

A Note on the Complexity of Constraint Interaction: Locality Conditions and Minimalist Grammars^{*}

Hans-Martin Gärtner¹ and Jens Michaelis²

¹ ZAS, Jägerstr. 10–11, 10117 Berlin, Germany
gaertner@zas.gwz-berlin.de

² Universität Potsdam, Institut für Linguistik, PF 601553, 14415 Potsdam, Germany
michael@ling.uni-potsdam.de

Abstract. *Locality Conditions (LCs)* on (unbounded) dependencies have played a major role in the development of generative syntax ever since the seminal work by Ross [22]. Descriptively, they fall into two groups. On the one hand there are *intervention-based LCs (ILCs)* often formulated as “minimality constraints” (“minimal link condition,” “minimize chain links,” “shortest move,” “attract closest,” etc.). On the other hand there are *containment-based LCs (CLCs)* typically defined in terms of (generalized) grammatical functions (“adjunct island,” “subject island,” “specifier island,” etc.). Research on LCs has been dominated by two very general trends. First, attempts have been made at unifying ILCs and CLCs on the basis of notions such as “government” and “barrier” (e.g. [4]). Secondly, research has often been guided by the intuition that, beyond empirical coverage, LCs somehow contribute to restricting the formal capacity of grammars (cf. [3, p. 125], [6, p. 14f]). Both these issues, we are going to argue, can be fruitfully studied within the framework of *minimalist grammars (MGs)* as defined by Stabler [25]. In particular, we are going to demonstrate that there is a specific asymmetry between the influence of ILCs and CLCs on complexity. Thus, MGs, including an ILC, namely, the *shortest move condition (SMC)* have been shown to belong to the *mildly context-sensitive* grammar formalisms by Michaelis [14]. The same has been shown in [16, 18] for a revised version of MGs introduced in [26], which includes the SMC and an additional CLC, namely, the *specifier island condition (SPIC)*. In particular [14] and [16, 18] show that, in terms of derivable string languages, both the original MG-type and the revised MG-type constitute a subclass of the class of *linear context-free rewriting systems (LCFRSs)* in the sense of [28, 29], and thus, a series of other formalism classes all generating the same class of string languages as LCFRSs. Here we will demonstrate that removing the SMC from the revised MG-version increases the generative power in such a way that the resulting formalism is not mildly context-sensitive anymore. This suggests that intuitions to the contrary notwithstanding, imposing an LC as such, here the SPIC, does not necessarily reduce formal complexity.

^{*} This work has been carried out partially funded by DFG-grant. Thanks to two anonymous referees for valuable comments on a previous version of this paper.

1 Introduction

Locality Conditions (LCs) on (unbounded) dependencies have played a major role in the development of generative syntax ever since the seminal work by Ross [22]. Descriptively, they fall into two groups. On the one hand there are *intervention-based LCs (ILCs)*, very sketchily illustrated in (1).

$$[\dots \alpha \dots [\dots \beta \dots \gamma \dots]] \quad (1)$$

An ILC constrains dependencies between α and γ across an intervening β , where intervention is defined in terms of *c-command* as well as the features and position of β relative to the features and positions of α and γ . ILCs are often formulated as “minimality constraints” (“minimal link condition,” “minimize chain links,” “shortest move,” “attract closest,” etc.) on the assumption that more minimal dependencies could have been formed between α and β , and/or β and γ . On the other hand *containment-based LCs (CLCs)* exist such as given in (2).

$$[\dots \alpha \dots [\beta \dots \gamma \dots]] \quad (2)$$

A CLC constrains dependencies between α and γ across a constituent β containing γ but excluding α . Typically the containers are defined in terms of (generalized) grammatical functions (“adjunct island,” “subject island,” “specifier island,” etc.). Research on LCs has been dominated by two very general trends. First, attempts have been made at unifying ILCs and CLCs on the basis of notions such as “government” and “barrier” (see e.g. [4]). Secondly, research has often been guided by the intuition that, beyond empirical coverage, LCs somehow contribute to restricting the formal capacity of grammars (cf. [3, p. 125], [6, p. 14f]). Both these issues, we are going to argue, can be fruitfully studied within the framework of *minimalist grammars (MGs)* as defined by Stabler [25].³ In particular, we are going to show that there is a specific asymmetry between the influence of ILCs and CLCs on complexity. Crucial starting point for our demonstration is the fact that MGs, including an ILC, namely, the *shortest move condition (SMC)* have been shown to belong to the *mildly context-sensitive*

³ Research on LCs in terms of *categorial grammar (CG)* has taken at least two rather divergent directions. Within the *combinatory* brand of CG, *CCG*, there has been a tendency to refrain from syntactically encoding LCs. This is based on the intuition that “the origin of constraints on long-range dependencies ultimately lies in semantic coherence properties of the nonstandard constituents that combination into islands creates [...]” [27, p. 65]. The *type logical* approach to CG has implemented CLCs rather directly in terms of unary operators that “block associativity” [19] or cause “structural inhibition” [20]. For one complexity result on the resulting *multi-modal CGs*, showing that under certain conditions they preserve the weak context-freeness of the Lambek calculus, see [9]. It would be attractive to compare and contrast our results on MGs more closely with CG-approaches. (In particular, it might be worth to explore hybrid versions of both CG-brands, especially given the fact that CCGs undergoing certain restrictions have been shown to constitute a mildly context-sensitive formalism (cf. [28, 29]), and thus, to be rather closely related to MGs.) However, this will have to be left for further research.

grammar formalisms by Michaelis [14]. The same has been shown in [16] for a revised version of MGs defined in [26], which includes the SMC and an additional CLC, namely, the *specifier island condition (SPIC)*.

