Transforming Linear Context–Free Rewriting Systems into Minimalist Grammars*

Jens Michaelis

Universität Potsdam, Institut für Linguistik, PF 601553, 14415 Potsdam, Germany
michael@ling.uni-potsdam.de

Abstract. The type of a minimalist grammar (MG) as introduced by Stabler [11, 12] provides an attempt of a rigorous algebraic formalization of the new perspectives adopted within the linguistic framework of transformational grammar due to the change from GB–theory to minimalism. Michaelis [6] has shown that MGs constitute a subclass of mildly context–sensitive grammars in the sense that for each MG there is a weakly equivalent linear context–free rewriting system (LCFRS). However, it has been left open in [6], whether the respective classes of string languages derivable by MGs and LCFRSs coincide. This paper completes the picture by showing that MGs in the sense of [11] and LCFRSs in fact determine the same class of derivable string languages.

1 Introduction

The type of a minimalist grammar (MG) as introduced in [11, 12] provides an attempt of a rigorous algebraic formalization of the new perspectives adopted within the linguistic framework of transformational grammar due to the change from GB–theory to minimalism. As shown in [6], MGs expose a subclass of mildly context–sensitive grammars in the sense that for each MG there is a weakly equivalent linear context–free rewriting system (LCFRS). More recently, in [1] it has been pointed out how the method to convert an MG into a weakly equivalent LCFRS can be employed to define a agenda–driven, chart–based recognizer for minimalist languages solving the recognition problem as to a given MG and an input string in deterministic polynomial time. Nevertheless, it has been left open until now, whether the respective classes of string languages derivable by MGs and LCFRSs coincide. This paper completes the picture by proving each LCFRS to necessarily generate a string language which, indeed, is also an MG–definable string language. Hence, one of the interesting outcomes is that MGs, beside LCFRSs, join to a series of formalism classes—among which there is e.g. the class of *multicomponent tree adjoining grammars (MCTAGs)* in their set–local variant of admitted adjunction (cf. [16])—all generating the same class

* This work has been carried out funded by DFG–grant STA 519/1-1. I especially wish to thank Kai–Uwe Kühnberger and Ed Stabler for inspiring discussions.

of string languages, which is known to be a *substitution–closed full AFL*. Furthermore, another consequence, type specific of the MG–formalism, arises from our particular construction of a weakly equivalent MG for a given LCFRS. The crucial point implied is that each MG can be transformed into a weakly equivalent MG that does not employ any kind of head movement or covert phrasal movement. This does not only prove a quite simpler formal setting for MGs to have the same weak generative capacity as the original one, but this also chimes in with current developments within the linguistic framework.

2 Linear Context–Free Rewriting Systems

The class of *linear context–free rewriting systems (LCFRSs)* [15, 16] constitutes a proper subclass of the class of *multiple context–free grammars (MCFGs)* [10] where in terms of derivable string languages both classes have identical generative power. MCFGs in their turn expose a special subtype of *generalized context–free grammars (GCFGs)* as introduced in [8].

Definition 2.1 ([8]). A five–tuple $G = \langle N, O, F, R, S\rangle$ for which (G1)–(G5) hold is called a *generalized context–free grammar (GCFG).*

(G1) N is a finite non–empty set of *nonterminal symbols.*

(G2) O is a set of *(linguistic) objects.*

(G3) F is a finite subset of $\bigcup_{n\in\mathbb{N}} F_n \setminus \{\emptyset\}$, where F_n is the set of partial functions from $\langle O\rangle^n$ into O, i.e. $F_0 \setminus \{\emptyset\}$ is the set of all constants in O.[1]

(G4) $R \subseteq \bigcup_{n\in\mathbb{N}}(F \cap F_n) \times N^{n+1}$ is a finite set of *(rewriting) rules.*[2]

(G5) $S \in N$ is the distinguished *start symbol.*

A rule $r = \langle f, A_0A_1 \cdots A_n\rangle \in (F \cap F_n) \times N^{n+1}$ for some $n \in \mathbb{N}$ is generally written $A_0 \to f(A_1, \ldots, A_n)$, and also just $A_0 \to f()$ in case $n = 0$. If the latter, i.e. $f() \in O$, then r is *terminating*, otherwise r is *nonterminating*. For each $A \in N$ and each $k \in \mathbb{N}$ the set $L^k_G(A) \subseteq O$ is given recursively in the following sense:

(L1) $\theta \in L^0_G(A)$ for each terminating rule $A \to \theta \in R$.

(L2) $\theta \in L^{k+1}_G(A)$, if $\theta \in L^k_G(A)$, or if there is a rule $A \to f(A_1, \ldots, A_n) \in R$ and there are $\theta_i \in L^k_G(A_i)$ for $1 \le i \le n$ such that $\langle\theta_1, \ldots, \theta_n\rangle \in Dom(f)$ and $f(\theta_1, \ldots, \theta_n) = \theta$.[3]

---

[1] $\mathbb{N}$ denotes the set of all non–negative integers. For $n \in \mathbb{N}$ and any sets $M_1, \ldots, M_n$, $\prod_{i=1}^n M_i$ is the set of all n–tuples $\langle m_1, \cdots, m_n\rangle$ with i–th component $m_i \in M_i$, where $\prod_{i=1}^0 M_i = \{\emptyset\}$, $\prod_{i=1}^1 M_i = M_1$, and $\prod_{i=1}^j M_i = \prod_{i=1}^{j-1} M_i \times M_j$ for $1 < j \le n$. We write $\langle M\rangle^n$ instead of $\prod_{i=1}^n M_i$ if for some set M, $M_i = M$ for $1 \le i \le n$.

[2] For any set M and $n \in \mathbb{N}$, M^{n+1} is the set of all finite strings in M of length $n+1$. M^* is the Kleene closure of M, including ϵ, the empty string. M_ϵ is the set $M \cup \{\epsilon\}$.

[3] For each partial function g from some set M_1 into some set M_2, $Dom(g)$ denotes the domain of g, i.e. the set of all $x \in M_1$ for which $g(x)$ is defined.

We say A *derives* θ *(in G)* if $\theta \in L_G^k(A)$ for some $A \in N$ and $k \in \mathbb{N}$. In this case θ is called an *A–phrase (in G)*. For each $A \in N$ the *language derivable from A (by G)* is the set $L_G(A)$ of all A–phrases (in G), i.e. $L_G(A) = \bigcup_{k \in \mathbb{N}} L_G^k(A)$. The set $L_G(S)$, also denoted by $L(G)$, is the *language derivable by G*.

Definition 2.2. An GCFG G_1 and G_2 are *weakly equivalent* if $L(G_1) = L(G_2)$.

Definition 2.3 ([10]). For $m \in \mathbb{N} \setminus \{0\}$ an *m–multiple context–free grammar (m–MCFG)* is a GCFG $G = \langle N, O_\Sigma, F, R, S \rangle$ which satisfies (M1)–(M4).

(M1) $O_\Sigma = \bigcup_{i=1}^{m} \langle \Sigma^* \rangle^i$ for some finite non–empty set Σ of *terminal symbols* with $\Sigma \cap N = \emptyset$. Hence O_Σ, the set of objects, is the set of all non–empty, finite tuples of finite strings in Σ such that each tuple has at most m components.

(M2) For each $f \in F$ let $n(f) \in \mathbb{N}$ be the *rank of f*, the number of components of an argument of f, i.e. $f \subseteq \langle O_\Sigma \rangle^{n(f)} \times O_\Sigma$. Then for each $f \in F$ there exists a number $\varphi(f) \in \mathbb{N} \setminus \{0\}$, called the *fan–out of f*, and there are numbers $d_i(f) \in \mathbb{N} \setminus \{0\}$ for $1 \leq i \leq n(f)$ such that f is a (total) function from $\prod_{i=1}^{n(f)} \langle \Sigma^* \rangle^{d_i(f)}$ into $\langle \Sigma^* \rangle^{\varphi(f)}$ for which (f1) and, in addition, the *anti–copying condition* (f2) hold.

