Electronic Notes in Theoretical Computer Science 53 (2004)
URL: http://www.elsevier.nl/locate/entcs/volume53.html 18 pages

Observations on Strict Derivational Minimalism

Jens Michaelis

Universität Potsdam, Institut für Linguistik, PF 601553, 14415 Potsdam, Germany

Abstract

Deviating from the definition originally presented in [12], Stabler [13] introduced—inspired by some recent proposals in terms of a minimalist approach to transformational syntax—a (revised) type of a *minimalist grammar (MG)* as well as a certain type of a *strict minimalist grammar (SMG)*. These two types can be shown to determine the same class of derivable string languages.

1 Introduction

The type of a *minimalist grammar (MG)* as introduced in [12] provides an attempt of a rigorous formalization of the perspectives adopted nowadays within the linguistic framework of transformational grammar. As shown in [4], this type of an MG constitutes a weakly equivalent subclass of *linear context–free rewriting systems (LCFRSs)* [14,15]. Recently, independent work of Harkema [2] and Michaelis [7] has proven the reverse to be true as well. Hence, MGs as defined in [12], beside LCFRSs, join to a series of *mildly context–sensitive 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. [15])—all generating the same class of string languages, which is known to be a *substitution–closed full AFL.* [1] Mainly inspired by the linguistic work presented in [3], in [13] a revised type of an MG has been proposed whose departure from the version in [12] can be seen as twofold: the revised type of an MG neither employs any kind of *head movement* nor *covert phrasal movement*, and an additional restriction is imposed on the move–operator as to which maximal projection may move *overtly*. Deviating from the operation *move* as originally defined in [12], a constituent has necessarily to belong to the transitive closure of the complement relation or to be a specifier of such a

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

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

constituent in order to be movable. Closely in keeping with some further suggestions in [3], a certain type of a *strict minimalist grammar (SMG)* has been introduced in [13] as well. This MG–type allows only movement of constituents belonging to the transitive closure of the complement relation. But different from the first type, the triggering licensee feature may head the head–label of any constituent within the reflexive–transitive closure of the specifier relation of a moving constituent. Furthermore, due to the general definition of a lexical item of an SMG, an SMG does not permit the creation of multiple specifiers in the course of a derivation. This paper answers to some important questions explicitly left open in [13]: the respective types of an MG and an SMG are shown to determine the same class of derivable string languages. This is done by proving both formalism types to be weakly equivalent to the same subclass of LCFRSs. The respective class of generated string languages is also shown to constitute a substitution–closed full AFL. Whether it coincides with the class of all LCFRS–definable string languages remains an open problem here.

2 Multiple Context–Free Grammars

LCFRSs form a proper subclass of *multiple context–free grammars (MCFGs)* [11], which in their turn are a subtype of *generalized context–free grammars* [8]. But LCFRSs define the same class of derivable string languages as MCFGs.

Definition 2.1 [8] A *generalized context–free grammar (GCFG)* is a five–tuple $G = \langle N, O, F, R, S\rangle$, where N is a finite non–empty set of *nonterminals*, and where O is a set of *(linguistic) objects*. F is a finite subset of $\bigcup_{n\in\mathbb{N}} F_n \backslash \{\emptyset\}$, F_n the set of partial functions from $\langle O\rangle^n$ into O. [2] R is a finite set of *(rewriting) rules*, i.e. a subset of $\bigcup_{n\in\mathbb{N}}(F\cap F_n)\times\langle N\rangle^{n+1}$. S is a distinguished symbol from N, the *start symbol*.

An $r = \langle f, \langle A_0, A_1, \ldots, A_n\rangle\rangle \in (F\cap F_n)\times\langle N\rangle^{n+1}$ for some $n\in\mathbb{N}$ is written $A_0 \rightarrow f(A_1,\ldots,A_n)$, and also $A_0\rightarrow f(\emptyset)$ if $n = 0$. In case $n = 0$, i.e. if f is a constant in O, r is *terminating*, otherwise r is *nonterminating*.

For each $A\in N$ and $k\in\mathbb{N}$, $L^k_G(A)\subseteq O$ is given recursively by means of $\theta\in L^0_G(A)$ for each terminating $A\rightarrow\theta\in R$, and $\theta\in L^{k+1}_G(A)$ if $\theta\in L^k_G(A)$, or if there are $A\rightarrow f(A_1,\ldots,A_n)\in R$ and $\theta_i\in L^k_G(A_i)$ 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] The set $L_G(A)=\bigcup_{k\in\mathbb{N}} L^k_G(A)$ is the *language derivable from A (by G)*. $L_G(S)$, also denoted by $L(G)$, is the *language derivable by G*.

Definition 2.2 [11] A *multiple context–free grammar (MCFG)* is a GCFG $G = \langle N, O, F, R, S\rangle$ with $O = \bigcup_{n\in\mathbb{N}}\langle\Sigma^*\rangle^{n+1}$, and satisfying (M1) and (M2),

[2] $\mathbb{N}$ is 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 $\langle m_1,\ldots,m_n\rangle$ with i–th component $m_i\in M_i$, where $\prod_{i=1}^n M_i := \{\emptyset\}$ for $n=0$. We write $\langle M\rangle^n$ instead of $\prod_{i=1}^n M_i$ if for some set M, $M_i = M$ for $1\le i\le n$.

[3] For each partial function g from a set M into a set M', $Dom(g)\subseteq M$ is the domain of g.

where Σ is a finite set of *terminals* with $\Sigma \cap N = \emptyset$.[4]

(M1) For each $f \in F$, some $n(f) \in \mathbb{N}$, $\varphi(f) \in \mathbb{N} \setminus \{0\}$ and $d_i(f) \in \mathbb{N} \setminus \{0\}$ for $1 \leq i \leq n(f)$ exist 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 (f2) hold.

(f1) Let $X_f = \{x_{ij} \mid 1 \leq i \leq n(f), 1 \leq j \leq d_i(f)\}$ be a set of pairwise distinct variables, for $1 \leq i \leq n(f)$ let $x_i = \langle x_{i1}, \ldots, x_{id_i(f)} \rangle$, and for $1 \leq h \leq \varphi(f)$ let f_h be the h–th component of f, i.e. the function from $Dom(f)$ into Σ^* such that $f(\theta) = \langle f_1(\theta), \ldots, f_{\varphi(f)}(\theta) \rangle$ for all $\theta \in Dom(f)$. Then, for each $1 \leq h \leq \varphi(f)$ there are an $l_h(f) \in \mathbb{N}$, a $\zeta(f_{hl}) \in \Sigma^*$ for $0 \leq l \leq l_h(f)$, and a $z(f_{hl}) \in X_f$ for $1 \leq l \leq l_h(f)$ such that f_h is represented by (c_{f_h}).

