On the Computational Complexity of Dominance Links in Grammatical Formalisms*

Sylvain Schmitz

LSV, ENS Cachan & CNRS, France
sylvain.schmitz@lsv.ens-cachan.fr

Abstract

Dominance links were introduced in grammars to model long distance scrambling phenomena, motivating the definition of *multiset-valued linear indexed grammars* (MLIGs) by Rambow (1994b), and inspiring quite a few recent formalisms. It turns out that MLIGs have since been rediscovered and reused in a variety of contexts, and that the complexity of their emptiness problem has become the key to several open questions in computer science. We survey complexity results and open issues on MLIGs and related formalisms, and provide new complexity bounds for some linguistically motivated restrictions.

1 Introduction

Scrambling constructions, as found in German and other SOV languages (Becker et al., 1991; Rambow, 1994a; Lichte, 2007), cause notorious difficulties to linguistic modeling in classical grammar formalisms like HPSG or TAG. A well-known illustration of this situation is given in the following two German sentences for "that Peter has repaired the fridge today" (Lichte, 2007),

> dass [Peter]　　 heute [den Kühlschrank] repariert hat
> that Peter$_{nom}$ today the fridge$_{acc}$　　　 repaired has
>
> dass [den Kühlschrank] heute [Peter]　　 repariert hat
> that the fridge$_{acc}$　　　 today Peter$_{nom}$ repaired has

with a flexible word order between the two complements of *repariert*, namely between the nominative *Peter* and the accusative *den Kühlschrank*.

Rambow (1994b) introduced a formalism, *unordered vector grammars with dominance links* (UVG-dls), for modeling such phenomena. These grammars are defined by vectors of context-free productions along with dominance links that should be enforced during derivations; for instance, Figure 1 shows how a flexible order between the complements of *repariert* could be expressed in an UVG-dl. Similar dominance mechanisms have been employed in various tree description formalisms (Rambow et al., 1995, 2001; Candito and Kahane, 1998;

1

$$\left\{\begin{array}{ccc} \text{VP} & \text{VP} & \overset{\ast}{\rightrightarrows}\ \text{VP} \\ /\ \backslash & /\ \backslash & | \\ \text{NP}_{nom}\ \ \text{VP} & \text{NP}_{acc}\ \ \text{VP} & \text{V} \\ & & | \\ & & \text{repariert} \end{array}\right\}$$
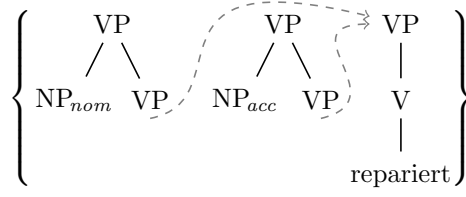
Figure 1: A vector of productions for the verb *repariert* together with its two complements.

Kallmeyer, 2001; Guillaume and Perrier, 2010) and TAG extensions (Becker et al., 1991; Rambow, 1994a).

However, the prime motivation for this survey is another grammatical formalism defined in the same article: *multiset-valued linear indexed grammars* (Rambow, 1994b, MLIGs), which can be seen as a low-level variant of UVG-dls that uses multisets to emulate unfulfilled dominance links in partial derivations. It is a natural extension of Petri nets, with broader scope than just UVG-dls; indeed, it has been independently rediscovered by de Groote et al. (2004) in the context of linear logic, and by Verma and Goubault-Larrecq (2005) in that of equational theories. Moreover, the decidability of its emptiness problem has proved to be quite challenging and is still uncertain, with several open questions depending on its resolution:

- provability in multiplicative exponential linear logic (de Groote et al., 2004),
- emptiness and membership of abstract categorial grammars (de Groote et al., 2004; Yoshinaka and Kanazawa, 2005),
- emptiness and membership of Stabler (1997)'s minimal grammars without shortest move constraint (Salvati, 2010),
- satisfiability of first-order logic on data trees (Bojańczyk et al., 2009), and of course
- emptiness and membership for the various formalisms that embed UVG-dls.