The type of MG introduced in [25] provides an attempt at a rigorous algebraic formalization of the perspectives currently adopted within the linguistic framework of transformational grammar (see e.g. [5]). An MG, roughly speaking, is a formal device which specifies a countable set of finite, binary (ordered) trees each being equipped with a leaf-labeling function assigning a string of features to each leaf, and with an additional binary relation, the asymmetric relation of (*immediate*) *projection*, defined on the set of pairs of siblings. The base of an MG is formed by a *lexicon* (a finite set of single node trees in the above sense) and two structure building functions: *merge* (combining two trees) and *move* (transforming a given tree). Both functions build structure by canceling particular matching instances of features within the leaf-labels of the trees to which they are applied. The closure of the lexicon under these two functions is the set of trees characterized by the MG. As shown in [14], this MG-type constitutes a mildly context-sensitive formalism in the sense that it provides a weakly equivalent subclass of *linear context-free rewriting systems (LCFRSs)* [28, 29]. Independent work in [8] and [17] has proven the reverse to hold as well. Hence, MGs as defined in [25], beside LCFRSs, join to a series of formalism classes—among which there are e.g. the class of *local unordered scattered context grammars* [21], the class of *multicomponent tree adjoining grammars* in their set-local variant of admitted adjunction (cf. [29]), the class of *multiple context-free grammars* [24], or the class of *simple positive range concatenation grammars* [1]—all generating the same class of string languages. For a list of some further of such classes of generating devices see e.g. [21].

Inspired, i.a., by the linguistic work presented in [12], in [26] a revised type of an MG has been proposed whose departure from the version in [25] 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 regulating which maximal projection may move *overtly* into the highest specifier position. Deviating from the operator *move* as originally defined in [25], a constituent has to belong to the transitive complement closure of a given tree or to be a specifier of such a constituent in order to be movable at all. Employing and extending the methods developed in [14] and [17], it was shown in [16, 18] and [15] that, in terms of derivable string languages, the revised MG-type is not only subsumed by LCFRSs, but is identical to a particular subclass of the latter: the righthand side of each rewriting rule of a corresponding LCFRS involves at most two nonterminals, and if two nonterminals appear on the righthand side then only simple strings of terminals are derivable from the first one. Whether the respective classes of string languages derivable by the corresponding LCFRS-subclass and the class of all LCFRSs—and thus the respective classes of string languages derivable by the class of revised MGs as defined in [26] and the class of MGs as defined in [25]—are identical seems to be an open problem. Here, we will demonstrate that removing the SMC from

the revised MG-version increases the generative power in such a way that the resulting formalism is not mildly context-sensitive. This suggests that intuitions to the contrary notwithstanding, imposing an LC as such, here the SPIC, does not necessarily reduce formal complexity.

The paper is structured as follows: Section 2 introduces (revised) minimalist grammars in two versions, one standard (MGs, Definition 3) and one without the SMC ($\text{MG}^{\text{-SMC}}$ s, Definition 4). In Section 3 we show, how to derive the language $\{a^{2^n} \mid n \in \mathbb{N}\}$, i.e. a language without the constant growth property, and thus a non-mildly context-sensitive language. Section 4 provides a short conclusion and an outlook on further research. Throughout the rest of the paper we use the term *minimalist grammar* and its abbreviation *MG* in order to refer to an MG of the revised type as defined in [26], unless explicitly indicated otherwise.

2 (Revised) Minimalist Grammars

Throughout we let $\neg\text{Syn}$ 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 $\neg\text{Syn} \cup \text{Syn}$.

- (F1) $\neg\text{Syn}$ is disjoint from Syn and partitioned into a set *Phon* of *phonetic features* and a set *Sem* of *semantic features*.
- (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 $x \in \text{Base}$, usually typeset as \mathbf{x} , the existence of a matching $x' \in \text{Select}$, denoted by $\bar{\mathbf{x}}$, is possible. For each $x \in \text{Licensees}$, usually depicted as $-\mathbf{x}$, the existence of a matching $x' \in \text{Licensors}$, denoted by $+\mathbf{x}$, is possible. *Base* includes at least the category *c*.

Definition 1. An *expression* (over *Feat*) is a five-tuple $\langle N_\tau, \triangleleft_\tau^*, \prec_\tau, <_\tau, \text{label}_\tau \rangle$ obeying (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_τ^* and \prec_τ are the respective binary relations of *dominance* and *precedence* on N_τ .⁴
- (E2) $<_\tau \subseteq N_\tau \times N_\tau$ is the asymmetric relation of (*immediate*) *projection* that holds for any two siblings in $\langle N_\tau, \triangleleft_\tau^*, \prec_\tau \rangle$, i.e., for each $x \in N_\tau$ different from the root of $\langle N_\tau, \triangleleft_\tau^*, \prec_\tau \rangle$ either $x <_\tau \text{ sibling}_\tau(x)$ or $\text{ sibling}_\tau(x) <_\tau x$ holds.⁵

⁴ Thus, \triangleleft_τ^* is the reflexive-transitive closure of $\triangleleft_\tau \subseteq N_\tau \times N_\tau$, the relation of *immediate dominance* on N_τ

⁵ For each $x \in N_\tau$ different from the root of $\langle N_\tau, \triangleleft_\tau^*, \prec_\tau \rangle$, $\text{ sibling}_\tau(x)$ is the (unique) sibling of x . If $x <_\tau y$ for any $x, y \in N_\tau$, x is said to (*immediately*) *project over* y .