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

(f2) Each $x \in X_f$ occurs at most once in all righthand sides of (c_{f_1})–$(\mathrm{c}_{f_{\varphi(f)}})$, i.e. for the set $I_{Dom(f)} = \{\langle i, j \rangle \mid 1 \leq i \leq n(f), 1 \leq j \leq d_i(f)\}$ and for the set $I_{Range(f)} = \{\langle h, l \rangle \mid 1 \leq h \leq \varphi(f), 1 \leq l \leq l_h(f)\}$, the binary relation $g_f \subseteq I_{Dom(f)} \times I_{Range(f)}$ such that $\langle \langle i, j \rangle, \langle h, l \rangle \rangle \in g_f$ iff $x_{ij} = z(f_{hl})$ is an injective partial function onto $I_{Range(f)}$.

(M2) There is a function d_G from N into $\mathbb{N}$ with $d_G(S) = 1$ 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 $1 \leq i \leq n$.

The *rank of* G, denoted by $rank(G)$, is the number $\max\{n(f) \mid f \in F\}$. The language derivable by G, the set $L(G)$, is called a *multiple context–free language (MCFL)*. Note that $L(G) \subseteq \Sigma^*$, because $d_G(S) = 1$.

Definition 2.3 [14,15] An MCFG G in the sense of Definition 2.2 such that for each $f \in F$ condition (f3) holds in addition to (f1) and (f2) is a called *linear context–free rewriting system (LCFRS)*. In this case $L(G)$ is a *linear context–free rewriting language (LCFRL)*.

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

The class of all MCFLs and the class of all LCFRLs are known to be identical (cf. [11, Lemma 2.2]). Theorem 11 in [9], therefore, leads to

Corollary 2.4 *For each MCFG G there is a weakly equivalent LCFRS G' with $rank(G') \leq 2$.*

Definition 2.5 An *MCFG*$_{1,2}$ (*LCFRS*$_{1,2}$) is an MCFG (LCFRS) G in the sense of Definition 2.2 (Definition 2.3) such that $rank(G) \leq 2$, and such that $d_1(f) = 1$ for each $f \in F$ with $n(f) = 2$. In this case the language derivable by G is an *MCFL*$_{1,2}$ (*LCFRL*$_{1,2}$).

[4] For each set M, M^* is the Kleene closure of M, including ϵ, the empty string. M_ϵ denotes the set $M \cup \{\epsilon\}$.

3 MCFGs in Monotone Function Form

We now introduce a special type of an MCFG, the type of an MCFG in *monotone function form (MFF)*, which will be of considerable interest in Section 6. Roughly, the idea leading to the corresponding definition is the fact that (at least in terms of weak equivalence) "synchronized parallelism" in an MCFG is in a certain sense independent of the order of the constituents (each of which represented by a terminal string) that are derivable as a tuple from a given nonterminal. More technically, for a given rule $r = A \to f(A_1, \dots, A_{n(f)})$, it is not the order of the components of a $d_i(f)$–tuple $\theta_i = \langle \theta_{i1}, \dots, \theta_{id_i(f)} \rangle$ derivable from the nonterminal A_i that "really matters," but rather the (partial) order of these components induced by their "left–to–right–appearance" within the components of the $\varphi(f)$–tuple $f(\theta_1, \dots, \theta_{n(f)})$ derivable from A by means of r. Using this insight, we will focus on the possibility of an "*a priori*–re–ordering" of the components of a corresponding $d_i(f)$–tuple θ_i in a particular way: it is a consequence of (f1) and (f2) that for each $1 \leq i \leq n(f)$ there is a permutation $\delta_i(f)$ on $\{1, \dots, d_i(f)\}$ such that for $1 \leq j, j' \leq d_i(f)$ with $j < j'$, if the variables x_{ij} and $x_{ij'}$ appear at all within some component $f_h(x_1, \dots, x_{n(f)})$ for some $1 \leq h \leq \varphi(f)$, these two variables are "monotonically" ordered by $\delta_i(f)$ w.r.t. the function g_f from (f2) in the sense that

$$\delta_i(f)(j) < \delta_i(f)(j') \text{ iff } g_f(i, \delta_i(f)(j)) <_{\mathbb{N}\times\mathbb{N}} g_f(i, \delta_i(f)(j'))$$

for each $\langle i, j \rangle, \langle i, j' \rangle \in \mathit{Dom}(g_f)$.[5] What we will rely on is that each MCFG G can be transformed into a weakly equivalent MCFG G' such that, in particular, for each function f in G', the corresponding "monotonic" order w.r.t. g_f for $1 \leq i \leq n(f)$ holds with $\delta_i(f)$ being the identity function on $\{1, \dots, d_i(f)\}$.

Definition 3.1 An MCFG $G = \langle N, O, F, R, S \rangle$ is in *monotone function form (MFF)* if for each $f \in F$ and $i, j, k \in \mathbb{N}$ with $\langle i, j \rangle, \langle i, k \rangle \in \mathit{Dom}(g_f)$ it holds that $j < k$ iff $g_f(i, j) <_{\mathbb{N}\times\mathbb{N}} g_f(i, k)$, where g_f is defined as in (f2).

Proposition 3.2 *For each MCFG$_{1,2}$ G is a weakly equivalent LCFRS$_{1,2}$ G' which is in MFF.*

Proof. By Corollary 2.2.10 and 2.4.4 of [5]. □

4 Minimalist Grammars

Throughout we let $\neg \mathit{Syn}$ and Syn be a finite set of *non–syntactic features* and a finite set of *syntactic features*, respectively, in accordance with (F1) and (F2) below. We take *Feat* to be the set $\neg \mathit{Syn} \cup \mathit{Syn}$.

(F1) $\neg \mathit{Syn}$ is disjoint from *Syn* and partitioned into a set *Phon* of *phonetic features* and a set *Sem* of *semantic features*.