Unsurprisingly in the light of their importance in different fields, several authors have started investigating the complexity of decisions problems for MLIGs (Demri et al., 2009; Lazić, 2010). We survey the current state of affairs, with a particular emphasis on two points:

1. the applicability of complexity results to UVG-dls, which is needed if we are to conclude anything on related formalisms with dominance links,

2. the effects of two linguistically motivated restrictions on such formalisms, lexicalization and boundedness/rankedness.

The latter notion is imported from Petri nets, and turns out to offer interesting new complexity trade-offs, as we prove that k-boundedness and k-rankedness are ExpTime-complete for MLIGs, and that the emptiness and membership problems are ExpTime-complete for k-bounded MLIGs but PTime-complete in the k-ranked case. This also implies an ExpTime lower bound for emptiness and membership in minimal grammars with shortest move constraint.

We first define MLIGs formally in Section 2 and review related formalisms in Section 3. We proceed with complexity results in Section 4 before concluding in Section 5.

Notations   In the following, Σ denotes a finite alphabet, Σ^* the set of finite sentences over Σ, and ε the empty string. The length of a string w is noted $|w|$, and the number of occurrence of a symbol a in w is noted $|w|_a$. A language is formalized as a subset of Σ^*. Let $\mathbb{N}^n$ denote the set of vectors of positive integers of dimension n. The i-th component of a vector $\bar{x}$ in $\mathbb{N}^n$ is $\bar{x}(i)$, $\bar{0}$ denotes the null vector, $\bar{1}$ the vector with 1 values, and $\bar{e}_i$ the vector with 1 as its i-th component and 0 everywhere else. The ordering $\leq$ on $\mathbb{N}^n$ is the componentwise ordering: $\bar{x} \leq \bar{y}$ iff $\bar{x}(i) \leq \bar{y}(i)$ for all $0 < i \leq n$. The size of a vector refers to the size of its binary encoding: $|\bar{x}| = \sum_{i=1}^{n} 1 + \max(0, \lfloor \log_2 \bar{x}(i) \rfloor)$.

We refer the reader unfamiliar with complexity classes and notions such as hardness or LOGSPACE reductions to classical textbooks (e.g. Papadimitriou, 1994).

2   Multiset-Valued Linear Indexed Grammars

Definition 1 (Rambow, 1994b)**.** An n-dimensional *multiset-valued linear indexed grammar* (MLIG) is a tuple $\mathcal{G} = \langle N, \Sigma, P, (S, \bar{x}_0) \rangle$ where N is a finite set of nonterminal symbols, Σ a finite alphabet disjoint from N, $V = (N \times \mathbb{N}^n) \uplus \Sigma$ the vocabulary, P a finite set of productions in $(N \times \mathbb{N}^n) \times V^*$, and $(S, \bar{x}_0) \in N \times \mathbb{N}^n$ the start symbol. Productions are more easily written as

$$(A, \bar{x}) \to u_0 (B_1, \bar{x}_1) u_1 \cdots u_m (B_m, \bar{x}_m) u_{m+1} \qquad (\star)$$

with each u_i in Σ^* and each $(B_i, \bar{x}_i)$ in $N \times \mathbb{N}^n$.

The *derivation* relation $\Rightarrow$ over sequences in V^* is defined by

$$\delta (A, \bar{y}) \delta' \Rightarrow \delta u_0 (B_1, \bar{y}_1) u_1 \cdots u_m (B_m, \bar{y}_m) u_{m+1} \delta'$$

if δ and δ' are in V^*, a production of form $(\star)$ appears in P, $\bar{x} \leq \bar{y}$, for each $1 \leq i \leq m$, $\bar{x}_i \leq \bar{y}_i$, and $\bar{y} - \bar{x} = \sum_{i=1}^{m} \bar{y}_i - \bar{x}_i$.

The *language* of a MLIG is the set of terminal strings derived from $(S, \bar{x}_0)$, i.e.

$$L(\mathcal{G}) = \{ w \in \Sigma^* \mid (S, \bar{x}_0) \Rightarrow^* w \}$$

and we denote by $\mathcal{L}(\text{MLIG})$ the class of MLIG languages.