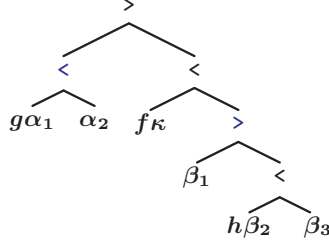


Fig. 1. A typical expression over *Feat*.

- (E3) $label_\tau$ is the *leaf-labeling function*, i.e., a total function from the set of all leaves of $\langle N_\tau, \triangleleft_\tau^*, \prec_\tau \rangle$ into $Syn^*Phon^*Sem^*$.⁶
- (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 $\tau = \langle N_\tau, \triangleleft_\tau^*, \prec_\tau, <_\tau, label_\tau \rangle \in Exp(Feat)$.⁸

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. $y <_\tau sibling_\tau(y)$. 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 $v = \langle N_v, \triangleleft_v^*, \prec_v, <_v, label_v \rangle$ is a *subexpression* of τ if $\langle N_v, \triangleleft_v^*, \prec_v \rangle$ is a subtree of $\langle N_\tau, \triangleleft_\tau^*, \prec_\tau \rangle$, and if $<_v = <_\tau \upharpoonright_{N_v \times N_v}$ and $label_v = label_\tau \upharpoonright_{N_v}$ hold.⁹ Thus, $v \in Exp(Feat)$. Such an v is a *maximal projection* (*in* τ) if v '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 .

⁶ For each set M , M^* is the Kleene closure of M , including ϵ , the empty string. $M\epsilon$ denotes the set $M \cup \{\epsilon\}$. For any $K, L \subseteq M^*$, KL is the product of K and L under concatenation, i.e., the (string) set $\{kl \mid k \in K, l \in L\} \subseteq M^*$.

⁷ We take \mathbb{N} to denote the set of all non-negative integers. A *tree domain* is a non-empty set $N_v \subseteq \mathbb{N}^*$ such that for all $\chi \in \mathbb{N}^*$ and $i \in \mathbb{N}$ it holds that $\chi \in N_v$ if $\chi\chi' \in N_v$ for some $\chi' \in \mathbb{N}^*$, and $\chi i \in N_v$ if $\chi j \in N_v$ for some $j \in \mathbb{N}$ with $i < j$. $\langle N_v, \triangleleft_v^*, \prec_v \rangle$ is the *natural (tree) interpretation* of N_v in the case that for all $\chi, \psi \in N_v$ it holds that $\chi \triangleleft_v \psi$ iff $\psi = \chi i$ for some $i \in \mathbb{N}$, and $\chi \prec_v \psi$ 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 Note that the leaf-labeling function $label_\tau$ can easily be extended to a total labeling function ℓ_τ from N_τ into $Feat^* \cup \{<, >\}$, where $<$ and $>$ are two new distinct symbols: to each non-leaf $x \in N_\tau$ we can assign a label from $\{<, >\}$ by ℓ_τ such that $\ell_\tau(x) = <$ iff $y <_\tau z$ for $y, z \in N_\tau$ with $x \triangleleft_\tau y$, $x \triangleleft_\tau z$, and $y \prec_\tau z$. In this sense a concrete $\tau \in Exp(Feat)$ is depictable in the way demonstrated in Fig. 1.

⁹ For a binary relation $r \subseteq A \times B$, A and B being two sets, and for any two sets A' and B' , $r \upharpoonright_{A' \times B'}$ is the restriction of r to $A' \times B'$, i.e., the set $\{<a, b> \in r \mid a \in A', b \in B'\}$. In case r is a function, we also write $r \upharpoonright_{A'}$ instead of $r \upharpoonright_{A' \times B'}$.

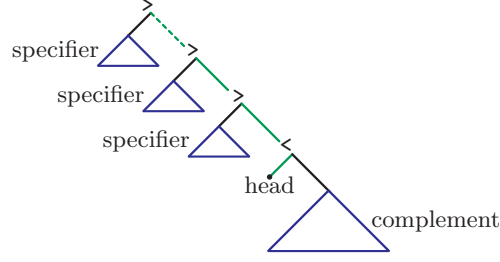


Fig. 2. The typical structure of a (minimalist) expression over *Feat*.

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

$spec_\tau \subseteq MaxProj(\tau) \times MaxProj(\tau)$ is the binary relation defined such that for all $v, \phi \in MaxProj(\tau)$ it holds that $v spec_\tau \phi$ iff $r_\phi = sibling_\tau(x)$ for some $x \in N_\tau$ with $r_v \triangleleft_\tau^+ x \triangleleft_\tau^+ head_\tau(r_v)$, where r_v and r_ϕ are the roots of v and ϕ , respectively. If $v spec_\tau \phi$ for some $v, \phi \in MaxProj(\tau)$ then ϕ is a *specifier of v* (in τ). $Spec(\tau)$ is the set $\{v \mid \tau spec_\tau v\}$.

An $v \in MaxProj(\tau)$ is said to *have*, or likewise, to *display (open) feature f* if the label assigned to v 's head by $label_\tau$ is non-empty and starts with an instance of $f \in Feat$.¹⁰

τ is *complete* if its head-label is in $\{c\}Phon^*Sem^*$, and if the label of each other leaf 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 *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 $\langle N_\tau, \triangleleft_\tau^*, \triangleleft_\tau, <_\tau \rangle$ via $label_\tau$, and replacing all instances of non-phonetic features with the empty string, afterwards.