[5] $<_{\mathbb{N}\times\mathbb{N}}$ denotes the lexical order on $\mathbb{N} \times \mathbb{N}$, i.e. for all $p, q, p', q' \in \mathbb{N}$, $\langle p, q \rangle <_{\mathbb{N}\times\mathbb{N}} \langle p', q' \rangle$ iff (a) $p < p'$ or (b) $p = p'$ and $q < q'$.

(F2) *Syn* is 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 a matching $x' \in Select$, denoted by =x, is possible. For each $x \in Licensees$, usually depicted as -x, the existence of a matching $x' \in Licensors$, denoted by +X, is possible. *Base* includes at least the category c.

Definition 4.1 A five–tuple $\langle N_\tau, \triangleleft_\tau^*, \prec_\tau, <_\tau, label_\tau \rangle$ is called an *expression (over Feat)* if it fulfills (E1)–(E4).

(E1) $\langle N_\tau, \triangleleft_\tau^*, \prec_\tau \rangle$ is a finite, binary (ordered) tree defined in the usual sense: N_τ is the finite, non–empty set of *nodes*, and $\triangleleft_\tau^*$ and $\prec_\tau$ are the respective binary relations of *dominance* and *precedence* on N_τ.[6]

(E2) $<_\tau \subseteq N_\tau \times N_\tau$ is the asymmetric relation of *(immediate) projection* that holds for any two siblings in $\langle N_\tau, \triangleleft_\tau^*, \prec_\tau \rangle$.

(E3) $label_\tau$ is the *leaf–labeling function* from the set of leaves of $\langle N_\tau, \triangleleft_\tau^*, \prec_\tau \rangle$ into $Syn^* Phon^* Sem^*$.

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

We take $Exp(Feat)$ to denote the set of all expressions over *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 with $x \triangleleft_\tau^* head_\tau(x)$ such that each $y \in N_\tau$ on the path from x to $head_\tau(x)$ with $y \neq x$ projects over its sibling, i.e. $y <_\tau sibling_\tau(y)$.[8] 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 five–tuple $\upsilon = \langle N_\upsilon, \triangleleft_\upsilon^*, \prec_\upsilon, <_\upsilon, label_\upsilon \rangle$ is a *subexpression of τ* in case $\langle N_\upsilon, \triangleleft_\upsilon^*, \prec_\upsilon \rangle$ is a subtree of $\langle N_\tau, \triangleleft_\tau^*, \prec_\tau \rangle$, $<_\upsilon = <_\tau \restriction_{N_\upsilon \times N_\upsilon}$ and $label_\upsilon = label_\tau \restriction_{N_\upsilon}$. Thus, $\upsilon \in Exp(Feat)$. Such an υ 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 $sibling_\tau(x) <_\tau x$. $MaxProj(\tau)$ is the set of all maximal projections in τ.

$comp_\tau \subseteq MaxProj(\tau) \times MaxProj(\tau)$ is the binary relation defined 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 $\upsilon \, comp_\tau \, \phi$ holds 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 $\{\upsilon \mid \tau \, comp_\tau^+ \, \upsilon\}$.

---

[6] Thus, $\triangleleft_\tau^*$ is the reflexive–transitive closure of $\triangleleft_\tau \subseteq N_\tau \times N_\tau$, the relation of *immediate dominance* on N_τ

[7] A *tree domain* is a non–empty set $N_\upsilon \subseteq \mathbb{N}^*$ such that for all $\chi \in \mathbb{N}^*$ and $i \in \mathbb{N}$ it holds that $\chi \in N_\upsilon$ if $\chi\chi' \in N_\upsilon$ for some $\chi' \in \mathbb{N}^*$, and $\chi i \in N_\upsilon$ if $\chi j \in N_\upsilon$ for some $j \in \mathbb{N}$ with $i < j$. $\langle N_\upsilon, \triangleleft_\upsilon^*, \prec_\upsilon \rangle$ is the *natural (tree) interpretation* of N_υ in the case that for all $\chi, \psi \in N_\upsilon$ it holds that $\chi \triangleleft_\upsilon \psi$ iff $\psi = \chi i$ for some $i \in \mathbb{N}$, and $\chi \prec_\upsilon \psi$ 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 $i < j$.

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

$spec_\tau \subseteq MaxProj(\tau) \times MaxProj(\tau)$ is the binary relation defined such that 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 if the label assigned to υ's head by $label_\tau$ is non–empty and starts with an instance of $f \in Feat$.

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

The *phonetic yield of* τ, denoted by $Y_{Phon}(\tau)$, is 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 $label_\tau$, and replacing all instances of non–phonetic features with the empty string, afterwards.

An $\upsilon = \langle N_\upsilon, \triangleleft_\upsilon^*, \prec_\upsilon, <_\upsilon, label_\upsilon \rangle \in Feat(Exp)$ is *(label preserving) isomorphic to* τ if there is a bijective function i from N_τ onto N_υ with $x \triangleleft_\tau y$ iff $i(x) \triangleleft_\upsilon i(y)$, $x \prec_\tau y$ iff $i(x) \prec_\upsilon i(y)$, $x <_\tau y$ iff $i(x) <_\upsilon i(y)$, and with $label_\tau(x) = label_\upsilon(i(x))$ for $x, y \in N_\tau$. i is an *isomorphism (from* τ *to* υ*)*.

Definition 4.2 For $\tau = \langle N_\tau, \triangleleft_\tau^*, \prec_\tau, <_\tau, label_\tau \rangle \in Exp(Feat)$ with $N_\tau = tN_\upsilon$ for some $t \in \mathbb{N}^*$ and some tree domain N_υ, and for $r \in \mathbb{N}^*$, $(\tau)_r$ denotes the *expression shifting* τ *to* r, i.e. the expression $\langle N_{\tau(r)}, \triangleleft_{\tau(r)}^*, \prec_{\tau(r)}, <_{\tau(r)}, label_{\tau(r)} \rangle$ over $Feat$ with $N_{\tau(r)} = rN_\upsilon$ such that the function $i_{\tau(r)}$ from N_τ onto $N_{\tau(r)}$ with $i_{\tau(r)}(tx) = rx$ for all $x \in N_\upsilon$ is an isomorphism from τ to $(\tau)_r$.[9]

For $\upsilon, \phi \in Exp(Feat)$ let $\chi = \langle N_\chi, \triangleleft_\chi^*, \prec_\chi, <_\chi, label_\chi \rangle$ be a complex expression 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. Then χ is of one of two forms: in order to refer to χ we write $[_<\upsilon, \phi]$ if $0 <_\chi 1$, and $[_>\upsilon, \phi]$ if $1 <_\chi 0$.