(f1) Define $I_{Dom(f)}$ by $\{\langle i, j \rangle \in \mathbb{N} \times \mathbb{N} \mid 1 \leq i \leq n(f), 1 \leq j \leq d_i(f)\}$ and take $X_f = \{x_{ij} \mid \langle i, j \rangle \in I_{Dom(f)}\}$ to be a set of pairwise distinct variables. For $1 \leq i \leq n(f)$ let $x_i = \langle x_{i1}, \ldots, x_{id_i(f)} \rangle$. For $1 \leq h \leq \varphi(f)$ let f_h be the h–th component of f, i.e. the function f_h from $\prod_{i=1}^{n(f)} \langle \Sigma^* \rangle^{d_i(f)}$ into Σ^* such that $f(\theta) = \langle f_1(\theta), \ldots, f_{\varphi(f)}(\theta) \rangle$ for all $\theta \in \prod_{i=1}^{n(f)} \langle \Sigma^* \rangle^{d_i(f)}$. Then, for each $1 \leq h \leq \varphi(f)$ there is an $l_h(f) \in \mathbb{N}$ such that f_h can be represented by

$$(\mathrm{c}_{f_h}) \qquad f_h(x_1, \ldots, x_{n(f)}) = \zeta(f_{h0})\, z(f_{h1})\, \zeta(f_{h1}) \cdots z(f_{hl_h(f)})\, \zeta(f_{hl_h(f)})$$

with $\zeta(f_{hl}) \in \Sigma^*$ for $0 \leq l \leq l_h(f)$ and $z(f_{hl}) \in X_f$ for $1 \leq l \leq l_h(f)$.

(f2) Define $I_{Range(f)}$ by $\{\langle h, l \rangle \in \mathbb{N} \times \mathbb{N} \mid 1 \leq h \leq \varphi(f), 1 \leq l \leq l_h(f)\}$ and let g_f denote the binary relation on $I_{Dom(f)} \times I_{Range(f)}$ such that $\langle \langle i, j \rangle, \langle h, l \rangle \rangle \in g_f$ iff $x_{ij} = z(f_{hl})$. Then g_f is a partial function from $I_{Dom(f)}$ into $I_{Range(f)}$,[4] i.e. there is at most one occurrence of each $x_{ij} \in X_f$ within all righthand sides of (c_{f_1})–$(\mathrm{c}_{f_{\varphi(f)}})$.

(M3) There is a function d_G from N to $\mathbb{N}$ such that, if $A_0 \to f(A_1, \ldots, A_n) \in R$ for some $n \in \mathbb{N}$ then $\varphi(f) = d_G(A_0)$ and $d_i(f) = d_G(A_i)$ for $1 \leq i \leq n$, where $\varphi(f)$ and $d_i(f)$ for $1 \leq i \leq n$ are as in (M2).

(M4) $d_G(S) = 1$ for the start symbol $S \in N$.

---

[4] Note that this implies that g_f is an injective, but not necessarily total, function from $I_{Dom(f)}$ onto $I_{Range(f)}$.

The *rank of G*, denoted by *rank*(G), is defined as $\max\{rank(f) \mid f \in F\}$. Note that $L(G) \subseteq \Sigma^*$ by (M4). In case that $m = 1$ and that each $f \in F \setminus F_0$ is the concatenation function from $\langle\Sigma^*\rangle^{n+1}$ to Σ^* for some $n \in \mathbb{N}$, G is a context–free grammar (CFG) and $L(G)$ a context–free language (CFL) in the usual sense.

Definition 2.4. Each $L \subseteq \Sigma^*$ for some set Σ such that there is an m–MCFG G with $L = L(G)$ is an *m–multiple context–free language (m–MCFL).*

Definition 2.5 ([15]). For $m \in \mathbb{N} \setminus \{0\}$ an m–MCFG $G = \langle N, O_\Sigma, F, R, S\rangle$ according to Definition 2.3 is called an *m–linear context–free rewriting system (m–LCFRS)* if for each $f \in F$ the *non–erasure condition* (f3) holds in addition to (f1) and (f2).

(f3) The function g_f from (f2) is total, i.e. each $x_{ij} \in X_f$ has to appear in one of the righthand sides of (c_{f_1})–$(\mathrm{c}_{f_{\varphi(f)}})$.

Definition 2.6. Each $L \subseteq \Sigma^*$ for some set Σ such that there is an m–LCFRS G with $L = L(G)$ is an *m–linear context–free rewriting language (m–LCFRL).*

For each $m \in \mathbb{N}\setminus\{0\}$, a given m–MCFG, m–MCFL, m–LCFRS, or m–LCFRL is likewise referred to simply as an *MCFG*, *MCFL*, *LCFRS*, or *LCFRL*, respectively.

The class of all MCFGs and the class of all LCFRSs are essentially the same. The latter was first described in [15] and has been studied in some detail in [16].[5] The MCFG–definition technically generalizes the LCFRS–definition by omitting the non–erasure condition (f3). But this bears no consequences as to matters of weak generative capacity as is fixed by Lemma 2.2 in [10]. Looking at the construction that Seki et al. [10] propose in order to end up with a weakly equivalent LCFRS for a given MCFG, it becomes clear that the following holds:

Corollary 2.7. *For each m–MCFG* $G = \langle N, O_\Sigma, F, R, S\rangle$ *with* $m \in \mathbb{N} \setminus \{0\}$ *there exists a weakly equivalent m–LCFRS* $G' = \langle N', O_\Sigma, F', R', S'\rangle$ *such that* $rank(G') \leq rank(G)$.

Combining this result with Theorem 11 in [9] we get

Corollary 2.8. *For every MCFG G there is an LCFRS G′ with* $rank(G') \leq 2$ *deriving the same language as G.*

[5] In particular, Weir [16] carefully develops the leading idea to come up with the definition of LCFRSs: the specific aim of the LCFRS–formalism is to provide a perspective under which several types of grammar formalisms, that all can be restated (in terms of weak equivalence) as specific GCFG–types dealing with more or less distinct types of objects, become directly comparable. This is achieved by taking the functions of the respective GCFGs to be simply function symbols rather than concrete operators applying to corresponding (tuples of) objects. Then, these function symbols are interpreted as *unique yield functions* which map the derived objects to tuples of terminal strings.

3 Minimalist Grammars

Throughout this section we let $\neg Syn$ and Syn be a set of *non–syntactic features* and a set of *syntactic features*, respectively, according to (F1)–(F3).

(F1) $\neg Syn$ is a finite set partitioned into a set *Phon* of *phonetic features* and a set *Sem* of *semantic features.*

(F2) *Syn* is a finite set disjoint from $\neg Syn$ and partitioned into a set *Base* of *(basic) categories*, a set *Select* of *selectors*, a set *Licensees* of *licensees* and a set *Licensors* of *licensors.* For each $x \in Base$, usually typeset as `x`, the existence of three pairwise distinct elements in *Select*, respectively denoted by `=x`, `=X` and `X=`, is possible. For each $x \in Licensees$, usually depicted in the form `-x`, the existence of two distinct elements in *Licensors*, denoted by `+x` and `+X`, is possible. Selectors and licensors of the form `=X`, `X=` or `+X` are said to be *strong*, those of the form `=x` or `+x` are said to be *weak.*

(F3) `c` is a distinguished element from *Base*, the *completeness category.*

We take *Feat* to be the set defined by $\neg Syn \cup Syn$.

Definition 3.1. A *tree domain* is a non–empty set $N_\tau \subseteq \mathbb{N}^*$ which is *prefix closed* and *left closed*, i.e. for all $\chi \in \mathbb{N}^*$ and $i \in \mathbb{N}$ it holds that $\chi \in N_\tau$ if $\chi\chi' \in N_\tau$ for some $\chi' \in \mathbb{N}^*$, and $\chi i \in N_\tau$ if $\chi j \in N_\tau$ for some $j \in \mathbb{N}$ with $i < j$.

Definition 3.2. A five–tuple $\tau = \langle N_\tau, \triangleleft^*_\tau, \prec_\tau, <_\tau, label_\tau \rangle$ fulfilling (E1)–(E4) is called an *expression (over Feat).*

(E1) $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau \rangle$ is a finite, binary (ordered) tree defined in the usual sense, where N_τ is the finite, non–empty set of *nodes*, and where $\triangleleft^*_\tau$ and $\prec_\tau$ are the binary relations of *dominance* and *precedence* on N_τ, respectively. Thus, $\triangleleft^*_\tau$ is the reflexive–transitive closure of $\triangleleft_\tau \subseteq N_\tau \times N_\tau$, the relation of *immediate dominance*, and each non–leaf in $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau \rangle$ has exactly two children.[6]

(E2) $<_\tau \subseteq N_\tau \times N_\tau$ is the asymmetric relation of *(immediate) projection (in τ)* that holds for any two siblings in $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau \rangle$, i.e. each node different from the root either *(immediately) projects* over its sibling or vice versa.

