition which arises from the interaction of two different licensees already permits us to derive languages not derivable by any TAG. As an example of a corresponding MG we finally present the MG $G_{\text{NON-LIL}}$ deriving the language $\{a_1^n b_1^n a_2^m b_2^m c_1^n d_1^n c_2^m d_2^m \mid m, n \geq 0\}$, and consisting of the following 17 lexical items:

24. That is, ϕ' is either the complement or the highest specifier of τ.
25. Recall that the MG lexicon is finite, that each label of a lexical head is a finite string of features, and that an MG builds structure exclusively by canceling particular feature instances of these labels after an lexical head has been introduced in the course of a derivation.
26. Recall that, due to the definition of *move*, the implementation of the shortest move constraint (SMC) within MGs allows at most one such maximal projection for each different licensee in order to let an expression occur in the derive a complete expression.

$\mathtt{x}_{b_1}\ b_1$		$\mathtt{=x}_{a_1}\ \mathtt{x}_{d_1}\ d_1$	$\mathtt{=x}_{d_1}\ \mathtt{y}_1\ c_1$	$\mathtt{y}_1\ \mathtt{-l}_1$
$\mathtt{=x}_{c_1}\ \mathtt{+L}_1\ \mathtt{x}_{b_1}\ b_1$	$\mathtt{=x}_{b_1}\ \mathtt{x}_{a_1}\ \mathtt{-l}_1\ a_1$	$\mathtt{=x}_{a_1}\ \mathtt{+L}_2\ \mathtt{x}_{d_1}\ d_1$	$\mathtt{=x}_{d_1}\ \mathtt{x}_{c_1}\ \mathtt{-l}_2\ c_1$	
$\mathtt{x}_{b_2}\ b_2$		$\mathtt{=x}_{a_2}\ \mathtt{x}_{d_2}\ d_2$	$\mathtt{=x}_{d_2}\ \mathtt{y}_2\ c_2$	$\mathtt{y}_2\ \mathtt{-l}_2$
$\mathtt{=x}_{c_2}\ \mathtt{+L}_2\ \mathtt{x}_{b_2}\ b_2$	$\mathtt{=x}_{b_2}\ \mathtt{x}_{a_2}\ \mathtt{-l}_2\ a_2$	$\mathtt{=x}_{a_2}\ \mathtt{+L}_1\ \mathtt{x}_{d_2}\ d_2$	$\mathtt{=x}_{d_2}\ \mathtt{x}_{c_2}\ \mathtt{-l}_1\ c_2$	
				$\mathtt{=y}_2\ \mathtt{=y}_1\ \mathtt{+L}_2\ \mathtt{+L}_1\ \mathtt{c}$

References

Chomsky, Noam. 1995a. Categories and Transformations. In Chomsky, 1995b, pages 219–394.

Chomsky, Noam. 1995b. *The Minimalist Program*. Cambridge, MA: MIT Press.

Cornell, Thomas L. 1996. A Minimalist Grammar for Deriving the Copy Language. Report No. 79, Working papers of the SFB 340, Tübingen University, Tübingen. Available at `http://www.sfs.nphil.uni-tuebingen.de/sfb/reports/`.

de Groote, Philippe, Glyn Morrill and Christian Retoré, editors. 2001. *Logical Aspects of Computational Linguistics (LACL '01)*, Lecture Notes in Artificial Intelligence Vol. 2099. Berlin, Heidelberg: Springer.

Harkema, Henk. 2000. A Recognizer for Minimalist Grammars. In *Proceedings of the Sixth International Workshop on Parsing Technologies (IWPT 2000)*, Trento, pages 111–122.

Harkema, Henk. 2001. A Characterization of Minimalist Languages. In de Groote, Morrill and Retoré, 2001, pages 193–211.

Joshi, Aravind K. 2000. Relationship Between Strong and Weak Generative Power of Formal Systems. In *Proceedings of the 5th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+5)*, Paris, pages 107–114.