An $v = \langle N_v, \triangleleft_v^*, \triangleleft_v, <_v, label_v \rangle \in Feat(Exp)$ is (*label preserving*) *isomorphic to τ* if there is a bijective function i from N_τ onto N_v with $x \triangleleft_\tau y$ iff $i(x) \triangleleft_v i(y)$, $x \triangleleft_\tau y$ iff $i(x) \triangleleft_v i(y)$, $x <_\tau y$ iff $i(x) <_v i(y)$ for $x, y \in N_\tau$, and with $label_\tau(x) = label_v(i(x))$ for each $x \in N_\tau$ being a leaf of τ . i is an *isomorphism (from τ to v)*.

Definition 2. For $\tau = \langle N_\tau, \triangleleft_\tau^*, \triangleleft_\tau, <_\tau, label_\tau \rangle \in Exp(Feat)$ with $N_\tau = tN_v$ for some $t \in \mathbb{N}^*$ and some tree domain N_v , and for $r \in \mathbb{N}^*$, $(\tau)_r$ denotes the *expression shifting τ to r* , i.e., the expression $\langle N_{\tau(r)}, \triangleleft_{\tau(r)}^*, \triangleleft_{\tau(r)}, <_{\tau(r)}, label_{\tau(r)} \rangle$ over

¹⁰ Thus the expression depicted in Fig. 1 has feature f , while its specifier and its complement have feature g and h , respectively.

$Feat$ with $N_{\tau(r)} = rN_v$ such that the function $i_{\tau(r)}$ from N_τ onto $N_{\tau(r)}$ with $i_{\tau(r)}(tx) = rx$ for all $x \in N_v$ is an isomorphism from τ to $(\tau)_r$.¹¹

For $v, \phi \in Exp(Feat)$ let $\chi = \langle N_\chi, \triangleleft_\chi^*, \prec_\chi, <_\chi, label_\chi \rangle$ be a complex expression over $Feat$ with root ϵ such that $(v)_0$ 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 $[\prec v, \phi]$ if $0 <_\chi 1$, and $[> v, \phi]$ if $1 <_\chi 0$.

Definition 3 ([26]). A *minimalist grammar (MG)* is a five-tuple of the form $G = \langle \neg Syn, Syn, Lex, \Omega, c \rangle$ with Ω being the operator set consisting of the structure building functions *merge* and *move* defined w.r.t. $Feat$ as in (me) and (mo) below, respectively, and with Lex being a *lexicon (over Feat)*, i.e., Lex is a finite set of simple expressions over $Feat$, and each lexical item $\tau \in Lex$ is of the form $\langle N_\tau, \triangleleft_\tau^*, \prec_\tau, <_\tau, label_\tau \rangle$ such that $N_\tau = \{\epsilon\}$, and such that $label_\tau(\epsilon)$ is in $(Select \cup Licensors)^* Base Licensees^* Phon^* Sem^*$.

(me) *merge* is a partial mapping from $Exp(Feat) \times Exp(Feat)$ into $Exp(Feat)$.

A pair $\langle v, \phi \rangle$ with $v, \phi \in Exp(Feat)$ belongs to $Dom(merge)$ if for some $\mathbf{x} \in Base$ and $\kappa, \lambda \in Feat^*$, conditions (i) and (ii) are fulfilled:¹²

- (i) the head-label of v is $\bar{\mathbf{x}}\kappa$ (i.e. v has selector $\bar{\mathbf{x}}$), and
- (ii) the head-label of ϕ is $\mathbf{x}\lambda$ (i.e. ϕ has category \mathbf{x}).

Then,

(me.1) $merge(v, \phi) = [\prec v', \phi']$ if v is simple, and

(me.2) $merge(v, \phi) = [> \phi', v']$ if v is complex,

where v' and ϕ' result from v and ϕ , respectively, just by deleting the instance of the feature that the respective head-label starts with (cf. Fig. 3).

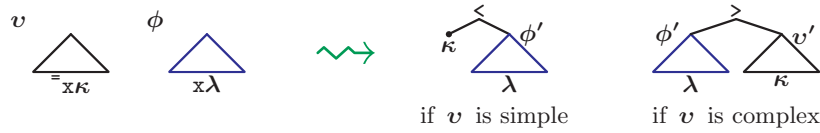


Fig. 3. $merge(v, \phi)$ according to (me).

(mo) *move* is a partial mapping from $Exp(Feat)$ to $Exp(Feat)$. An $v \in Exp(Feat)$ is in $Dom(move)$ if for some $\bar{\mathbf{x}} \in Licensees$ and $\kappa \in Feat^*$, (i)–(iii) are true:

¹¹ For any $t \in \mathbb{N}^*$ and $N \subseteq \mathbb{N}^*$, tN is just the concatenation product $\{t\}N$, i.e., the set $\{tx \mid x \in N\} \subseteq \mathbb{N}^*$ (cf. fn. 6). Note that for each $\tau = \langle N_\tau, \triangleleft_\tau^*, \prec_\tau, <_\tau, label_\tau \rangle$ from $Exp(Feat)$, a $t \in \mathbb{N}^*$ and tree domain N_v with $N_\tau = tN_v$ exist by (E4).

¹² For a partial function f from a set A into a set B , $Dom(f)$ is the domain of f , i.e., the set of all $x \in A$ for which $f(x)$ is defined.