(E3) $label_\tau$ is the *leaf–labeling function (of τ)* which assigns an element from $Syn^* Phon^* Sem^*$ to each leaf of $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau \rangle$, i.e. each leaf–label is a finite sequence of features from *Feat*.

(E4) $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau \rangle$ is a subtree of some natural tree domain interpretation.[7]

[6] Up to an isomorphism $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau \rangle$ is the *natural (tree) interpretation* of some tree domain. In other words, up to an isomorphism N_τ is a tree domain such that for all $\chi, \psi \in N_\upsilon$ it holds that $\chi \triangleleft_\tau \psi$ iff $\psi = \chi i$ for some $i \in \mathbb{N}$, and $\chi \prec_\tau \psi$ iff $\chi = \omega i \chi'$ and $\psi = \omega j \psi'$ for some $\omega, \chi', \psi' \in \mathbb{N}^*$ and $i, j \in \mathbb{N}$ with $i < j$.

[7] That is, there is some tree domain N_υ with $N_\tau \subseteq N_\upsilon$ such that, as to the natural tree interpretation $\langle N_\upsilon, \triangleleft^*_\upsilon, \prec_\upsilon \rangle$ of N_υ, the root r_τ of $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau \rangle$ meets the condition that for each $x \in N_\upsilon$ it holds that $x \in N_\tau$ iff $r_\tau \triangleleft^*_\upsilon x$. Moreover it holds that $\triangleleft^*_\upsilon = \triangleleft^*_\tau \restriction_{N_\upsilon \times N_\upsilon}$ and $\prec_\upsilon = \prec_\tau \restriction_{N_\upsilon \times N_\upsilon}$.

The set of all expressions over *Feat* is denoted by $Exp(Feat)$.

Let $\tau = \langle N_\tau, \triangleleft^*_\tau, \prec_\tau, <_\tau, label_\tau \rangle \in Exp(Feat)$.

For each $x \in N_\tau$, the *head of x (in τ)*, denoted by $head_\tau(x)$, is the (unique) leaf of τ such that $x \triangleleft^*_\tau head_\tau(x)$, and such that each $y \in N_\tau$ on the path from x to $head_\tau(x)$ with $y \neq x$ projects over its sibling. The *head of τ* is the head of τ's root. τ is said to be a *head* (or *simple*) if N_τ consists of exactly one node, otherwise τ is said to be a *non–head* (or *complex*).

A *subexpression of τ* is a five–tuple $\upsilon = \langle N_\upsilon, \triangleleft^*_\upsilon, \prec_\upsilon, <_\upsilon, label_\upsilon \rangle$ such that $\langle N_\upsilon, \triangleleft^*_\upsilon, \prec_\upsilon \rangle$ is a subtree of $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau \rangle$, and such that $<_\upsilon = <_\tau \restriction_{N_\upsilon \times N_\upsilon}$ and $label_\upsilon = label_\tau \restriction_{N_\upsilon}$. Thus, υ is an expression over *Feat*. The set of all subexpressions of τ is denoted by $Subexp(\tau)$.

An expression $\upsilon \in Subexp(\tau)$ is a *maximal projection (in τ)* if υ's root is a node $x \in N_\tau$ such that x is the root of τ, or such that $sibling_\tau(x) <_\tau x$.[8] Thus, the number of maximal projections in τ and the number of leaves of τ coincide, and two maximal projections in τ are identical in case they share the same head. We take $MaxProj(\tau)$ to be the set of all maximal projections in τ. Note that for each subexpression $\upsilon \in MaxProj(\tau)$ it holds that $MaxProj(\upsilon) \subseteq MaxProj(\tau)$.

$comp_\tau \subseteq MaxProj(\tau) \times MaxProj(\tau)$ is the binary relation such that for all $\upsilon, \phi \in MaxProj(\tau)$ it holds that $\upsilon \, comp_\tau \, \phi$ iff $head_\tau(r_\upsilon) <_\tau r_\phi$, where r_υ and r_ϕ are the roots of υ and ϕ, respectively. If $\upsilon \, comp_\tau \, \phi$ for some $\upsilon, \phi \in MaxProj(\tau)$ then ϕ is a *complement of υ (in τ)*. $comp^+_\tau$ is the transitive closure of $comp_\tau$. $Comp^+(\tau)$ is the set $\{\upsilon \mid \tau \, comp^+_\tau \, \upsilon\}$.

$spec_\tau \subseteq MaxProj(\tau) \times MaxProj(\tau)$ is the binary relation such that for all $\upsilon, \phi \in MaxProj(\tau)$ it holds that $\upsilon \, spec_\tau \, \phi$ iff $r_\phi = sibling_\tau(x)$ for some $x \in N_\tau$ with $r_\upsilon \triangleleft^+_\tau x \triangleleft^+_\tau head_\tau(r_\upsilon)$, where r_υ and r_ϕ are the roots of υ and ϕ, respectively. If $\upsilon \, spec_\tau \, \phi$ for some $\upsilon, \phi \in MaxProj(\tau)$ then ϕ is a *specifier of υ (in τ)*. $spec^*_\tau$ is the reflexive–transitive closure of $spec_\tau$. $Spec(\tau)$ and $Spec^*(\tau)$ are the sets $\{\upsilon \mid \tau \, spec_\tau \, \upsilon\}$ and $\{\upsilon \mid \tau \, spec^*_\tau \, \upsilon\}$, respectively.

An $\upsilon \in MaxProj(\tau)$ is said to *have feature f* or, likewise, to *be with feature f* if for some $f \in Feat$ the label assigned to the head of υ by $label_\tau$ is non–empty and starts with an instance of f.

τ is *complete* if its head–label is in $\{\mathtt{c}\}Phon^*Sem^*$ and each other of its leaf–labels is in $Phon^*Sem^*$. Hence, a complete expression over *Feat* is an expression that has category c, and this instance of c is the only instance of a syntactic feature within all leaf–labels.

The *yield of τ*, denoted by $Y(\tau)$, is defined as the string which results from concatenating in "left–to–right–manner" the labels assigned to the leaves of $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau \rangle$ via $label_\tau$. The *phonetic yield (of τ)*, denoted by $Y_{Phon}(\tau)$, is the string which results from $Y(\tau)$ by replacing all instances of non–phonetic features in $Y(\tau)$ with the empty string.

An expression $\upsilon = \langle N_\upsilon, \triangleleft^*_\upsilon, \prec_\upsilon, <_\upsilon, label_\upsilon \rangle \in Feat(Exp)$ is *(labeling preserving) isomorphic to τ* if there exists a bijective function i from N_τ onto N_υ such that $x \triangleleft_\tau y$ iff $i(x) \triangleleft_\upsilon i(y)$, $x \prec_\tau y$ iff $i(x) \prec_\upsilon i(y)$, and $x <_\tau y$ iff $i(x) <_\upsilon i(y)$

[8] $sibling_\tau(x)$ denotes the (unique) sibling of a given $x \in N_\tau$ different to τ's root

for all $x, y \in N_\tau$, and such that $label_\tau(x) = label_\upsilon(i(x))$ for all $x \in N_\tau$. This function i is an *isomorphism (from τ to υ)*.

Definition 3.3. For each given $\tau = \langle N_\tau, \triangleleft^*_\tau, \prec_\tau, <_\tau, label_\tau \rangle \in Exp(Feat)$ with $N_\tau = r_\tau N_\upsilon$ for some $r_\tau \in \mathbb{N}^*$ and some tree domain N_υ, and for each given $r \in \mathbb{N}^*$, the *expression shifting τ to r*, denoted by $(\tau)_r$, is the expression $\langle N_{\tau(r)}, \triangleleft^*_{\tau(r)}, \prec_{\tau(r)}, <_{\tau(r)}, label_{\tau(r)} \rangle \in Exp(Feat)$ with $N_{\tau(r)} = rN_\upsilon$ such that the function $i_{\tau(r)}$ from N_τ onto $N_{\tau(r)}$ with $i_{\tau(r)}(r_\tau x) = rx$ for all $x \in N_\upsilon$ constitutes an isomorphism from τ to $(\tau)_r$.[9]

Introducing a related notational convention, we assume υ and ϕ to be expressions over *Feat*, and consider the expressions $(\upsilon)_0$ and $(\phi)_1$ shifting υ to 0 and ϕ to 1, respectively. We write $[_<\upsilon, \phi]$ (respectively, $[_>\upsilon, \phi]$) in order to refer to the complex expression $\chi = \langle N_\chi, \triangleleft^*_\chi, \prec_\chi, <_\chi, label_\chi \rangle$ over *Feat* with root ϵ such that $(\upsilon)_0$ and $(\phi)_1$ are the two subexpressions of χ whose roots are immediately dominated by ϵ, and such that $0 <_\chi 1$ (respectively, $1 <_\chi 0$).