Definition 4.3 [13] A *minimalist grammar (MG)* is a five–tuple of the form $\langle \neg Syn, Syn, Lex, \Omega, \mathtt{c} \rangle$, where Lex is a *lexicon (over Feat)*, i.e. a finite set of simple expressions over $Feat$ each of the form $\langle N_\tau, \triangleleft_\tau^*, \prec_\tau, <_\tau, label_\tau \rangle$ with $N_\tau = \{\epsilon\}$ and $label_\tau(\epsilon) \in (Select \cup Licensors)^* Base\, Licensees^* Phon^* Sem^*$, and where Ω is the operator set consisting of the structure building functions *merge* and *move* defined w.r.t. $Feat$ as in (me) and (mo) below, respectively.

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

(i) υ has selector $\mathtt{=x}$, and

[9] Note that, by (E4), for each $\tau = \langle N_\tau, \triangleleft_\tau^*, \prec_\tau, <_\tau, label_\tau \rangle \in Exp(Feat)$ a $t \in \mathbb{N}^*$ and a tree domain N_υ with $N_\tau = tN_\upsilon$ do exist.

(ii) ϕ has category `x`.

Then,

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

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

where υ' and ϕ' result from υ and ϕ, respectively, just by deleting the instance of the feature that the respective head–label starts with.

(mo) *move* is a partial mapping from $Exp(Feat)$ into $Exp(Feat)$. An expression $\upsilon \in Exp(Feat)$ is in $Dom(move)$ if for some `-x` $\in Licensees$ conditions (i)–(iii) are true:

(i) υ has licensor feature `+X`,

(ii) there is exactly one $\phi \in MaxProj(\upsilon)$ that has feature `-x`, and

(iii) there exists a $\chi \in Comp^{+}(\upsilon)$ with $\phi = \chi$ or $\phi \in Spec(\chi)$.

Then,

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

where $\upsilon' \in Exp(Feat)$ 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. $\phi' \in Exp(Feat)$ arises from ϕ by deleting the instance of `-x` the head–label of ϕ starts with.

Definition 4.4 [13] A *strict minimalist grammar (SMG)* is a five–tuple of the form $\langle \neg Syn, Syn, Lex, \Omega, \mathtt{c} \rangle$, where *Lex* is a finite set of expressions over *Feat* each of the form $\langle N_\tau, \triangleleft^*_\tau, \prec_\tau, <_\tau, label_\tau \rangle$ with $N_\tau = \{\epsilon\}$ and $label_\tau(\epsilon)$ is in $Select_\epsilon (Select \cup Licensors)_\epsilon Base\ Licensees^* Phon^* Sem^*$, and where Ω is the operator set consisting of the structure building functions *merge* and $move^{\mathrm{s}}$ defined w.r.t. *Feat* as in (me) above and (smo) below, respectively.

(smo) $move^{\mathrm{s}}$ is a partial mapping from $Exp(Feat)$ into $Exp(Feat)$. An expression $\upsilon \in Exp(Feat)$ is in $Dom(move)$ if for some `-x` $\in Licensees$ conditions (i)–(iii) are true:

(i) υ has licensor feature `+X`,

(ii) there is exactly one $\phi \in MaxProj(\upsilon)$ that has feature `-x`, and

(iii) there exists a $\chi \in Comp^{+}(\upsilon)$ with $\phi \in Spec^{*}(\chi)$. [10]

Then,

$move^{\mathrm{s}}(\upsilon) = [_{>}\chi', \upsilon']$,

where $\upsilon' \in Exp(Feat)$ results from υ by canceling the instance of `+X` the head–label of υ starts with, while the subtree χ is replaced by a single node labeled ϵ. $\chi' \in Exp(Feat)$ arises from χ by deleting the instance of `-x` the head–label of ϕ starts with.

[10] Note that such a $\chi \in Comp^{+}(\upsilon)$ is unique.

For each (S)MG $G = \langle \neg Syn, Syn, Lex, \Omega, \mathsf{c} \rangle$ the *closure of* G, $CL(G)$, is the set $\bigcup_{k \in \mathbb{N}} CL^k(G)$, where $CL^0(G) = Lex$, and for $k \in \mathbb{N}$, $CL^{k+1}(G) \subseteq Exp(Feat)$ is recursively defined as the set

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

where $move' \in \Omega \setminus \{merge\}$. $L(G)$ denotes the *(string) language derivable by* G, i.e. the set $\{Y_{Phon}(\tau) \mid \tau \in CL(G)$ and τ complete$\}$.

Definition 4.5 A set L is a *(strict) minimalist language ((S)ML)* if there exists an (S)MG G with $L = L(G)$.

5 (S)MLs as MCFLs

A method of transforming an MG as defined in [12] into an MCFG is presented in [4]. As demonstrated in [5], this method can be adapted to transform an (S)MG as defined in [13] into an $\mathrm{MCFG}_{1,2}$. But note that this adaptation is not of trivial kind, since in the original MG–definition *move* was defined as in (mo) above, but without condition (iii), i.e. a maximal projection could move completely independently of its position within an expression. Also, the handling of derivable tuples by means of the rewriting rules and functions has to be changed rather significantly in order to arrive at an MCFG as desired. [11]

6 MCFLs as (S)MLs

Throughout this section, $G = \langle N, O, F, R, S \rangle$ denotes an $\mathrm{MCFG}_{1,2}$ in the sense of Definition 2.5. In order to define an MG $G_{\mathrm{MG}} = \langle \neg Syn, Syn, Lex, \Omega, \mathsf{c} \rangle$ in the sense of Definition 4.3 such that $L(G_{\mathrm{MG}}) = L(G)$, we suppose w.l.o.g. G to be an $\mathrm{LCFRS}_{1,2}$ in MFF (cf. Proposition 3.2).