Example 2. To illustrate this definition, and its relevance for free word order languages, consider the 3-dimensional MLIG with productions

$$(S, \bar{0}) \to \varepsilon \mid (S, \bar{1}), \qquad\qquad (S, \bar{e}_1) \to a \, (S, \bar{0}),$$
$$(S, \bar{e}_2) \to b \, (S, \bar{0}), \qquad\qquad (S, \bar{e}_3) \to c \, (S, \bar{0})$$

and start symbol $(S, \bar{0})$. It generates the MIX language of all sentences with the same number of a, b, and c's (see Figure 2 for an example derivation):

$$L_{\text{mix}} = \{ w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c \} \, .$$

3

$$S, (0,0,0)$$
$$|$$
$$S, (1,1,1)$$
$$\diagup \quad \diagdown$$
$$b \quad S, (1,0,1)$$
$$|$$
$$S, (2,1,2)$$
$$\diagup \quad \diagdown$$
$$c \quad S, (2,1,1)$$
$$\diagup \quad \diagdown$$
$$a \quad S, (1,1,1)$$
$$\diagup \quad \diagdown$$
$$a \quad S, (0,1,1)$$
$$\diagup \quad \diagdown$$
$$b \quad S, (0,0,1)$$
$$\diagup \quad \diagdown$$
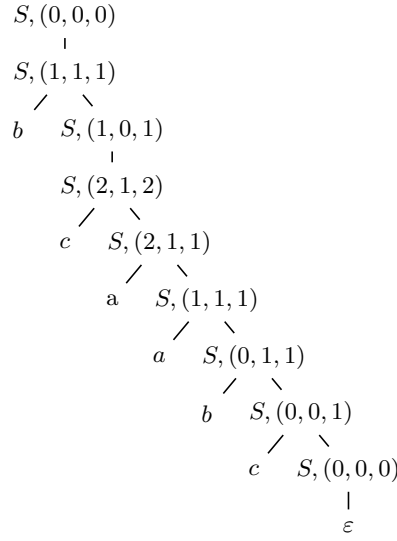$$c \quad S, (0,0,0)$$
$$|$$
$$\varepsilon$$

Figure 2: A derivation for *bcaabc* in the grammar of Example 2.

The *size* $|\mathcal{G}|$ of a MLIG $\mathcal{G}$ is essentially the sum of the sizes of each of its productions of form $(\star)$:

$$|\mathcal{G}| = |\overline{x}_0| + \sum_P \left(m + 1 + |\overline{x}| + \sum_{i=1}^{m} |\overline{x}_i| + \sum_{i=0}^{m+1} |u_i| \right).$$

2.1 Normal Forms

A MLIG is in *extended two form* (ETF) if all its productions are of form

terminal $(A, \overline{0}) \to a$ or $(A, \overline{0}) \to \varepsilon$, or

nonterminal $(A, \overline{x}) \to (B_1, \overline{x}_1)(B_2, \overline{x}_2)$ or $(A, \overline{x}) \to (B_1, \overline{x}_1)$,

with a in Σ, A, B_1, B_2 in N, and $\overline{x}$, $\overline{x}_1$, $\overline{x}_2$ in $\mathbb{N}^n$. Using standard constructions, any MLIG can be put into ETF in linear time or logarithmic space.

A MLIG is in *restricted index normal form* (RINF) if the productions in P are of form $(A, \overline{0}) \to \alpha$, $(A, \overline{0}) \to (B, \overline{e}_i)$, or $(A, \overline{e}_i) \to (B, \overline{0})$, with A, B in N, $0 < i \le n$, and α in $(\Sigma \cup (N \times \{\overline{0}\}))^*$. The direct translation into RINF proposed by Rambow (1994a) is exponential if we consider a binary encoding of vectors, but using techniques developed for Petri nets (Dufourd and Finkel, 1999), this blowup can be avoided:

Proposition 3. *For any MLIG, one can construct an equivalent MLIG in RINF in logarithmic space.*

2.2 Restrictions

Two restrictions on dominance links have been suggested in an attempt to reduce their complexity, sometimes in conjunction: lexicalization and *k*-boundedness. We provide here characterizations for them in terms of MLIGs. We can combine the two restrictions, thus defining the class of *k*-bounded lexicalized MLIGs.