- (i) the head-label of v is $+X\kappa$ (i.e. v has licenser $+X$),
- (ii) there is exactly one $\phi \in \text{MaxProj}(v)$ with head-label $-x\lambda$ for some $\lambda \in \text{Feat}^*$ (i.e. there is exactly one $\phi \in \text{MaxProj}(v)$ that has feature $-x$), and
- (iii) there exists a $\chi \in \text{Comp}^+(v)$ with $\phi = \chi$ or $\phi \in \text{Spec}(\chi)$.

Then,

$$\text{move}(v) = [>\phi', v'],$$

where $v' \in \text{Exp}(\text{Feat})$ results from v by canceling the instance of $+X$ the head-label of v starts with, while the subtree ϕ is replaced by a single node labeled ϵ . $\phi' \in \text{Exp}(\text{Feat})$ arises from ϕ by deleting the instance of $-x$ the head-label of ϕ starts with (cf. Fig. 4).



Fig. 4. $\text{move}(v)$ according to (mo).

Note that it is condition (ii) of (mo) which can be seen as providing a strict implementation of the *shortest movement condition (SMC)*: competing open licenses in one and the same given expression do not allow one to derive a complete expression from the given one. It is condition (iii) of (mo) which provides an implementation of the *specifier island condition (SPIC)*: a constituent has to belong to the transitive complement closure of a given tree or to be a specifier of such a constituent in order to be movable at all.

Since we are interested in the question of whether the generative capacity of our formalism is affected by giving up the SMC and sticking to the SPIC, we next present the definition of an MG without SMC.

Definition 4. A *minimalist grammar without SMC* (MG^{-SMC}) is a five-tuple of the form $\langle \neg\text{Syn}, \text{Syn}, \text{Lex}, \Omega, c \rangle$ where Ω is the operator set consisting of the structure building functions *merge* and move^{-SMC} defined w.r.t. Feat as in (me) above and (mo^{-SMC}) below, respectively, and where Lex is a lexicon over Feat defined as in Definition 3.

(mo^{-SMC}) move^{-SMC} is a partial mapping from $\text{Exp}(\text{Feat})$ to $\mathcal{P}_{\text{fin}}(\text{Exp}(\text{Feat}))$.¹³
 An $v \in \text{Exp}(\text{Feat})$ is in $\text{Dom}(\text{move})$ if for some $-x \in \text{Licensees}$ and $\kappa \in \text{Feat}^*$,
 (i)–(iii) are true:

¹³ $\mathcal{P}_{\text{fin}}(\text{Exp}(\text{Feat}))$ is the class of all finite subsets of $\text{Exp}(\text{Feat})$.

- (i) the head-label of v is $\mathbf{+X}\kappa$ (i.e. v has licenser $\mathbf{+X}$),
- (ii) there is some $\phi \in \text{MaxProj}(v)$ with head-label $\mathbf{-x}\lambda$ for some $\lambda \in \text{Feat}^*$ (i.e. there is some $\phi \in \text{MaxProj}(v)$ that has feature $\mathbf{-x}$), and
- (iii) there exists a $\chi \in \text{Comp}^+(v)$ with $\phi = \chi$ or $\phi \in \text{Spec}(\chi)$.

Then,

$$\text{move}^{-\text{SMC}}(v) = \left\{ \left[\triangleright \phi', v' \right] \left| \begin{array}{l} \phi \in \text{MaxProj}(\phi) \text{ with head-label } \mathbf{-x}\lambda \\ \text{for some } \lambda \in \text{Feat}^* \text{ such that there is} \\ \text{a } \chi \in \text{Comp}^+(v) \text{ for which } \phi = \chi \text{ or} \\ \phi \in \text{Spec}(\chi) \end{array} \right. \right\},$$

where $v' \in \text{Exp}(\text{Feat})$ results from v by canceling the instance of $\mathbf{+X}$ the head-label of v starts with, while the subtree ϕ is replaced by a single node labeled ϵ . $\phi' \in \text{Exp}(\text{Feat})$ arises from ϕ by deleting the instance of $\mathbf{-x}$ the head-label of ϕ starts with (cf. Fig. 4).

Let $G = \langle \neg\text{Syn}, \text{Syn}, \text{Lex}, \Omega, \mathbf{c} \rangle$ be an MG, respectively an $\text{MG}^{-\text{SMC}}$. Then the closure of G , $\text{CL}(G)$, is the set $\bigcup_{k \in \mathbb{N}} \text{CL}^k(G)$, where $\text{CL}^0(G) = \text{Lex}$, and for $k \in \mathbb{N}$, $\text{CL}^{k+1}(G) \subseteq \text{Exp}(\text{Feat})$ is recursively defined as the set

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

in case G is an MG, respectively as the set

$$\begin{aligned} & \text{CL}^k(G) \cup \{ \text{merge}(v, \phi) \mid \langle v, \phi \rangle \in \text{Dom}(\text{merge}) \cap \text{CL}^k(G) \times \text{CL}^k(G) \} \\ & \cup \bigcup_{v \in \text{Dom}(\text{move}^{-\text{SMC}}) \cap \text{CL}^k(G)} \text{move}^{-\text{SMC}}(v), \end{aligned}$$

in case G is an $\text{MG}^{-\text{SMC}}$. The set $\{Y_{\text{phon}}(\tau) \mid \tau \in \text{CL}(G) \text{ and } \tau \text{ complete}\}$, denoted by $L(G)$, is the (string) language derivable by G .