We next introduce a type of MGs strongly in line with the definition given in [11]. But different to there, we do so by demanding all selection features of an MG to be weak, and all licensor features to be strong. In this sense, we only define a subtype of MGs as introduced in [11]. For this subtype there is no need to explicitly define what is meant by *(overt) head movement* or *covert (phrasal) movement*, respectively, since there are no features which potentially trigger these kinds of movement (cf. [11]). Moreover, this subtype will be sufficient to prove the class of LCFRSs to be weakly embeddable into the class of MGs.

Definition 3.4. A five–tuple $G = \langle \neg Syn, Syn, Lex, \Omega, \mathtt{c} \rangle$ that obeys (N0)–(N2) is called a *minimalist grammar (MG)*.

(N0) All syntactic features from *Select* are weak, while all syntactic features from *Licensors* are strong.

(N1) *Lex* is a *lexicon (over Feat)*, i.e. a finite subset of *Exp*(*Feat*) such that for each $\tau = \langle N_\tau, \triangleleft^*_\tau, \prec_\tau, <_\tau, label_\tau \rangle \in Lex$ the set of nodes, N_τ, is a tree domain and the leaf–labeling function, $label_\tau$, maps each leaf of τ onto an element from $Select^* Licensors_\epsilon Select^* Base_\epsilon Licensees^* Phon^* Sem^*$.

(N2) The set Ω consists of the structure building functions *merge* and *move* defined w.r.t. $\neg Syn \cup Syn$ as in (me) and (mo) below, respectively.

(me) The operator *merge* is as a partial mapping from $Exp(Feat) \times Exp(Feat)$ into $Exp(Feat)$. A pair $\langle \upsilon, \phi \rangle$ of some expressions υ and ϕ over *Feat* belongs to $Dom(merge)$ if for some $\mathtt{x} \in Base$ conditions (i) and (ii) are fulfilled.

(i) υ has selector $\mathtt{=x}$, and

(ii) ϕ has category $\mathtt{x}$.

[9] Hence, $\langle N_{\tau(\epsilon)}, \triangleleft^*_{\tau(\epsilon)}, \prec_{\tau(\epsilon)} \rangle$ is identical to the natural tree interpretation of the tree domain N_υ. Note that by (E4), for every $\tau = \langle N_\tau, \triangleleft^*_\tau, \prec_\tau, <_\tau, label_\tau \rangle \in Exp(Feat)$ there do exist an $r_\tau \in \mathbb{N}^*$ and a tree domain N_υ with $N_\tau = r_\tau N_\upsilon$.

Thus, there are $\kappa_\upsilon, \kappa_\phi \in Syn^*$ $\nu_\upsilon, \nu_\phi \in Phon^* Sem^*$ such that $\texttt{=x}\kappa_\upsilon\nu_\upsilon$ and $\texttt{x}\kappa_\phi\nu_\phi$ are the head–labels of υ and ϕ, respectively. The value of $\langle\upsilon, \phi\rangle$ under *merge* is subject to two distinct subcases.

(me.1) $$merge(\upsilon, \phi) = [_<\upsilon', \phi'\,] \text{ if } \upsilon \text{ is simple,}$$

where υ' and ϕ' are the expressions resulting from υ and ϕ, respectively, by replacing the head–labels: the head–label of υ becomes $\kappa_\upsilon\nu_\upsilon$ in υ', that of ϕ becomes $\kappa_\phi\nu_\phi$ in ϕ'. Hence, υ' and ϕ' respectively result from υ and ϕ just by deleting the instance of the feature that the corresponding head–label starts with.

(me.2) $$merge(\upsilon, \phi) = [_>\phi', \upsilon'\,] \text{ if } \upsilon \text{ is complex,}$$

where υ' and ϕ' are defined the same way as in (me.1).

(mo) The operator *move* is a partial mapping from *Exp*(*Feat*) into *Exp*(*Feat*). An $\upsilon \in Exp(Feat)$ belongs to *Dom*(*move*) if for some $\texttt{-x} \in Licensees$ conditions (i) and (ii) are true.

(i) υ has licensor feature `+X`, and

(ii) there is exactly one maximal projection ϕ in υ that has feature `-x`.

Thus, there are $\kappa_\upsilon, \kappa_\phi \in Syn^*$, $\nu_\upsilon, \nu_\phi \in Phon^* Sem^*$ such that $\texttt{+X}\kappa_\upsilon\nu_\upsilon$ and $\texttt{-x}\kappa_\phi\nu_\phi$ are the head–labels of υ and ϕ, respectively. The outcome of the application of *move* to υ is defined as

$$move(\upsilon) = [_>\phi', \upsilon'\,]\,,$$

where the expression υ' results from υ by canceling the instance of `+X` the head–label of υ starts with, while the subtree ϕ is replaced by a single node labeled ϵ. ϕ' is the expression arising from ϕ just by deleting the instance of `-x` that the head–label of ϕ starts with.

For each MG $G = \langle\neg Syn, Syn, Lex, \Omega, \texttt{c}\rangle$ the *closure of Lex (under the functions in* Ω), briefly referred to as the *closure of* G and denoted by $CL(G)$, is defined as $\bigcup_{k\in\mathbb{N}} CL^k(G)$, a countable union of subsets of *Exp*(*Feat*), where for $k \in \mathbb{N}$ the sets $CL^k(G)$ are inductively given by

(C1) $CL^0(G) = Lex$

(C2) $$\begin{aligned} CL^{k+1}(G) = CL^k(G) & \\ &\cup \{merge(\upsilon,\phi) \mid \langle\upsilon,\phi\rangle \in Dom(merge) \cap CL^k(G) \times CL^k(G)\} \\ &\cup \{move(\upsilon) \mid \upsilon \in Dom(move) \cap CL^k(G)\}\end{aligned}$$

Recall that the functions *merge* and *move* are structure building by strict feature consumption. Thus, since $CL^0(G) = Lex$, each application of *merge* or *move* deriving some $\tau \in CL(G)$ can be seen as "purely lexically driven."

Every $\tau \in CL(G)$ is an *expression of* G. The *(string) language derivable by* G is the set $\{Y_{Phon}(\tau) \mid \tau \in CL(G) \text{ such that } \tau \text{ is complete}\}$, denoted by $L(G)$.

Definition 3.5. Each $L \subseteq Phon^*$ for some set *Phon* such that there is an MG G with $L = L(G)$ is called a *minimalist language (ML).*

Just in order to complete the picture in terms of a formal definition we give

Definition 3.6. An MG G and an MCFG G' are *weakly equivalent* if they derive the same (string) language, i.e. if $L(G) = L(G')$.

4 MCFLs as MLs

In this section we take $G = \langle N, O_\Sigma, F, R, S\rangle$ to be an arbitrary, but fixed m–MCFG for some $m \in \mathbb{N} \setminus \{0\}$. In order to prepare the construction of a weakly equivalent MG we start by considering the functions from F of the MCFG G in some more detail.

Let $f \in F$. We first choose non–negative integers $n(f)$, $1 \leq \varphi(f) \leq m$ and $1 \leq d_i(f) \leq m$ for $1 \leq i \leq n(f)$, existing according to (M2) such that

f is a (total) function from $\prod_{i=1}^{n(f)} \langle \Sigma^* \rangle^{d_i(f)}$ into $\langle \Sigma^* \rangle^{\varphi(f)}$.