4

Lexicalization   Lexicalization in UVG-dls reflects the strong dependence between syntactic constructions (vectors of productions representing an extended domain of locality) and lexical anchors. We define here a restriction of MLIGs with similar complexity properties:

Definition 4. A terminal derivation $\alpha \Rightarrow^p w$ with w in Σ^* is *c-lexicalized* for some $c > 0$ if $p \leq c \cdot |w|$.[1]   A MLIG is *lexicalized* if there exists c such that any terminal derivation starting from $(S, \overline{x}_0)$ is c-lexicalized, and we denote by $\mathcal{L}(\text{MLIG}_\ell)$ the set of lexicalized MLIG languages.

Looking at the grammar of Example 2, any terminal derivation $(S, \overline{0}) \Rightarrow^p w$ verifies $p = \frac{4 \cdot |w|}{3} + 1$, and the grammar is thus lexicalized.

Boundedness   As dominance links model long-distance dependencies, bounding the number of simultaneously pending links can be motivated on competence/performance grounds (Joshi et al., 2000; Kallmeyer and Parmentier, 2008), and on complexity/expressiveness grounds (Søgaard et al., 2007; Kallmeyer and Parmentier, 2008; Chiang and Scheffler, 2008). The *shortest move constraint* (SMC) introduced by Stabler to enforce a strong form of minimality also falls into this category of restrictions.

Definition 5. A MLIG derivation $\alpha_0 \Rightarrow \alpha_1 \Rightarrow \cdots \Rightarrow \alpha_p$ is of *rank k* for some $k \geq 0$ if, no vector with a sum of components larger than k can appear in any α_j, i.e. for all $\overline{x}$ in $\mathbb{N}^n$ such that there exist $0 \leq j \leq p$, δ, δ' in V^* and A in N with $\alpha_j = \delta(A, \overline{x})\delta'$, one has $\sum_{i=1}^{n} \overline{x}(i) \leq k$.
A MLIG is *k-ranked* (noted kr-MLIG) if any derivation starting with $\alpha_0 = (S, \overline{x}_0)$ is of rank k. It is *ranked* if there exists k such that it is k-ranked.

A 0-ranked MLIG is simply a context-free grammar (CFG), and we have more generally the following:

Lemma 6. *Any n-dimensional k-ranked MLIG $\mathcal{G}$ can be transformed into an equivalent CFG $\mathcal{G}'$ in time $O(|\mathcal{G}| \cdot (n+1)^{k^3})$.*

Proof. We assume $\mathcal{G}$ to be in ETF, at the expense of a linear time factor. Each A in N is then mapped to at most $(n+1)^k$ nonterminals $(A, \overline{y})$ in $N' = N \times \mathbb{N}^n$ with $\sum_{i=1}^{n} \overline{y}(i) \leq k$. Finally, for each production $(A, \overline{x}) \rightarrow (B_1, \overline{x}_1)(B_2, \overline{x}_2)$ of P, at most $(n+1)^{k^3}$ choices are possible for productions $(A, \overline{y}) \rightarrow (B_1, \overline{y}_1)(B_2, \overline{y}_2)$ with $(A, \overline{y})$, $(B_1, \overline{y}_1)$, and $(B_2, \overline{y}_2)$ in N'. $\square$

A definition quite similar to k-rankedness can be found in the Petri net literature:

Definition 7. A MLIG derivation $\alpha_0 \Rightarrow \alpha_1 \Rightarrow \cdots \Rightarrow \alpha_p$ is *k-bounded* for some $k \geq 0$ if, no vector with a coordinate larger than k can appear in any α_j, i.e. for all $\overline{x}$ in $\mathbb{N}^n$ such that there exist $0 \leq j \leq p$, δ, δ' in V^* and A in N with $\alpha_j = \delta(A, \overline{x})\delta'$, and for all $1 \leq i \leq n$, one has $\overline{x}(i) \leq k$.
A MLIG is *k-bounded* (noted kb-MLIG) if any derivation starting with $\alpha_0 = (S, \overline{x}_0)$ is k-bounded. It is *bounded* if there exists k such that it is k-bounded.

---

[1] This restriction is slightly stronger than that of *linearly restricted* derivations (Rambow, 1994b), but still allows to capture UVG-dl lexicalization.