Of course, in [2] and [7] respective methods are presented how to construct, for an arbitrary MCFG, a weakly equivalent MG of the type originally given in [12]. Starting from an $\mathrm{MCFG}_{1,2}$, w.r.t. each of both methods, the lexicon of the resulting MG can even be interpreted as the lexicon of an MG in the sense of Definition 4.3 without leading to a change in the closure of the lexicon under the structure building functions. [12] A difference in the closure of the lexicon under the structure building functions may arise, however, if the lexicon of the MG resulting from the construction according to [7] is interpreted as the

[11] The respective considerations in [5] are even somewhat more involved than it would be necessary as to our concerns here: there, a corresponding transformation is given w.r.t. a type of an (S)MG which, in contrast to the definition in [13], still allows (overt) head movement and covert phrasal movement to take place. The "plain" case of transforming an MG in the proper sense of [13] into an $\mathrm{MCFG}_{1,2}$ is considered in [6].

[12] As far as the approach presented in [2] is concernced some slight modifications of the original construction are actually necessary before.

lexicon of an SMG, i.e. if the operator *move* is replaced by the operator $move^{\mathrm{s}}$ in order to build the corresponding closure.[13] This is not possible w.r.t. the MG G_{MG} which we develop here, since it fulfills (a) and (b) of Proposition 6.5, implying that the language derivable by G_{MG} is also an SML (Corollary 6.4). This result yields the interesting consequence that the class of MLs and that of SMLs are identical, confirming the corresponding conjecture in [13].

Motivating the construction below, let $A \to f(A_1, \dots, A_{n(f)}) \in R$ and $p_i \in L_G(A_i)$ for $1 \leq i \leq n(f)$, hence, $p = f(p_1, \dots, p_{n(f)}) \in L_G(A)$: our aim is to define G_{MG} such that there is some $\tau \in CL(G_{\mathrm{MG}})$ derivable from some expressions $\upsilon_1, \dots, \upsilon_{n(f)} \in CL(G_{\mathrm{MG}})$, thereby successively "calculating" the $\varphi(f)$–tuple p in $n(f) + 3\varphi(f) + \sum_{h=1}^{\varphi(f)} 2l_h(f)$ steps.[14] Each expression υ_i, for $1 \leq i \leq n(f)$, will be related to A_i and p_i, and the resulting expression τ to A and p in a specific way (cf. Definition 6.1). Roughly speaking, as for τ, for each $1 \leq h \leq d_G(A)$ there is a $\tau_h \in MaxProj(\tau)$ that has a particular licensee, and up to the phonetic yields of the proper subtrees potentially extractable from τ_h, p's component p_h is the phonetic yield of τ_h.

•• Let $Phon = \Sigma$ and $Sem = \emptyset$.

•• For $1 \leq h \leq m$ and $0 \leq n \leq 1$ let $\texttt{-l}_{\langle h,\, n\rangle}$ be a licensee and $\texttt{+L}_{\langle h,\, n\rangle}$ the matching licensor such that *Licensees* and *Licensors* both have cardinality $2m$.[15]

•• For each $A \in N$ introduce new, pairwise distinct basic categories $\tilde{\texttt{a}}$ and $\texttt{a}_h$ as well as corresponding selectors $\texttt{=}\tilde{\texttt{a}}$ and $\texttt{=a}_h$ for $1 \leq h \leq d_G(A)$. For each $A \to f(A_1, \dots, A_{n(f)}) \in R$ 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 selectors $\texttt{=a}_{\langle f,\, \varphi(f)+1,\, 0\rangle}$ and $\texttt{=a}_{\langle f,\, h,\, l\rangle}$, where $1 \leq h \leq \varphi(f)$ and $0 \leq l \leq l_h(f)$.[14] Finally, assume $\texttt{c} \in Base$ to be different from all other elements in *Base*.

•• Next we define $Lex \subseteq Exp(\neg Syn \cup Syn)$.[16] The first item defined to belong to *Lex* is

$$\alpha_{\texttt{c}} \;=\; \texttt{=}\tilde{\texttt{s}}\,\texttt{c}\,,$$

where $\tilde{\texttt{s}} \in Base$ is the corresponding category arising from $S \in N$. The form of all other items in *Lex* depends on the rules in R. 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 $\varphi(f) = d_G(A)$, $n(f) = 2$, $d_1(f) = d_G(B) = 1$ and $d_2(f) = d_G(C)$. The following elements belong to *Lex*:

$$\alpha_{\langle A,\, f,\, B,\, C\rangle} \;=\; \texttt{=c}_1\,\texttt{a}_{\langle f,\, \varphi(f)+1,\, 0\rangle}$$

[13] Note that the use of multiple specifiers is of rather constitutive moment within the approach of Harkema [2], causing that, in particular, the MG–lexicon which results from an $\mathrm{MCFG}_{1,2}$ according to his construction does generally not match the SMG–definition.
[14] Recall that $\varphi(f) = d_G(A)$ by (M3).
[15] Here, $m = \max\{d_G(A) \,|\, A \in N\}$.
[16] Since it is a head with root ϵ, we identify a lexical item with its (unique) label.

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

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

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

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

$$\alpha_{\langle A,f,h,l_h(f)\rangle} = \begin{cases} {}^{=}\mathtt{a}_{\langle f,h+1,0\rangle}\,{}^{=}\tilde{\mathtt{b}}\;\mathtt{a}_{\langle f,h,l_h(f)\rangle}\,\zeta(f_{hl_h(f)}) & \text{if } z(f_{hl_h(f)}) = x_{11} \\ {}^{=}\mathtt{a}_{\langle f,h+1,0\rangle}\,\mathtt{+L}_{\langle j,1\rangle}\,\mathtt{a}_{\langle f,h,l_h(f)\rangle}\,\zeta(f_{hl_h(f)}) & \text{otherwise,} \\ \qquad \text{where } 1 \le j \le d_2(f) \text{ with } z(f_{hl_h(f)}) = x_{2j}{}^{17} & \end{cases}$$

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

$$\alpha_{\langle A,f,h,l\rangle} = \begin{cases} {}^{=}\mathtt{a}_{\langle f,h,l+1\rangle}\,{}^{=}\tilde{\mathtt{b}}\;\mathtt{a}_{\langle f,h,l\rangle}\,\zeta(f_{hl}) & \text{if } z(f_{hl}) = x_{11} \\ {}^{=}\mathtt{a}_{\langle f,h,l+1\rangle}\,\mathtt{+L}_{\langle j,1\rangle}\,\mathtt{a}_{\langle f,h,l\rangle}\,\zeta(f_{hl}) & \text{otherwise,} \\ \qquad \text{where } 1 \le j \le d_2(f) \text{ with } z(f_{hl(f)}) = x_{2j}{}^{17} & \end{cases}$$