Next we define

$$I_{Dom(f)} = \{\langle i, j\rangle \,|\, 1 \leq i \leq n(f), 1 \leq j \leq d_i(f)\},$$

we let $X_f = \{x_{ij} \,|\, \langle i, j\rangle \in I_{Dom(f)}\}$ be a set of pairwise distinct variables, and we set $x_i = \langle x_{i1}, \ldots, x_{id_i(f)}\rangle$ for $1 \leq i \leq n(f)$. Then, for $1 \leq h \leq \varphi(f)$ we take f_h to be the h–th component of f, i.e. $f(\theta) = \langle f_1(\theta), \ldots, f_{\varphi(f)}(\theta)\rangle$ for all $\theta \in \prod_{i=1}^{n(f)} \langle \Sigma^* \rangle^{d_i(f)}$, and we fix $l_h(f) \in \mathbb{N}$, $\zeta(f_{hl}) \in \Sigma^*$ for $0 \leq l \leq l_h(f)$, and $z(f_{hl}) \in X_f$ for $1 \leq l \leq l_h(f)$, existing by (f1), such that f_h is represented by

$$f_h(x_1, \ldots, x_{n(f)}) = \zeta(f_{h0})\, z(f_{h1})\, \zeta(f_{h1}) \cdots z(f_{hl_h(f)})\, \zeta(f_{hl_h(f)})\ .$$

Proceeding for each $f \in F$ we let

$$I_{Range(f)} = \{\langle h, l\rangle \,|\, 1 \leq h \leq \varphi(f), 1 \leq l \leq l_h(f)\}\ ,$$

and we take g_f to be the injective partial function from $I_{Dom(f)}$ onto $I_{Range(f)}$ which exists according to (f2) such that

$$g_f(i, j) = \langle h, l\rangle \text{ iff } x_{ij} = z(f_{hl}) \text{ for each } \langle i, j\rangle \in Dom(g_f).$$

Sticking to a further notational convention introduced in Section 2, we take

d_G to denote the function from N into $\mathbb{N}$

existing due to (M3) and (M4). Thus $1 \leq d_G(A) \leq m$ for $A \in N$, where $d_G(S) = 1$.

We now define an MG $G_{\text{MG}} = \langle \neg Syn, Syn, Lex, \Omega, \mathsf{c}\rangle$ according to (N0)–(N2) and prove it to be weakly equivalent to the MCFG G afterwards. To do so successfully, we assume w.l.o.g. G to be an LCFRS with $rank(G) = 2$, what is possible according to Corollary 2.8.

Let us start with a motivation of the concrete construction we suggest below. For that, we consider some $r = A \to f(A_1, \ldots, A_{n(f)}) \in R$ and let $p_h \in L_G(A_h)$

for $1 \leq h \leq n(f)$. Thus we have $p = f(p_1, \ldots, p_{n(f)}) \in L_G(A) \subseteq \langle \Sigma^* \rangle^{d_G(A)}$. Our aim is to define G_{MG} such that we are able to derive an expression $\tau \in CL(G_{\mathrm{MG}})$ from existing expressions $\upsilon_1, \ldots, \upsilon_{n(f)} \in CL(G_{\mathrm{MG}})$, thereby successively "calculating" the $\varphi(f)$–tuple p in $n(f) + 3\varphi(f) + \sum_{h=1}^{\varphi(f)} 2l_h(f)$ steps. Recall that we have $d_G(A) = \varphi(f)$. Each υ_i for some $1 \leq i \leq n(f)$ will be related to A_i and p_i, and the resulting expression τ to A and p in a specific way (cf. Definition 4.1). Roughly speaking, as for τ, for each $1 \leq h \leq d_G(A)$ there will be some $\tau_h \in \mathit{MaxProj}(\tau)$ provided with a particular licensee instance. Up to those proper subtrees of τ_h which are themselves maximal projections with some licensee feature, the component p_h will be the phonetic yield of τ_h.

• • • First we let $\mathit{Phon} = \Sigma$ and $\mathit{Sem} = \emptyset$.

• • • Defining the sets *Licensees* and *Licensors*, for $1 \leq h \leq m$ and $0 \leq n \leq 2$ we take $\texttt{-l}_{\langle h,\, n \rangle}$ to be a licensee and $\texttt{+L}_{\langle h,\, n \rangle}$ to be a corresponding strong licensor such that *Licensees* and *Licensors* both are sets of cardinality $3m$.

• • • In order to define the sets *Base* and *Select*, for each $A \in N$ we introduce new, pairwise distinct basic categories $\texttt{a}_{\langle h,\, n \rangle}$ as well as corresponding weak selection features of the form $\texttt{=a}_{\langle h,\, n \rangle}$ with $1 \leq h \leq d_G(A)$ and $1 \leq n \leq 2$.

Furthermore, for each $A \to f(A_1, \ldots, A_{n(f)}) \in R$ we introduce new, pairwise distinct basic categories $\texttt{a}_{\langle f,\, \varphi(f)+1,\, 0 \rangle}$ and $\texttt{a}_{\langle f,\, h,\, l \rangle}$ as well as corresponding weak selection features of the form $\texttt{=a}_{\langle f,\, \varphi(f)+1,\, 0 \rangle}$ and $\texttt{=a}_{\langle f,\, h,\, l \rangle}$, where $1 \leq h \leq \varphi(f)$ and $0 \leq l \leq l_h(f)$. Recall that $\varphi(f) = d_G(A)$ by choice of d_G.

Finally, we let $\texttt{c} \in \mathit{Base}$ be the completeness category and assume it to be different from every other element in *Base*.

• • • Next we define the set *Lex*, the lexicon over $\neg \mathit{Syn} \cup \mathit{Syn}$. While doing so, we identify each lexical item with its (unique) head–label, taking such an item to be a simple expression with root ϵ. First of all we define one entry which is the only one that will allow "to finally derive" a complete expression of G_{MG}.

$$\alpha_{\texttt{c}} = \texttt{=s}_{\langle 1,\, 1 \rangle}\, \texttt{+L}_{\langle 1,\, 1 \rangle}\, \texttt{c}\,,$$

where $\texttt{s}_{\langle 1,\, 1 \rangle} \in \mathit{Base}$ is the corresponding category arising from $S \in N$, the start symbol in G. The form of all other entries in *Lex* depends on the production rules belonging to R. Since G is of rank 2, we distinguish three cases.

<u>*Case 1*</u>. $A \to f(B, C) \in R$ for some A, B, $C \in N$ and $f \in F$. In this case $\varphi(f) = d_G(A)$, $n(f) = 2$, $d_1(f) = d_G(B)$ and $d_2(f) = d_G(C)$. Then, the following element belongs to *Lex*:

$$\alpha_{\langle A,\, f,\, B,\, C \rangle} = \texttt{=c}_{\langle 1,\, 2 \rangle}\, \texttt{=b}_{\langle 1,\, 1 \rangle}\, \texttt{a}_{\langle f,\, \varphi(f)+1,\, 0 \rangle}$$

<u>*Case 2*</u>. $A \to f(B)$ for some $A, B \in N$ and $f \in F$. In this case $\varphi(f) = d_G(A)$, $n(f) = 1$ and $d_1(f) = d_G(B)$. Then, as an element of *Lex* we take

$$\alpha_{\langle A,\, f,\, B,\, - \rangle} = \texttt{=b}_{\langle 1,\, 1 \rangle}\, \texttt{a}_{\langle f,\, \varphi(f)+1,\, 0 \rangle}$$

$\underline{\textit{Case 3}}$. $A \to f()$ for some $A \in N$ and $f \in F$. In this case $\varphi(f) = d_G(A)$ and $n(f) = 0$. Since f is a constant in $\langle \Sigma^* \rangle^{\varphi(f)}$, we have $l_h(f) = 0$ for each $1 \le h \le \varphi(f)$, i.e. $f() = \langle \zeta(f_{10}), \ldots, \zeta(f_{\varphi(f)0}) \rangle$. The following entry is in *Lex*:

$$\alpha_{\langle A,\, f,\, -,\, - \rangle} \;=\; \mathtt{a}_{\langle f,\, \varphi(f)+1,\, 0 \rangle}$$

Moreover, in all three cases for each $1 \le h \le \varphi(f)$ further entries are added to *Lex* depending on whether $l_h(f) = 0$ or not.