The SMC in minimal grammars translates exactly into 1-boundedness of the corresponding MLIGs (Salvati, 2010).

Clearly, any k-ranked MLIG is also k-bounded, and conversely any n-dimensional k-bounded MLIG is (kn)-ranked, thus a MLIG is ranked iff it is bounded. The counterpart to Lemma 6 is:

Lemma 8. *Any n-dimensional k-bounded MLIG $\mathcal{G}$ can be transformed into an equivalent CFG $\mathcal{G}'$ in time $O(|\mathcal{G}| \cdot (k+1)^{n^2})$.*

Proof. We assume $\mathcal{G}$ to be in ETF, at the expense of a linear time factor. Each A in N is then mapped to at most $(k+1)^n$ nonterminals $(A, \overline{y})$ in $N' = N \times \{0, \ldots, k\}^n$. Finally, for each production $(A, \overline{x}) \to (B_1, \overline{x}_1)(B_2, \overline{x}_2)$ of P, each nonterminal $(A, \overline{y})$ of N' with $\overline{x} \leq \overline{y}$, and each index $0 < i \leq n$, there are at most $k+1$ ways to split $(\overline{y}(i) - \overline{x}(i)) \leq k$ into $\overline{y}_1(i) + \overline{y}_2(i)$ and span a production $(A, \overline{y}) \to (B_1, \overline{x}_1 + \overline{y}_1)(B_2, \overline{x}_2 + \overline{y}_2)$ of P'. Overall, each production is mapped to at most $(k+1)^{n^2}$ context-free productions. $\square$

One can check that the grammar of Example 2 is not bounded (to see this, repeatedly apply production $(S, \overline{0}) \to (S, \overline{1})$), as expected since MIX is not a context-free language.

2.3 Language Properties

Let us mention a few more results pertaining to MLIG languages:

Proposition 9 (Rambow, 1994b). *$\mathcal{L}(\mathrm{MLIG})$ is a substitution closed full abstract family of languages.*

Proposition 10 (Rambow, 1994b). *$\mathcal{L}(\mathrm{MLIG}_\ell)$ is a subset of the context-sensitive languages.*

Natural languages are known for displaying some limited cross-serial dependencies, as witnessed in linguistic analyses, e.g. of Swiss-German (Shieber, 1985), Dutch (Kroch and Santorini, 1991), or Tagalog (Maclachlan and Rambow, 2002). This includes the copy language

$$L_{\mathrm{copy}} = \{ww \mid w \in \{a, b\}^*\} \,,$$

which does not seem to be generated by any MLIG:

Conjecture 11 (Rambow, 1994b). *L_{copy} is not in $\mathcal{L}(\mathrm{MLIG})$.*

Finally, we obtain the following result as a consequence of Lemmas 6 and 8:

Corollary 12. *$\mathcal{L}(k\mathrm{r}\text{-}\mathrm{MLIG}) = \mathcal{L}(k\mathrm{b}\text{-}\mathrm{MLIG}) = \mathcal{L}(k\mathrm{b}\text{-}\mathrm{MLIG}_\ell)$ is the set of context-free languages.*

3 Related Formalisms

We review formalisms connected to MLIGs, starting in Section 3.1 with Petri nets and two of their extensions, which turn out to be exactly equivalent to MLIGs. We then consider various linguistic formalisms that employ dominance links (Section 3.2).
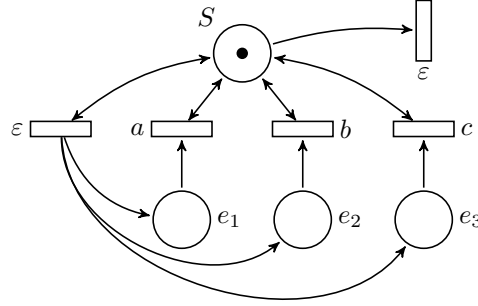
Figure 3: The labeled Petri net corresponding to the right linear MLIG of Example 2.