Definition 5. A set L is a *minimalist language (ML)* if $L = L(G)$ for some MG G , and it is a *minimalist language without SMC (ML^{-SMC})*, if $L = L(G)$ for some $\text{MG}^{-\text{SMC}}$ G .

Corollary 1. *Each ML has the constant growth property.*¹⁴ □

This corollary is an immediate consequence of the fact (cf. [16, 18]) that each language derivable by an MG in the sense of Definition 3 is a language derivable by a *linear context-free rewriting system (LCFRS)* in the sense of [28, 29].

¹⁴ For each set M and each $L \subseteq M^*$, L has the constant growth property, if there is an $N \in \mathbb{N}$ such that for all $w_1, w_2 \in L$ with $|w_1| < |w_2|$, and for which there is no $w_3 \in L$ with $|w_1| < |w_3| < |w_2|$, it holds that $|w_2| - |w_1| \leq N$. Here, for $w \in M^*$, $|w|$ denotes the length of w .

2.1 The notion of a relevant expression

The notion of what is a *relevant expression* within the closure of an MG, respectively $\text{MG}^{-\text{SMC}}$, G , is of some importance. We refer to an expression $\tau \in \text{CL}(G)$ as *relevant* if it serves to derive a complete expression. In particular, we want to emphasize that both in the case of an MG and in the case of an $\text{MG}^{-\text{SMC}}$, condition (iii) of the definition of the move-operator guarantees the following: whenever an expression $v \in \text{CL}(G)$ can be employed in order to generate a complete expression, there is no maximal projection $\psi \in \text{MaxProj}(v)$ such that ψ displays an unchecked licensee and is properly contained within some $\chi \in \text{Comp}^+(v)$.

To put it differently, if we applied the move-operator to some $v \in \text{CL}(G)$ such that some $\chi \in \text{Comp}^+(v)$ becomes a specifier of the resulting expression, it would be impossible to check off in a later derivation step any licensee feature displayed by some $\psi \in \text{MaxProj}(v)$ properly contained in χ , because applying the move-operator to v , ψ would end up in a position not matching condition (iii) of the definition of the move-operator, and this property is inherited by any expression subsequently derived. In the syntactic literature this is often referred to as a “freezing effect.” Complying with this effect, we can, in the case of an $\text{MG}^{-\text{SMC}}$, “hide” an unbounded, finite number of different instances of the same licensee “along” the transitive complement closure of a single expression; and we exploit exactly this possibility in the next section, when we define an $\text{MG}^{-\text{SMC}}$ deriving a language which is not mildly context-sensitive.

3 A non-mildly context-sensitive $\text{ML}^{-\text{SMC}}$

We are now going to present an $\text{MG}^{-\text{SMC}}$ deriving a language which does not fulfil the constant growth property, namely, the language $\{a^{2^n} \mid n \in \mathbb{N}\}$.

Example 1. Assume $G_{\text{ex}} = \langle \neg\text{Syn}, \text{Syn}, \text{Lex}, \Omega, \mathbf{c} \rangle$ to be the $\text{MG}^{-\text{SMC}}$ for which

$$\begin{array}{lll} \text{Sem} = \emptyset & \text{Base} = \{\mathbf{c}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}\} & \text{Licensees} = \{-\mathbf{l}, -\mathbf{m}\} \\ \text{Phon} = \{\mathbf{a}\} & \text{Select} = \{^{\bar{}}\mathbf{c}, ^{\bar{}}\mathbf{w}, ^{\bar{}}\mathbf{x}, ^{\bar{}}\mathbf{y}, ^{\bar{}}\mathbf{z}\} & \text{Licensors} = \{+\mathbf{L}, +\mathbf{M}\} \end{array}$$

and for which *Lex* consists of the following 9 simple expressions:¹⁵

$$\begin{array}{llll} & & \beta_1 = \mathbf{w-m} & \beta_2 = ^{\bar{}}\mathbf{wx-l} \\ \gamma_1 = ^{\bar{}}\mathbf{x+My-m} & \gamma_2 = ^{\bar{}}\mathbf{y+Lz-l} & \gamma_3 = ^{\bar{}}\mathbf{zy-l} & \gamma_4 = ^{\bar{}}\mathbf{zx-l} \\ & & \delta_1 = ^{\bar{}}\mathbf{x+Mc} & \delta_2 = ^{\bar{}}\mathbf{c+Lca} \end{array}$$

Instead of a strictly formal proof that $L(G_{\text{ex}}) = \{a^{2^n} \mid n \in \mathbb{N}\}$ we will give the crucial details in a descriptive manner.

¹⁵ Since all lexical entries of G_{ex} are heads, we simply represent each of them by its (unique) label.

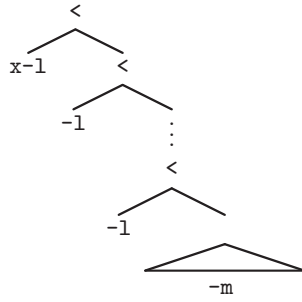


Fig. 9. Stopping the derivation cycle by merging with γ_4 .

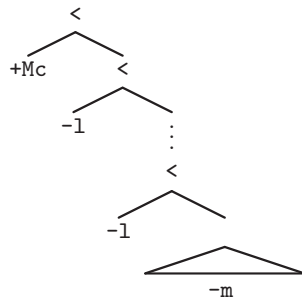


Fig. 10. Leaving the derivation cycle by merging with δ_1 .