Case 2. $A \to f(B)$ 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, the following element belongs to *Lex*:

$$\alpha_{\langle A,f,B,-\rangle} = {}^{=}\mathtt{b}_1\,\mathtt{a}_{\langle f,\varphi(f)+1,0\rangle}$$

Case 3. $A \to f()$ for some $A \in N$ and $f \in F$. Then $\varphi(f) = d_G(A)$ and $n(f) = 0$. $l_h(f) = 0$ for $1 \le h \le \varphi(f)$, i.e. $f() = \langle \zeta(f_{10}), \ldots, \zeta(f_{\varphi(f)0})\rangle$, since f is a constant in $\langle \Sigma^* \rangle^{\varphi(f)}$. The following entry belong *Lex*:

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

In Case 2 and 3, also the following items are in *Lex*:

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

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

$$\alpha_{\langle A,f,h,l_h(f)\rangle} = {}^{=}\mathtt{a}_{\langle f,h+1,0\rangle}\,\mathtt{+L}_{\langle j,1\rangle}\,\mathtt{a}_{\langle f,h,l_h(f)\rangle}\,\zeta(f_{hl_h(f)}) \;, \quad \text{where } 1 \le j \le d_1(f) \text{ such that } z(f_{hl_h(f)}) = x_{1j}.$$

[17] Since $d_1(f) = 1$, such a j exists and is unique.

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

$$\alpha_{\langle A,\, f,\, h,\, l\rangle} \;=\; {}^{=}\mathtt{a}_{\langle f,\, h,\, l+1\rangle}\, \mathtt{+L}_{\langle j,\, 1\rangle} \mathtt{a}_{\langle f,\, h,\, l\rangle}\, \zeta(f_{hl}) \;, \quad \text{where } 1 \leq j \leq d_1(f) \text{ such that } z(f_{hl}) = x_{1j}.$$

<u>In Case 1–3</u>, finally the following items are in *Lex*:

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

for each $1 \leq h < \varphi(f)$ the simple expression

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

and as last item the expression

$$\widetilde{\alpha}_{\langle A,\, 1\rangle} \;=\; \begin{cases} {}^{=}\mathtt{a}_{\langle f,\, 1,\, 0\rangle}\, \mathtt{+L}_{\langle 1,\, 0\rangle}\, \tilde{\mathtt{a}} & \text{if } \varphi(f) = 1 \\ {}^{=}\mathtt{a}_2\, \mathtt{+L}_{\langle 1,\, 0\rangle}\, \tilde{\mathtt{a}} & \text{otherwise} \end{cases}$$

Definition 6.1 For each $A \in N$ and each $p = \langle \pi_1, \ldots, \pi_{d_G(A)}\rangle$ with $\pi_i \in \Sigma^*$ for $1 \leq i \leq d_G(A)$ an expression $\tau \in CL(G_{\text{MG}})$ is said to *correspond to* the pair $\langle A, p\rangle$ if (Y1)–(Y4) are fulfilled, where $\tau_1 = \tau$.

(Y1) τ's head–label is of the form $\mathtt{a}_1\mathtt{-l}_{\langle 1,\, 1\rangle}\pi_{\langle 1,\, 1\rangle}$ or $\tilde{\mathtt{a}}\,\pi_{\langle 1,\, 1\rangle}$ for a $\pi_{\langle 1,\, 1\rangle} \in \Sigma^*$.

(Y2) For each $2 \leq h \leq d_G(A)$ there is exactly one $\tau_h \in \mathit{Comp}^+(\tau)$ whose head–label is of the form $\mathtt{-l}_{\langle h,\, 1\rangle}\,\pi_{\langle h,\, 1\rangle}$ for some $\pi_{\langle h,\, 1\rangle} \in \Sigma^*$.

(Y3) For each $1 \leq h \leq d_G(A)$ it holds that

$$\{\upsilon \in \mathit{MaxProj}(\tau_h) \setminus \{\tau_h\} \,|\, \upsilon \text{ has some licensee}\} = \{\tau_i \,|\, h < i \leq d_G(A)\},$$

i.e. for each $1 \leq h < d_G(A)$ the subexpression τ_{h+1} is the unique maximal maximal projection in τ_h that has some licensee feature.

(Y4) For each $1 \leq h \leq d_G(A)$ the string π_h is the phonetic yield of υ_h. Here we have $\upsilon_{d_G(A)} = \tau_{d_G(A)}$, and for $1 \leq h < d_G(A)$ the expression υ_h results from τ_h by replacing the subtree τ_{h+1} with a single node labeled ϵ.

Proposition 6.2 *For each $\tau \in CL(G_{\text{MG}})$ that has category feature $\mathtt{a}_1$ or $\tilde{\mathtt{a}}$ for some $A \in N$, there is some $p \in L_G(A)$ such that τ corresponds to $\langle A\,, p\rangle$.*

Proof (sketch). We exclude the trivial case by assuming that there is some $\tau \in CL(G_{\text{MG}})$ such that τ has category $\mathtt{a}_1$ or $\tilde{\mathtt{a}}$ for some $A \in N$. We take $K \in \mathbb{N}$ to be the smallest number, thereby existing, for which $CL^K(G_{\text{MG}})$ includes such a τ. The definition of *Lex* leaves us with the fact that $K > 0$.

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 $\tau \in CL^{k+1}(G_{\text{MG}}) \setminus CL^{k}(G_{\text{MG}})$ such that τ has category $\mathtt{a}_1$ or $\tilde{\mathtt{a}}$ for some $A \in N$. By definition of *Lex* the procedure to derive τ as an expression of G_{MG} is deterministic in the following sense: there are some $r = A \rightarrow f(A_1, \ldots, A_{n(f)}) \in R$, some $k_0 \in \mathbb{N}$ with $k_0 = k + 1 - 3\varphi(f) - \sum_{h=1}^{\varphi(f)} 2l_h(f)$ and some $\chi_0 \in CL^{k_0}(G_{\text{MG}})$ 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:

<u>Case 1</u>. There is some $r = A \rightarrow f(B, C) \in R$, there is some $\upsilon \in CL^{k_1}(G_{\text{MG}})$ for some $k_0 \leq k_1 \leq k$, and there is some $\phi \in CL^{k_0}(G_{\text{MG}})$ such that υ and ϕ have category feature $\tilde{\mathtt{b}}$ and $\mathtt{c}_1$, respectively, and

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

More explicitly, k_1 can be specified by

$$k_1 = k_0 + 2l + 1 + h + \textstyle\sum_{h'=0}^{h-1} 2l_{\varphi(f)-h'}(f)$$

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

$$g_f^{-1}(\varphi(f) - h, l_{\varphi(f)-h}(f) - l) = \langle 1, 1\rangle.$$

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 $\langle B, p_B\rangle$ and $\langle C, p_C\rangle$, respectively. In this case we define $p \in L_G(A)$ by $p = f(p_B, p_C)$. Note that $p_B \in \Sigma^*$ by assumption on G.