For each $1 \le h \le \varphi(f)$ with $l_h(f) = 0$ we just add

$$\alpha_{\langle A,\, f,\, h,\, 0 \rangle} \;=\; \mathtt{=a}_{\langle f,\, h+1,\, 0 \rangle}\; \mathtt{a}_{\langle f,\, h,\, 0 \rangle}\; \mathtt{-l}_{\langle h,\, 0 \rangle}\; \zeta(f_{h0})\;.$$

For each $1 \le h \le \varphi(f)$ with $l_h(f) > 0$ we add

$$\alpha_{\langle A,\, f,\, h,\, 0 \rangle} \;=\; \mathtt{=a}_{\langle f,\, h,\, 1 \rangle}\; \mathtt{a}_{\langle f,\, h,\, 0 \rangle}\; \mathtt{-l}_{\langle h,\, 0 \rangle}\; \zeta(f_{h0})\;, \text{ and}$$

$$\alpha_{\langle A,\, f,\, h,\, l_h(f) \rangle} \;=\; \mathtt{=a}_{\langle f,\, h+1,\, 0 \rangle}\; \mathtt{+L}_{\langle j,\, i \rangle}\; \mathtt{a}_{\langle f,\, h,\, l_h(f) \rangle}\; \zeta(f_{hl_h(f)})\;,$$
$$\text{where } 1 \le i \le n(f) \text{ and } 1 \le j \le d_i(f) \text{ with } z(f_{hl_h(f)}) = x_{ij}.$$

For each $1 \le h \le \varphi(f)$ and for each $1 \le l < l_h(f)$ we add

$$\alpha_{\langle A,\, f,\, h,\, l \rangle} \;=\; \mathtt{=a}_{\langle f,\, h,\, l+1 \rangle}\; \mathtt{+L}_{\langle j,\, i \rangle} \mathtt{a}_{\langle f,\, h,\, l \rangle}\; \zeta(f_{hl})\;,$$
$$\text{where } 1 \le i \le n(f) \text{ and } 1 \le j \le d_i(f) \text{ with } z(f_{hl}) = x_{ij}.$$

Finally, in all three cases for $1 \le n \le 2$ we take as an element of *Lex*

$$\alpha_{\langle A,\, \varphi(f),\, n \rangle} \;=\; \mathtt{=a}_{\langle f,\, 1,\, 0 \rangle}\; \mathtt{+L}_{\langle \varphi(f),\, 0 \rangle}\; \mathtt{a}_{\langle \varphi(f),\, n \rangle} \mathtt{-l}_{\langle \varphi(f),\, n \rangle}\;,$$

and for $1 \le h < \varphi(f)$ we take

$$\alpha_{\langle A,\, h,\, n \rangle} \;=\; \mathtt{=a}_{\langle h+1,\, n \rangle}\; \mathtt{+L}_{\langle h,\, 0 \rangle}\; \mathtt{a}_{\langle h,\, n \rangle}\; \mathtt{-l}_{\langle h,\, n \rangle}\;.$$

Definition 4.1. For every given $A \in N$, $p = \langle \pi_1, \ldots, \pi_{d_G(A)} \rangle$ with $\pi_i \in \Sigma^*$ for $1 \le i \le d_G(A)$, and $1 \le n \le 2$ an expression $\tau \in CL(G_{\mathrm{MG}})$ is said to *correspond to* the triple $\langle A, p, n \rangle$ if (Z1)–(Z4) are fulfilled, where $\tau_{\langle 1, n \rangle} = \tau$.

(Z1) The head–label of τ is of the form $\mathtt{a}_{\langle 1,\, n \rangle}\; \mathtt{-l}_{\langle 1,\, n \rangle}\; \pi_{\langle 1,\, n \rangle}$ for some $\pi_{\langle 1,\, n \rangle} \in \Sigma^*$.

(Z2) For each $2 \le h \le d_G(A)$ there is exactly one $\tau_{\langle h,\, n \rangle} \in \mathit{Comp}^+(\tau)$ whose head–label is of the form $\mathtt{-l}_{\langle h,\, n \rangle}\; \pi_{\langle h,\, n \rangle}$ for some $\pi_{\langle h,\, n \rangle} \in \Sigma^*$.

(Z3) For each $1 \le h \le d_G(A)$ it holds that

$$\{\upsilon \in \mathit{MaxProj}(\tau_{\langle h,\, n \rangle}) \setminus \{\tau_{\langle h,\, n \rangle}\} \mid \upsilon \text{ has some licensee feature}\}$$
$$= \{\tau_{\langle i,\, n \rangle} \mid h < i \le d_G(A)\},$$

i.e. for each $1 \le h < d_G(A)$ the subexpression $\tau_{\langle h+1,\, n \rangle}$ is the unique maximal maximal projection in $\tau_{\langle h,\, n \rangle}$ that has some licensee feature.

(Z4) For each $1 \leq h \leq d_G(A)$ the string π_h is the phonetic yield of $\upsilon_{\langle h,\, n\rangle}$. Here we have $\upsilon_{\langle d_G(A),\, n\rangle} = \tau_{\langle d_G(A),\, n\rangle}$, and for $1 \leq h < d_G(A)$ the expression $\upsilon_{\langle h,\, n\rangle}$ results from $\tau_{\langle h,\, n\rangle}$ by replacing the subtree $\tau_{\langle h+1,\, n\rangle}$ with a single node labeled ϵ.

Proposition 4.2. *Let $\tau \in CL(G_{MG})$ such that τ has category feature $\mathtt{a}_{\langle 1,\, n\rangle}$ for some $A \in N$ and $1 \leq n \leq 2$. Then there is some $p \in L_G(A)$ such that τ corresponds to $\langle A, p, n\rangle$.*

Proof (sketch). In order to avoid the trivial case we assume that there is some expression $\tau \in CL(G_{\mathrm{MG}})$ such that τ has category $\mathtt{a}_{\langle 1,\, n\rangle}$ for some $A \in N$ and $1 \leq n \leq 2$. Then there is a smallest $K \in \mathbb{N}$ for which $CL^K(G_{\mathrm{MG}})$ includes such a τ. According to the definition of *Lex* we have $K > 0$. The proof follows from an induction on $k \in \mathbb{N}$ with $k + 1 \geq K$.

For some $k \in \mathbb{N}$ with $k + 1 \geq K$ consider some arbitrary, but fixed expression $\tau \in CL^{k+1}(G_{\mathrm{MG}}) \setminus CL^k(G_{\mathrm{MG}})$ such that τ has category $\mathtt{a}_{\langle 1,\, n\rangle}$ for some $A \in N$ and $1 \leq n \leq 2$. Taking into account the definition of *Lex* it turns out that the procedure to derive τ as an expression of G_{MG} is deterministic in the following sense: there are some $r = A \to f(A_1, \ldots, A_{n(f)}) \in R$, some $k_0 \in \mathbb{N}$ with $k_0 = k + 1 - 3\varphi(f) - \sum_{h=1}^{\varphi(f)} 2l_h(f)$ and some $\chi_0 \in CL^{k_0}(G_{\mathrm{MG}})$ such that χ_0 serves to derive τ in G_{MG}. χ_0 has category feature $\mathtt{a}_{\langle f,\, \varphi(f)+1,\, 0\rangle}$ and is of one of three forms depending on r:

Case 1. There is some $r = A \to f(B, C) \in R$, and there are $\upsilon, \phi \in CL^{k_0}(G_{\mathrm{MG}})$ such that υ and ϕ have category feature $\mathtt{b}_{\langle 1,\, 1\rangle}$ and $\mathtt{c}_{\langle 1,\, 2\rangle}$, respectively, and

$$\chi_0 = merge(\upsilon, merge(\alpha_{\langle A,\, f,\, B,\, C\rangle}, \phi)).$$

By induction hypothesis there are some $p_B \in L_G(B)$ and $p_C \in L_G(C)$ such that υ and ϕ correspond to $\langle B, p_B, 1\rangle$ and $\langle C, p_C, 2\rangle$, respectively. In this case we define $p \in L_G(A)$ by $p = f(p_B, p_C)$.

Case 2. There are some $r = A \to f(B) \in R$ and $\upsilon \in CL^{k_0}(G_{\mathrm{MG}})$ such that υ has category feature $\mathtt{b}_{\langle 1,\, 1\rangle}$, and such that

$$\chi_0 = merge(\alpha_{\langle A,\, f,\, B,\, -\rangle}, \upsilon).$$

By induction hypothesis there is some $p_B \in L_G(B)$ such that υ corresponds to $\langle B, p_B, 1\rangle$. Let $p = f(p_B) \in L_G(A)$.

Case 3. There is some $r = A \to f() \in R$ and χ_0 is a lexical item,

$$\chi_0 = \alpha_{\langle A,\, f,\, -,\, -\rangle}.$$

In this case we simply let $p = f() \in L_G(A)$.