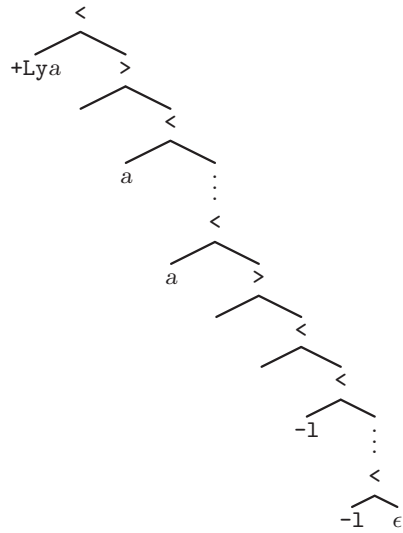


Fig. 11. Finishing the derivation by successively merging with δ_2 and checking off the remaining instances of licensee -1, thereby introducing an instance of an a for each -1.

Then, after the “cycle end-marking” instance of licensee $-m$ has been checked through an application of $move^{-SMC}$, all instances of licensee -1 get successively checked by first merging with δ_2 and applying $move^{-SMC}$ afterwards. Hence, in particular, for each instance of licensee -1 exactly one instance of (phonetic) feature a is introduced via merging with δ_2 (cf. Fig. 11). Therefore, $\{a^{2^n} \mid n \in \mathbb{N}\}$ is in fact the language derived by G_{ex} .

4 Conclusion and Outlook

We have shown that removing the SMC from MGs increases their generative capacity beyond mild context-sensitivity. In particular, we have provided an exemplary MG^{-SMC} deriving the language $\{a^{2^n} \mid n \in \mathbb{N}\}$, i.e. a language lacking the constant growth property. Importantly, this effect arises in spite of the fact that MG^{-SMC} s are constrained by a CLC, namely, the SPIC. This suggests that, intuitions to the contrary notwithstanding, the imposition of LCs on grammars as such, in the case at hand the SPIC, does not automatically reduce their generative capacity.

Note also that closely in keeping with some further suggestions in [12], a certain type of a *strict minimalist grammar (SMG)* has been introduced in [26] as well. This MG-type allows only movement of constituents belonging to the transitive complement closure of a tree. But in contrast to the MG-type matching our Definition 3, the triggering licensee feature may head the head-label of any constituent within the reflexive-transitive specifier closure 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 during the course of a derivation. Beside these differences, SMGs have implemented the SMC within the definition of the move-operator in the same way MGs have. SMGs and MGs have been shown to be weakly equivalent in [15, 18] confirming a conjecture explicitly stated in [26]. Note that, if we defined *strict minimalist grammars without SMC* (SMG^{-SMC}) by relaxing the condition (ii) of the move-operator as we did for MGs in Definition 4, our example MG^{-SMC} would also match the criteria such an SMG^{-SMC} had to fulfil. This, of course, is of interest, since it suggests that providing the implementation of the SPIC with a “final strictness,” does not prevent us from being able to derive non-mildly context-sensitive languages when giving up the SMC.

There has recently been made another interesting attempt, namely, by Kobele [11], to look at the consequences concerning the generative capacity of MGs when changing the original definition from [25] and allowing a certain kind of feature percolation. In fact, Kobele proved that, if the syntactic features of a head are presented as strings which are checked “from left to right,” and if movement to a specifier position generally allows for the possibility that the syntactic features of the specifier’s head are inherited by the attracting head in such a way that they become an integral part of the syntactic features of the attracting head, then MGs modified in this respect allow one to derive any language of type

0.^{17,18} We conjecture that this is true of $\text{MG}^{-\text{SMC}}$ s as well. The reason for this is that $\text{MG}^{-\text{SMC}}$ s (i.e. MGs for which the SPIC, the specifier island condition, SPIC, but not the SMC, the shortest move condition, hold) on the one hand and MGs of the Kobele-type (i.e. MGs generally allowing for feature percolation from specifiers to heads) on the other appear to constitute complementary pictures of one and the same thing. The sort of feature percolation Kobele considers allows to “collect” instantiations of the same feature type within a single head-label without violating the SMC, since only the first feature instantiation within the head-label is visible to the move-operator. The implementation of the SPIC in MGs discussed in our paper and the simultaneous dropping of the SMC allows to “collect” instantiations of the same feature type within the transitive complement closure, since only the lowest, i.e. most deeply embedded, instantiation can be checked off without leading to a crashing derivation. This connection certainly deserves more attention and has to be elaborated quite carefully.¹⁹

To end on a more linguistic note, we observe that constraint interaction among LCs (ILCs/CLCs) and its impact on the generative capacity of grammars is still deplorably understudied and consequently not very well understood. Our own attempts here have obviously been rather sketchy. Further research will have to provide an exact characterization of the LC/complexity connection. This would involve a clearer picture of the tight relation between the SMC and LCFRSs, which has guided much of the research on the complexity of MGs.²⁰

¹⁷ This is at least true, when—by means of an “MG-external” encoding—we treat type 0-languages as recursively enumerable subsets of the natural numbers, because what Kobele concretely proves, is that each arbitrary *abacus* in the sense of [13] can be simulated by a corresponding MG. How to define such an MG directly deriving a given type 0-language, seems to be an open problem.

¹⁸ Note also that “permitting percolation of unchecked features of the attracted head into the attracting one,” “representing head-features as strings” and, depending on this representation, “demanding a left-to-right-checking of features” should be seen as properties of a particular instantiation of a slightly more general case still implying the same result on generative capacity (cf. [11]). Here we concentrate on this particular instantiation, just with the intend of keeping our exposition somewhat simpler and more accessible.