<u>Case 2</u>. There are some $r = A \rightarrow f(B) \in R$ and $\upsilon \in CL^{k_0}(G_{\text{MG}})$ such that υ has category feature $\mathtt{b}_1$, and such that

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

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

<u>Case 3</u>. There is some $r = A \rightarrow f() \in R$ and χ_0 is a lexical item, namely,

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

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

If $k + 1 = K$ (the base case of our induction) then χ_0 is necessarily of the last form by choice of K. We also see that the given $\tau \in CL^{k+1}(G_{\text{MG}}) \setminus CL^{k}(G_{\text{MG}})$ corresponds to $\langle A, p\rangle$ in any case. The single derivation steps in order 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 $0 \leq l < l_{\varphi(f)-h}(f)$

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

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

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

if $\varphi(f) = 2$, and if $g_f^{-1}(\varphi(f)-h, l_{\varphi(f)-h}(f)-l) = \langle 1,1\rangle$ then

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

else

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

[checks licensee $\texttt{-l}_{\langle j,\,1\rangle}$ with $g_f(2,j) = \langle \varphi(f)-h, l_{\varphi(f)-h}(f)-l\rangle$]

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

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

For $0 \leq h < \varphi(f) - 1$

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

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

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

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

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

<u>Either</u>

step $3\varphi(f) - 1 + \sum_{h'=1}^{\varphi(f)} 2l_{h'}(f)$

$\chi_{\varphi(f)+2\varphi(f)-1} = merge(\alpha_{\langle A,1\rangle}, \chi_{3\varphi(f)-2})$

step $3\varphi(f) + \sum_{h'=1}^{\varphi(f)} 2l_{h'}(f)$

$\chi_{3\varphi(f)} = move(\chi_{3\varphi(f)-1})$
[checks licensee $\texttt{-l}_{\langle 1,0\rangle}$]

<u>or</u>

step $3\varphi(f) - 1 + \sum_{h'=1}^{\varphi(f)} 2l_{h'}(f)$

$\chi_{3\varphi(f)-1} = merge(\widetilde{\alpha}_{\langle A,1\rangle}, \chi_{3\varphi(f)-2})$

step $3\varphi(f) + \sum_{h'=1}^{\varphi(f)} 2l_{h'}(f)$

$\chi_{3\varphi(f)} = move(\chi_{3\varphi(f)-1})$
[checks licensee $\texttt{-l}_{\langle 1,0\rangle}$]

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

An embedded induction on $0 \leq h < \varphi(f)$ and $0 \leq l < l_{\varphi(f)-h}(f)$ yields that τ indeed corresponds to $\langle A, p\rangle$. We omit further details at this point. □

Proposition 6.3 *Let $A \in N$ and $p \in \langle \Sigma^* \rangle^{d_G(A)}$ with $p \in L_G(A)$. Furthermore let $x \in \{\mathtt{a}_1, \tilde{\mathtt{a}}\}$. Then there is some $\tau \in CL(G_{\text{MG}})$ that has category feature x such that τ corresponds to $\langle A, p\rangle$.*

Proof (sketch). Let $A \in N$ and $p \in \langle \Sigma^* \rangle^{d_G(A)}$ such that $p \in L_G(A)$. Then, again w.l.o.g., we are concerned with one of three possible cases.

<u>Case 1</u>. There is some $r = A \to f(B, C) \in R$, 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 $p = f(p_B, p_C) \in L_G^{k+1}(A) \backslash L_G^k(A)$. By hypothesis on k there are some $\upsilon, \phi \in CL(G_{\text{MG}})$ such that υ and ϕ have category feature $\tilde{\mathtt{b}}$ and $\mathtt{c}_1$, respectively, and such that υ and ϕ correspond to $\langle B, p_B\rangle$ and $\langle C, p_C\rangle$, respectively. We can therefore define $\chi_0 \in CL(G_{\text{MG}})$ by

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

Note that we have $d_G(B) = 1$ by assumption on G.

Case 2. There is some $r = A \to f(B) \in R$, and for some $k \in \mathbb{N}$ there is some $p_B \in L_G^k(B)$ such that $p = f(p_B) \in L_G^{k+1}(A) \setminus L_G^k(A)$. Here, by induction hypothesis we can choose some $\upsilon \in CL(G_{\text{MG}})$ such that υ has category feature $\mathtt{b}_1$ and corresponds to $\langle B, p_B\rangle$. Then we define $\chi_0 \in CL(G_{\text{MG}})$ by

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

Case 3. There is some $r = A \to f() \in R$ such that $p = f() \in L_G^0(A)$. In this case χ_0 is taken to be the lexical item defined by

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

In any case we may refer to the proof of the last proposition, claiming that there is some derivation in G_{MG} in which χ_0 serves to derive a $\tau \in CL(G_{\text{MG}})$ which has the demanded properties. □

Corollary 6.4 *$\pi \in L(G)$ iff $\pi \in L(G_{\text{MG}})$ for each $\pi \in \Sigma^*$.*

Proof. To see that the "if–part" holds, consider $\tau \in CL(G_{\text{MG}})$ which is complete, and whose phonetic yield is π for some $\pi \in \Sigma^*$. By definition of *Lex* there is some $\tau' \in CL(G_{\text{MG}})$ which has category $\tilde{\mathtt{s}}$ such that $\tau = merge(\alpha_{\mathtt{C}}, \tau')$. By Proposition 6.2 there is some $p' \in L_G(S) = L(G)$ such that τ' corresponds to $\langle S, p'\rangle$. Because $d_G(S) = 1$, and because the phonetic yield of $\alpha_{\mathtt{C}}$ is empty, we have $p' = \pi$.