Note that, if $k + 1 = K$ (constituting the base case of our induction) then χ_0 is necessarily of the last form by choice of K. In any case, it turns out that the given

$\tau \in CL^{k+1}(G_{\mathrm{MG}}) \setminus CL^k(G_{\mathrm{MG}})$ corresponds to $\langle A, p, n\rangle$. The single derivation steps to end up with τ starting from χ_0 are explicitly given by the following procedure.

Procedure (derive τ from χ_0).

For $0 \leq h < \varphi(f)$

$\psi_{\langle h+1,\,0\rangle} = \chi_h$

for $0 \leq l < l_{\varphi(f)-h}(f)$

step $2l+1+h+\sum_{h'=0}^{h-1} 2l_{\varphi(f)-h'}(f)$

$\psi_{\langle h+1,\,2l+1\rangle} = \mathit{merge}(\alpha_{\langle A,\,f,\,\varphi(f)-h,\,l_{\varphi(f)-h}(f)-l\rangle}, \psi_{\langle h+1,\,2l\rangle})$

step $2l+2+h+\sum_{h'=0}^{h-1} 2l_{\varphi(f)-h'}(f)$

$\psi_{\langle h+1,\,2l+2\rangle} = \mathit{move}(\psi_{\langle h+1,\,2l+1\rangle})$

[checks licensee $\texttt{-l}_{\langle j,\,i\rangle}$
with $g_f(i,j) = \langle \varphi(f)-h, l_{\varphi(f)-h}(f)-l\rangle$]

step $h+1+\sum_{h'=0}^{h} 2l_{\varphi(f)-h'}(f)$

$\chi_{h+1} = \mathit{merge}(\alpha_{\langle A,\,f,\,\varphi(f)-h,\,0\rangle}, \psi_{\langle h+1,\,2l_{\varphi(f)-h}(f)\rangle})$

For $0 \leq h < \varphi(f)$

step $2h+1+\varphi(f)+\sum_{h'=1}^{\varphi(f)} 2l_{h'}(f)$

$\chi_{\varphi(f)+2h+1} = \mathit{merge}(\alpha_{\langle A,\,\varphi(f)-h,\,n\rangle}, \chi_{\varphi(f)+2h})$

step $2h+2+\varphi(f)+\sum_{h'=1}^{\varphi(f)} 2l_{h'}(f)$

$\chi_{\varphi(f)+2h+2} = \mathit{move}(\chi_{\varphi(f)+2h+1})$

[checks licensee $\texttt{-l}_{\langle \varphi(f)-h,\,0\rangle}$]

$\tau = \chi_{3\varphi(f)}$

An embedded induction on $0 \leq h < \varphi(f)$ and $0 \leq l < l_{\varphi(f)-h}(f)$ yields that τ indeed corresponds to $\langle A, p, n \rangle$, which shows that the proposition is true. The reader is encouraged to verify the details. One crucial point that we like to stress here concerns Case 1 and 2: since G is an LCFRS, the injective function g_f from $I_{Dom(f)}$ onto $I_{Ran(f)}$ is total, i.e. g_f is bijective. This guarantees that each instance of a licensee feature appearing within the yield of χ_0 gets checked off by some derivation step $2l + 2 + h + \sum_{h'=0}^{h-1} 2l_{\varphi(f)-h'}(f)$ with $0 \leq h < \varphi(f)$ and $0 \leq l \leq l_{\varphi(f)-h}(f)$. □

At this point it seems to be suitable to emphasize the reason for a specific peculiarity intrinsic to G_{MG} by definition of *Lex*: assume $\tau \in CL(G_{\text{MG}})$ to have category feature $\mathtt{a}_{\langle 1,\, n \rangle}$ for some $A \in N$ and $1 \leq n \leq 2$. Consider the derivation process from appropriate $\chi_0 \in CL(G_{\text{MG}})$ to τ as given above, showing that τ corresponds to $\langle A, p, n \rangle$ according to Definition 4.1 for the respective $p \in L_G(A)$. The question that might arise is, why, by virtual means of G_{MG}, each component p_{h+1} of p for some $0 \leq h < d_G(A)$ is first related to some maximal projection that has licensee $\texttt{-l}_{\langle h+1,\, 0 \rangle}$ and not directly to some maximal projection that has licensee $\texttt{-l}_{\langle h+1,\, n \rangle}$. The answer is straightforward: the corresponding instance of $\texttt{-l}_{\langle h+1,\, n \rangle}$ becomes potentially subject to the move–operator exactly after the expression $\chi_{d_G(A)-h}$ has been selected by a lexical head under an application of *merge*. To put it differently, the resulting expression χ contains a maximal projection $\chi'_{d_G(A)-h}$ that has licensee $\texttt{-l}_{\langle h,\, 0 \rangle}$. Namely, $\chi'_{d_G(A)-h}$ is the complement of χ, i.e. the right co–constituent of the head of χ. If we now look at the representations of the components of the involved function $f \in F$ by means of variables and constants from Σ^*, we see that it is possible that the variable x_{nh} occurs within such a representation of some component $f_{h'}$ of f with $1 \leq h' < h$. This means that we have to be aware of the fact that, beside $\chi'_{d_G(A)-h}$, χ may include a further maximal projection $\chi''_{d_G(A)-h}$ that has licensee $\texttt{-l}_{\langle h,\, n \rangle}$. If $\chi'_{d_G(A)-h}$ and $\chi''_{d_G(A)-h}$ had the same licensee, we would never be able to check off one of both respective instances due to the definition of *move*. Therefore, a derivation of a complete expression of G would unavoidably be blocked.

Proposition 4.3. *Let $A \in N$, $p \in \langle \Sigma^* \rangle^{d_G(A)}$ and $1 \leq n \leq 2$. If $p \in L_G(A)$ then there is some $\tau \in CL(G_{MG})$ such that τ corresponds to $\langle A, p, n \rangle$.*

Proof (sketch). Once more an induction will do the job to prove the proposition. Let $A \in N$, $p \in \langle \Sigma^* \rangle^{d_G(A)}$ and $1 \leq n \leq 2$. Assume that $p \in L_G(A)$. Then, w.l.o.g. we are concerned with one of three possible cases.

Case 1. There is some $r = A \rightarrow f(B, C) \in R$, and for some $k \in \mathbb{N}$ there are some $p_B \in L_G^k(B)$ and $p_C \in L_G^k(C)$ such that $p = f(p_B, p_C) \in L_G^{k+1}(A) \setminus L_G^k(A)$. By hypothesis on k there exist some $\upsilon, \phi \in CL(G_{\text{MG}})$ such that υ and ϕ correspond to $\langle B, p_B, 1 \rangle$ and $\langle C, p_C, 2 \rangle$, respectively. Therefore, we can define $\chi_0 \in CL(G_{\text{MG}})$ by

$$\chi_0 = \mathit{merge}(\upsilon, \mathit{merge}(\alpha_{\langle A,\, f,\, B,\, C \rangle}, \phi).$$

Case 2. There is some $r = A \rightarrow f(B, C) \in R$, and for some $k \in \mathbb{N}$ there is some $p_B \in L^k_G(B)$ such that $p = f(p_B) \in L^{k+1}_G(A) \setminus L^k_G(A)$. Here, by induction hypothesis we can choose some $\upsilon \in CL(G_{\mathrm{MG}})$ such that υ corresponds to $\langle B, p_B, 1\rangle$. Then we define $\chi_0 \in CL(G_{\mathrm{MG}})$ by

$$\chi_0 = merge(\alpha_{\langle A,\, f,\, B,\, -\rangle}, \upsilon).$$

Case 3. There is some $r = A \rightarrow f() \in R$ such that $p = f() \in L^0_G(A)$. In this case we take χ_0 to be a particular lexical item. We set

$$\chi_0 = \alpha_{\langle A,\, f,\, -,\, -\rangle}.$$

In all three cases the respective derivation procedure from the proof of the last proposition shows that χ_0 serves to derive a $\tau \in CL(G_{\mathrm{MG}})$ that has the demanded properties. □

Corollary 4.4. *$\pi \in L(G)$ iff $\pi \in L(G_{MG})$ for each $\pi \in \Sigma^*$.*

Proof. First suppose that $\tau \in CL(G_{\mathrm{MG}})$ is complete such that $\pi \in \Sigma^*$ is the phonetic yield of τ. Then, due to the definition of *Lex*, there is some expression $\tau' \in CL(G_{\mathrm{MG}})$ which has category $\mathtt{s}_{\langle 1,\, 1\rangle}$ such that $\tau = move(merge(\alpha_{\mathtt{c}}, \tau'))$. By Proposition 4.2 there is some $p' \in L_G(S) = L(G)$ such that τ' corresponds to $\langle S, p', 1\rangle$. Because $d_G(S) = 1$, this implies that p' is not only the phonetic yield of τ', but also the phonetic yield of the specifier of τ. Since the phonetic yield of $\alpha_{\mathtt{c}}$ is empty, we conclude that $p' = \pi$.