¹⁹ To support our conjecture, it should also be mentioned here that Kobele [10] pointed out, how in a different framework, namely, *mirror theoretic grammars (MTGs)* developed in [10] as a formalization of the syntactic theory proposed in [2], it is possible to define an (unrestricted) MTG deriving the language $\{a^{2^n} \mid n \in \mathbb{N}\}$. In fact, MTGs in their unrestricted version can be seen as strongly related to the MG-type Kobele considers in [11] exactly in the way they allow for feature percolation—though, in place of percolation from specifiers to heads, we are concerned with percolation from complements to (selecting) heads in the MTG-case—and the corresponding kind of percolation is employed to derive $\{a^{2^n} \mid n \in \mathbb{N}\}$ by an MTG.

²⁰ Note that a *prima facie* problematic domain of grammar, namely, multiple-wh-fronting constructions (cf. [23]) can be harmonized with the SMC if, among other things, one assumes wh-cluster formation triggered by special clustering features.

References

- [1] Pierre Boullier. Proposal for a natural language processing syntactic backbone. Report No. 3342, INRIA research reports, INRIA Rocquencourt, 1998. Available at <http://www.inria.fr/rrrt/rr-3342.html>.
- [2] Michael Brody. Mirror theory. Syntactic representation in perfect syntax. *Linguistic Inquiry*, 31:29–65, 2000.
- [3] Noam Chomsky. On wh-movement. In P. Culicover, T. Wasow, and A. Akmajian, editors, *Formal Syntax*, pages 71–132. Academic Press, New York, NY, 1977.
- [4] Noam Chomsky. *Barriers*. MIT Press, Cambridge, MA, 1986.
- [5] Noam Chomsky. *The Minimalist Program*. MIT Press, Cambridge, MA, 1995.
- [6] Noam Chomsky. Beyond explanatory adequacy. MIT Occasional Papers in Linguistics (MITOPL # 20), Massachusetts Institute of Technology, Department of Linguistics and Philosophy, Cambridge, MA, 2001.
- [7] Philippe de Groote, Glyn Morrill, and Christian Retoré, editors. *Logical Aspects of Computational Linguistics (LACL '01)*, LNAI Vol. 2099. Springer, Berlin, Heidelberg, 2001.
- [8] Henk Harkema. A characterization of minimalist languages. In de Groote et al. [7], pages 193–211.
- [9] Gerhard Jäger. On the generative capacity of multi-modal categorial grammars. *Research on Language and Computation*, 1:105–125, 2003.
- [10] Gregory M. Kobele. Formalizing mirror theory. *Grammars*, 5:177–221, 2003.
- [11] Gregory M. Kobele. Features moving madly. *Research on Language and Computation*, to appear. Draft version available at <http://www.linguistics.ucla.edu/people/grads/kobele/papers.htm>.
- [12] Hilda Koopman and Anna Szabolcsi. *Verbal Complexes*. MIT Press, Cambridge, MA, 2000.
- [13] Joachim Lambek. How to program an (infinite) abacus. *Canadian Mathematical Bulletin*, 4:295–302, 1961.
- [14] Jens Michaelis. Derivational minimalism is mildly context-sensitive. In M. Moortgat, editor, *Logical Aspects of Computational Linguistics (LACL '98)*, LNAI Vol. 2014, pages 179–198. Springer, Berlin, Heidelberg, 2001.
- [15] Jens Michaelis. Observations on strict derivational minimalism. In *FGMOL '01. Preproceedings*. Joint conference of the 6th conference on Formal Grammar and the 7th meeting of the Association for Mathematics of Language, Helsinki, 2001.
- [16] Jens Michaelis. *On Formal Properties of Minimalist Grammars*. PhD thesis, Potsdam University, Potsdam, 2001.
- [17] Jens Michaelis. Transforming linear context-free rewriting systems into minimalist grammars. In de Groote et al. [7], pages 228–244.
- [18] Jens Michaelis. Implications of a revised perspective on minimalist grammars. Draft, Potsdam University, 2002. Available at <http://www.ling.uni-potsdam.de/~michael/papers.html>.
- [19] Michael Moortgat. Multimodal linguistic inference. *Journal of Logic, Language and Information*, 5:349–385, 1996.
- [20] Glyn Morrill. *Type Logical Grammar*. Kluwer, Dordrecht, 1994.
- [21] Owen Rambow and Giorgio Satta. Independent parallelism in finite copying parallel rewriting systems. *Theoretical Computer Science*, 223:87–120, 1999.
- [22] John R. Ross. *Constraints on Variables in Syntax*. PhD thesis, MIT, Cambridge, MA, 1967.

- [23] Catherine Rudin. On multiple questions and multiple wh-fronting. *Natural Language and Linguistic Theory*, 6:445–501, 1988.
- [24] Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context-free grammars. *Theoretical Computer Science*, 88:191–229, 1991.
- [25] Edward P. Stabler. Derivational minimalism. In C. Retoré, editor, *Logical Aspects of Computational Linguistics (LACL '96)*, LNAI Vol. 1328, pages 68–95. Springer, Berlin, Heidelberg, 1997.
- [26] 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.
- [27] Mark Steedman. *Surface Structure and Interpretation*. MIT Press, Cambridge, MA, 1996.
- [28] 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.
- [29] David J. Weir. *Characterizing Mildly Context-Sensitive Grammar Formalisms*. PhD thesis, University of Pennsylvania, Philadelphia, PA, 1988.