In order to prove the "only if–part," assume that $\pi \in L(G) = L_G(S)$ for some $\pi \in \Sigma^*$. By Proposition 6.3 there is some $\tau' \in CL(G_{\text{MG}})$ with category feature $\tilde{\mathtt{s}}$ such that τ' corresponds to $\langle S, \pi\rangle$. Then, because it holds that $d_G(S) = 1$, $\tau = merge(\alpha_{\mathtt{C}}, \tau')$ is defined and complete, and π is the phonetic yield of τ. □

Proposition 6.5 *G_{MG} fulfills (a) and (b).*

(a) For each $\alpha \in Lex$, the (unique) label of α is in particular an element of $Select_\epsilon(Select \cup Licensors)_\epsilon Base Licensees^* Phon^* Sem^*$, since it is even of the form $s_1 s_2 \mathtt{x} \lambda \pi \iota$ for some $s_1 \in Select_\epsilon$, $s_2 \in (Select \cup Licensors)_\epsilon$, $\mathtt{x} \in Base$, $\lambda \in Licensees_\epsilon$ and $\pi\iota \in \neg Syn^*$.

(b) Whenever, for a given $\upsilon \in CL(G_{\text{MG}})$ and $\mathtt{-x} \in Licensees$, there is some $\phi \in MaxProj(\upsilon)$ that has licensee $\mathtt{-x}$ then $\phi \in Comp^+(\upsilon)$.

Proof (sketch). Property (a) is true due to the definition of *Lex*.

The validity of (b) arises from the combination of several facts. First, each expression $\chi \in CL(G_{\text{MG}})$ serves to derive a complete expression of G_{MG}. In this sense we may say that $CL(G_{\text{MG}})$ contains no "useless" expressions. This in turn implies that each $\chi \in CL(G_{\text{MG}})$ is subject to one of the following possibilities:

(i) $\chi \in \mathit{Lex}$.

(ii) There are some $k \in \mathbb{N}$ and $r = A \to f(A_1, \dots, A_{n(f)}) \in R$, and some $\tau \in CL^{k+1}(G_{\text{MG}})$ and $\chi_0 \in CL^{k_0}(G_{\text{MG}})$ with $k_0 = k - 3\varphi(f) - \sum_{h=1}^{\varphi(f)} 2l_h(f)$ such that χ_0 serves to derive τ according to a respective procedure from above, and such that χ is derived within this procedure in order to finally end up with τ.

(iii) χ is complete, i.e. there is some $\tau' \in CL(G_{\text{MG}})$ that has category feature $\tilde{\mathtt{s}}$ such that $\chi = \mathit{merge}(\alpha_{\mathtt{c}}, \tau')$.

As far as expressions of G_{MG} which result from an application of the merge–operator are concerned, property (b) is therefore guaranteed by the fact that an expression which is merged into a specifier position contains no maximal projection that has any licensee feature, since this expression has category feature $\tilde{\mathtt{b}}$ for some $B \in N$ with $d_G(B) = 1$.

As it regards expressions that result from an application of the move–operator, property (b) essentially results from our assumption that G is in MFF, and from a further fact concerning the licensees of the form $\mathtt{-l}_{\langle h,0\rangle}$ for some $1 \leq h \leq m$: whenever for some expression $\chi \in CL(G_{\text{MG}})$ and some $1 \leq h \leq m$ there is some $\tau_{\langle h,0\rangle} \in \mathit{MaxProj}(\chi)$ that has licensee $\mathtt{-l}_{\langle h,0\rangle}$, each $\tau_{\langle i,0\rangle} \in \mathit{MaxProj}(\chi)$ that has some licensee $\mathtt{-l}_{\langle i,0\rangle}$ with $h < i \leq m$ belongs to $\mathit{Comp}^+(\tau_{\langle h,0\rangle})$. □

Corollary 6.6 *The language $L(G_{\text{MG}})$ is an ML as well as an SML.*

Proof. This corollary is an immediate consequence of the preceding proposition. Note, in particular, that from (b) of Proposition 6.5 it follows that the closure of *Lex* under the structure building operators is the same set of expressions over $\neg\mathit{Syn} \cup \mathit{Syn}$ independently of whether the move–operator is defined as in (mo) or (smo). □

7 AFL–Properties

In [13] three further problems concerning the properties of the revised MG–type have been explicitly left open. All three fall under a more general question: does the class of MLs constitute an *abstract family of languages (AFL)*? In fact a stronger result holds:

Proposition 7.1 *The class of all MLs is a substitution–closed full AFL.*

Proof. Because the class of all MLs is, as shown above, identical to the class of all $\text{MCFL}_{1,2}$'s, we can likewise prove that the latter is a substitution–closed full AFL: it straightforwardly follows from the definition that the class of all

$MCFL_{1,2}$'s includes all regular sets and is closed under substitution. Thus, by Theorem 1.6 [10, p. 129] it remains to confirm that the class of all $MCFL_{1,2}$'s is closed under intersection with regular sets. But exactly this is done implicitly within the proof which Seki et al. [11] give verifying their Theorem 3.9. □

References

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

[2] Henk Harkema. A characterization of minimalist languages. In de Groote et al. [1], pages 193–211.

[3] Hilda Koopman and Anna Szabolcsi. *Verbal Complexes*. MIT Press, Cambridge, MA, 2000.

[4] 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, pages 179–198. Springer, Berlin, Heidelberg, 2001. Also available at `http://www.ling.uni-potsdam.de/~michael/papers.html`.

[5] Jens Michaelis. *On Formal Properties of Minimalist Grammars*. PhD thesis, Potsdam University, Potsdam, 2001.

[6] Jens Michaelis. Some formal implications of a revised perspective on minimalist grammars. Draft, Potsdam University, 2001. Available at `http://www.ling.uni-potsdam.de/~michael/papers.html`.

[7] Jens Michaelis. Transforming linear context–free rewriting systems into minimalist grammars. In de Groote et al. [1], pages 228–244. Also available at `http://www.ling.uni-potsdam.de/~michael/papers.html`.

[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] Arto Salomaa. *Formal Languages*. Academic Press, New York, NY, 1973.

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

[12] 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.

[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] 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.

[15] David J. Weir. *Characterizing Mildly Context–Sensitive Grammar Formalisms.* PhD thesis, University of Pennsylvania, Philadelphia, PA, 1988.