Now assume that $\pi \in L(G) = L_G(S)$ for some $\pi \in \Sigma^*$. By Proposition 4.3 there is some $\tau' \in CL(G_{\mathrm{MG}})$ such that τ' corresponds to $\langle S, \pi, 1\rangle$. Then, because $d_G(S) = 1$, π is the phonetic yield of τ'. Moreover, $\tau = move(merge(\alpha_{\mathtt{c}}, \tau'))$ is defined and complete, and π is the phonetic yield of τ. □

5 Conclusion

We have shown that each LCFRS can be transformed into a weakly equivalent MG as defined in [11]. As shown in [6], the converse holds, as well. Hence, combining these results crucially implies that MGs fit in within a series of different formal grammar types, each of which constituting a class of generating devices that have the same weak generative power as LCFRSs, respectively MCFGs.[10] The presented result, therefore, also provides an answer to several questions as to the properties of MLs that have been left open so far, and that can be subsumed under a more general question: does the class of MG–definable string languages constitute an *abstract family of languages (AFL)*? The answer to this question is now known to be positive; even a stronger property is true for this language class, since it is provably a *substitution–closed full AFL* (cf. [10]).

Taking into account our specific construction of a weakly equivalent MG for a given LCFRS, we have moreover shown that each MG in the sense of [11] can

[10] For a list of some of such classes of generating devices, beside MCTAGs, see e.g. [9].

be converted into a weakly equivalent one which does not employ any kind of head movement or covert phrasal movement.[11] In fact, it is this subtype of MGs which Harkema's recognizer [1] for MLs is actually defined for. But maybe even more crucially, this implication could be considered to provide some technical support to Stabler's proposal of a revised type of an MG given in [13], which, in particular, completely dispense with those two types of movement motivated by recent linguistic work which heads in exactly this direction (see e.g. [3], [4], [5]). The same holds as to the type of a *strict MG (SMG)*, also introduced in [13] keeping closely to some further suggestions in [4], which likewise banishes any kind of head movement and covert phrasal movement from the list of possibilities to "displace material" by means of the structure building functions of an MG. Furthermore, each lexical item of an MG of revised type as well as an SMG is by definition a simple expression, i.e. a head, and in case of an SMG the label of such a head is an element from $\mathit{Select}_\epsilon(\mathit{Select} \cup \mathit{Licensors})_\epsilon \mathit{Base}\,\mathit{Licensees}^* \neg \mathit{Syn}^*$. This latter property is also common to the MG G_{MG} as it results according to our construction in Section 4 from a given LCFRS G, i.e. the constructed MG G_{MG} being weakly equivalent to the LCFRS G. Thus, the creation of *multiple specifiers* is avoided during the derivation of an expression of G_{MG}. To put it differently, whenever, for some $\tau \in \mathit{CL}(G_{\mathrm{MG}})$ and some $\upsilon \in \mathit{MaxProj}(\tau)$, we have $\mathit{Spec}(\upsilon) \neq \emptyset$, $\mathit{Spec}(\upsilon)$ is a singleton set. Indeed, this specific property constitutes one of the main differences to the construction of a weakly equivalent MG for a given LCFRS as it is independently developed in [2]. Although quite similar in some other aspects, within Harkema's construction the use of multiple specifiers is rather a constitutive element.[12]

The last remarks should, certainly, be treated with some care: as demonstrated in [7], thereby confirming the corresponding conjecture in [13], the revised MG–type and the SMG–type introduced in [13] determine the same class of derivable string languages. This is an immediate consequence of the fact that both types are provably weakly equivalent to one and the same particular subtype of LCFRSs, respectively MCFGs. However, as to our knowledge, it is an open problem whether this LCFRS–subtype in its turn is weakly equivalent to the class of all LCFRSs, and thus to the class of MGs in the sense of [11]. Note that, deviating from the definition in [11], the revised MG–type as well as the SMG–type does not only dispense with any kind of head movement and covert phrasal movement, but also an additional restriction is imposed on the move–operator as to which maximal projection may move overtly.[13] Therefore, neither

[11] This is still true, if we additionally allow *affix hopping* to take place within an MG in the way suggested in [14].

[12] Another significant difference between our approach and the one of Harkema is given by the fact that within our resulting, weakly equivalent MG no maximal projection moves more than one time in order to check its licensee features, i.e. the (non–trivial) chains created by applications of the move–operator are all simple.

[13] As to an MG of revised type, an $\upsilon \in \mathit{Exp}(\neg\mathit{Syn} \cup \mathit{Syn})$ belongs to the domain of the move–operator only if, in addition to condition (i) and (ii) of (mo), it holds that there is some $\chi \in \mathit{Comp}^+(\upsilon)$ with $\phi = \chi$ or $\phi \in \mathit{Spec}(\chi)$ for the unique maximal

our method to convert a given LCFRS into a weakly equivalent MG in the sense of [11], nor the method of Harkema does necessarily yield, at the same time, a weakly equivalent MG or, likewise, SMG in the sense of [13]; and there is no straightforward adaption of either of both methods in order to achieve this.

References

1. Henk Harkema. A recognizer for minimalist grammars. In *Proceedings of the Sixth International Workshop on Parsing Technologies (IWPT 2000),* Trento, pages 111–122, 2000.
2. Henk Harkema. A characterization of minimalist languages, 2001. This volume.
3. Richard S. Kayne. Overt vs. covert movement. *Syntax*, 1:128–191, 1998.
4. Hilda Koopman and Anna Szabolcsi. *Verbal Complexes.* MIT Press, Cambridge, MA, 2000.
5. Anoop Mahajan. Eliminating head movement. In *GLOW Newsletter # 44*, pages 44–45, 2000. Abstract of the talk held at *the 23rd Generative Linguistics in the Old World Conference (GLOW 2000)*, Vitoria–Gasteiz/Bilbao.
6. Jens Michaelis. 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, to appear. Also available at `http://www.ling.uni-potsdam.de/~michael/papers.html`.
7. Jens Michaelis. On formal properties of minimalist grammars. Potsdam University, Potsdam, in preperation.
8. Carl J. Pollard. *Generalized Phrase Structure Grammars, Head Grammars, and Natural Language.* PhD thesis, Stanford University, Stanford, CA, 1984.
9. Owen Rambow and Giorgio Satta. Independent parallelism in finite copying parallel rewriting systems. *Theoretical Computer Science*, 223:87–120, 1999.
10. Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context–free grammars. *Theoretical Computer Science*, 88:191–229, 1991.
11. Edward P. Stabler. Derivational minimalism. In C. Retoré, editor, *Logical Aspects of Computational Linguistics (LACL '96)*, Lecture Notes in Artificial Intelligence Vol. 1328, pages 68–95. Springer, Berlin, Heidelberg, 1997.
12. Edward P. Stabler. Acquiring languages with movement. *Syntax*, 1:72–97, 1998.
13. Edward P. Stabler. Remnant movement and complexity. In G. Bouma, G.–J. M. Kruijff, E. Hinrichs, and R. T. Oehrle, editors, *Constraints and Resources in Natural Language Syntax and Semantics*, pages 299–326. CSLI Publications, Stanford, CA, 1999.
14. Edward P. Stabler. Recognizing head movement, 2001. This volume.
15. K. Vijay–Shanker, David J. Weir, and Aravind K. Joshi. 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, 1987.
16. David J. Weir. *Characterizing Mildly Context–Sensitive Grammar Formalisms.* PhD thesis, University of Pennsylvania, Philadelphia, PA, 1988.

---

projection ϕ that has licensee `-x`. As to an SMG, an $\upsilon \in Exp(\neg Syn \cup Syn)$ belongs to the domain of the move–operator only if, in addition to condition (i) and (ii) of (mo), it holds that there is a $\chi \in Comp^{+}(\upsilon)$ with $\phi \in Spec^{*}(\chi)$ for the unique maximal projection ϕ that has licensee `-x`, and applying the move–operator to υ, it is the constituent χ that is raised into the specifier position.