3.1 Petri Nets

Definition 13 (Petri, 1962). A marked *Petri net*[2] is a tuple $\mathcal{N} = \langle \mathcal{S}, T, f, \overline{m}_0 \rangle$ where $\mathcal{S}$ and T are disjoint finite sets of places and transitions, f a flow function from $(\mathcal{S} \times T) \cup (T \times \mathcal{S})$ to $\mathbb{N}$, and $\overline{m}_0$ an initial marking in $\mathbb{N}^{\mathcal{S}}$. A transition $t \in T$ can be fired in a *marking* $\overline{m}$ in $\mathbb{N}^{\mathcal{S}}$ if $f(p, t) \geq \overline{m}(p)$ for all $p \in \mathcal{S}$, and reaches a new marking $\overline{m}'$ defined by $\overline{m}'(p) = \overline{m}(p) - f(p, t) + f(t, p)$ for all $p \in \mathcal{S}$, written $\overline{m}\, [t\rangle\, \overline{m}'$. Another view is that place p holds $\overline{m}(p)$ *tokens*, $f(p, t)$ of which are first removed when firing t, and then $f(t, p)$ added back. Firings are extended to sequences σ in T^* by $\overline{m}\, [\varepsilon\rangle\, \overline{m}$, and $\overline{m}\, [\sigma t\rangle\, \overline{m}'$ if there exists $\overline{m}''$ with $\overline{m}\, [\sigma\rangle\, \overline{m}''\, [t\rangle\, \overline{m}'$.

A *labeled* Petri net with reachability acceptance is endowed with a labeling homomorphism $\varphi : T^* \to \Sigma^*$ and a finite acceptance set $F \subseteq \mathbb{N}^{\mathcal{S}}$, defining the language (Peterson, 1981)

$$L(\mathcal{N}, \varphi, F) = \{ \varphi(\sigma) \in \Sigma^* \mid \exists \overline{m} \in F, \overline{m}_0\, [\sigma\rangle\, \overline{m} \} \ .$$

Labeled Petri nets (with acceptance set $\{\overline{0}\}$) are notational variants of *right linear* MLIGs, defined as having production in $(N \times \mathbb{N}^n) \times (\Sigma^* \cup (\Sigma^* \cdot (N \times \mathbb{N}^n)))$. This is is case of the MLIG of Example 2, which is given in Petri net form in Figure 3, where circles depict places (representing MLIG nonterminals and indices) with black dots for initial tokens (representing the MLIG start symbol), boxes transitions (representing MLIG productions), and arcs the flow values. For instance, production $(S, \overline{e}_3) \to c\,(S, \overline{0})$ is represented by the rightmost, c-labeled transition, with $f(S, t) = f(e_3, t) = f(t, S) = 1$ and $f(e_1, t) = f(e_2, t) = f(t, e_1) = f(t, e_2) = f(t, e_3) = 0$.

Extensions The subsumption of Petri nets is not innocuous, as it allows to derive lower bounds on the computational complexity of MLIGs. Among several extensions of Petri net with some branching capacity (see e.g. Mayr, 1999; Haddad and Poitrenaud, 2007), two are of singular importance: It turns out that MLIGs in their full generality have since been independently rediscovered under the names *vector addition tree automata* (de Groote et al., 2004, VATA) and *branching VASS* (Verma and Goubault-Larrecq, 2005, BVASS).

---

[2]Petri nets are also equivalent to *vector addition system* (Karp and Miller, 1969, VAS) and *vector addition systems with states* (Hopcroft and Pansiot, 1979, VASS).

Semilinearity Another interesting consequence of the subsumption of Petri nets by MLIGs is that the former generate some non semilinear languages, i.e. with a Parikh image which is not a semilinear subset of $\mathbb{N}^{|\Sigma|}$ (Parikh, 1966). Hopcroft and Pansiot (1979, Lemma 2.8) exhibit an example of a VASS with a non semilinear reachability set, which we translate as a 2-dimensional right linear MLIG with productions[3]

$$(S, \overline{e}_2) \to (S, \overline{e}_1), \qquad\qquad (S, \overline{0}) \to (A, \overline{0}) \mid (B, \overline{0}),$$
$$(A, \overline{e}_1) \to (A, 2\overline{e}_2), \qquad\qquad (A, \overline{0}) \to a\,(S, \overline{0}),$$
$$(B, \overline{e}_1) \to b\,(B, \overline{0}) \mid b, \qquad\qquad (B, \overline{e}_2) \to b\,(B, \overline{0}) \mid b$$

and $(S, \overline{e}_2)$ as start symbol, that generates the non semilinear language

$$L_{\mathrm{nsm}} = \left\{ a^n b^m \mid 0 \le n, 0 < m \le 2^n \right\}.$$

Proposition 14 (Hopcroft and Pansiot, 1979)**.** *There exist non semilinear Petri nets languages.*