Kayne, Richard S. 1998. Overt vs. covert movement. *Syntax*, 1:128–191.

Kayne, Richard S. 1999. Prepositional complementizers as attractors. *Probus*, 11:39–73.

Koopman, Hilda and Anna Szabolcsi. 2000. *Verbal Complexes*. Cambridge, MA: MIT Press.

Mahajan, Anoop. 2000. Eliminating Head Movement. In *GLOW Newsletter #44*, pages 44–45. Abstract of the talk held at *the 23rd Generative Linguistics in the Old World Conference (GLOW 2000)*, Vitoria–Gasteiz/Bilbao.

Michaelis, Jens. 2001a. Derivational Minimalism is Mildly Context–Sensitive. In M. Moortgat, editor, *Logical Aspects of Computational Linguistics (LACL '98)*, Lecture Notes in Artificial Intelligence Vol. 2014. Springer, Berlin, Heidelberg, pages 179–198.

Michaelis, Jens. 2001b. Transforming Linear Context–Free Rewriting Systems into Minimalist Grammars. In de Groote *et al.* (2001), pages 228–244. Also available at `http://www.ling.uni-potsdam.de/~michael/papers.html`.

Michaelis, Jens and Christian Wartena. 1999. LIGs with Reduced Derivation Sets. In Gosse Bouma, Geert–Jan M. Kruijff, Erhard Hinrichs and Richard T. Oehrle, editors, *Constraints and Resources in Natural Language Syntax and Semantics*. CSLI Publications, Stanford, CA, pages 263–279.

Rambow, Owen and Giorgio Satta. 1999. Independent Parallelism in Finite Copying Parallel Rewriting Systems. *Theoretical Computer Science*, 223:87–120.

Schabes, Yves and Richard C. Waters. 1995. Tree Insertion Grammar: A Cubic–time Parsable Formalism that Lexicalizes Context–Free Grammar Without Changing the Trees Produced. *Computational Linguistics*, 21:479–513.

Seki, Hiroyuki, Takashi Matsumura, Mamoru Fujii and Tadao Kasami. 1991. On multiple context–free grammars. *Theoretical Computer Science*, 88:191–229.

Stabler, Edward P. 1997. Derivational Minimalism. In C. Retoré, editor, *Logical Aspects of Computational Linguistics (LACL '96)*, Lecture Notes in Artificial Intelligence Vol. 1328. Springer, Berlin, Heidelberg, pages 68–95.

Stabler, Edward P. 1998. Acquiring Languages with Movement. *Syntax*, 1:72–97.

Stabler, Edward P. 2001. Recognizing Head Movement. In de Groote *et al.* (2001), pages 245–260. Draft available at `http://www.humnet.ucla.edu/humnet/linguistics/people/stabler/`.

Stabler, Edward P. and Edward L. Keenan. 2000. Structural Similarity. In *Algebraic Methods in Language Processing.* Proceedings of the 16th Twente Workshop on Language Technology (TWLT 16) joint with the 2nd AMAST Workshop on Language Processing, Iowa City, IA, pages 251–265.

Vijay–Shanker, K. and David J. Weir. 1994. The equivalence of four extensions of context–free grammars. *Mathematical Systems Theory*, 27:511–546.

Vijay–Shanker, K. David J. Weir and Aravind K. Joshi. 1987. Characterizing Structural Descriptions Produced By Various Grammatical Formalisms. In *25th Annual Meeting of the Association for Computational Linguistics (ACL '87)*, Stanford, CA, pages 104–111. ACL.

Wartena, Christian. 1999. *Storage Structures and Conditions on Movement in Natural Language Syntax*. Ph.D. thesis, Potsdam University, Potsdam. Available at `http://www.ling.uni-potsdam.de/~wartena/`.

Weir, David J. 1988. *Characterizing Mildly Context–Sensitive Grammar Formalisms*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.