The non semilinearity of MLIGs entails that of all the grammatical formalisms mentioned next in Section 3.2; this answers in particular a conjecture by Kallmeyer (2001) about the semilinearity of V-TAGs.

3.2 Dominance Links

UVG-dl Rambow (1994b) introduced UVG-dls as a formal model for scrambling and tree description grammars.

Definition 15 (Rambow, 1994b)**.** An *unordered vector grammars with dominance links* (UVG-dl) is a tuple $\mathcal{G} = \langle N, \Sigma, W, S \rangle$ where N and Σ are disjoint finite sets of nonterminals and terminals, $V = N \cup \Sigma$ is the vocabulary, W is a set of vectors of productions with dominance links, i.e. each element of W is a pair (P, D) where each P is a multiset of productions in $N \times V^*$ and D is a relation from nonterminals in the right parts of productions in P to nonterminals in their left parts, and S in N is the start symbol.

A *terminal derivation* of w in Σ^* in an UVG-dl is a context-free derivation of form $S \xRightarrow{p_1} \alpha_1 \xRightarrow{p_2} \alpha_2 \cdots \alpha_{p-1} \xRightarrow{p_p} w$ such that the control word $p_1 p_2 \cdots p_p$ is a permutation of a member of W^* and the dominance relations of W hold in the associated derivation tree. The *language* $L(\mathcal{G})$ of an UVG-dl $\mathcal{G}$ is the set of sentences w with some terminal derivation. We write $\mathcal{L}(\text{UVG-dl})$ for the class of UVG-dl languages.

An alternative semantics of derivations in UVG-dls is simply their translation into MLIGs: associate with each nonterminal in a derivation the multiset of productions it has to spawn. Figure 4 presents the two vectors of an UVG-dl for the MIX language of Example 2, with dashed arrows indicating dominance links. Observe that production $S \to S$ in the second vector has to spawn eventually one occurrence of each $S \to aS$, $S \to bS$, and $S \to cS$, which corresponds exactly to the MLIG of Example 2.

---

[3]Adding terminal symbols c in each production would result in a lexicalized grammar, still with a non semilinear language.
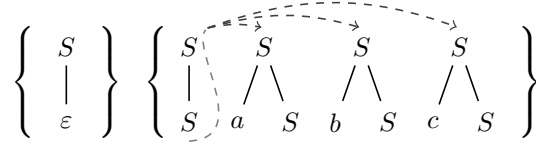
Figure 4: An UVG-dl for L_{mix}.

The ease of translation from the grammar of Figure 4 into a MLIG stems from the impossibility of splitting any of its vectors (P, D) into two nonempty ones (P_1, D_1) and (P_2, D_2) while preserving the dominance relation, i.e. with $P = P_1 \uplus P_2$ and $D = D_1 \uplus D_2$. This *strictness* property can be enforced without loss of generality since we can always add to each vector (P, D) a production $S \rightarrow S$ with a dominance link to each production in P. This was performed on the second vector in Figure 4; remark that the grammar without this addition is an *unordered vector grammar* (Cremers and Mayer, 1974, UVG), and still generates L_{mix}.

Theorem 16 (Rambow, 1994b). *Every MLIG can be transformed into an equivalent UVG-dl in logarithmic space, and conversely.*

Proof sketch. One can check that Rambow (1994b)'s proof of the inclusion $\mathcal{L}(\mathrm{MLIG}) \subseteq \mathcal{L}(\mathrm{UVG\text{-}dl})$ incurs at most a quadratic blowup from a MLIG in RINF, and invoke Proposition 3. More precisely, given a MLIG in RINF, productions of form $(A, \overline{0}) \rightarrow \alpha$ with A in N and α in $(\Sigma \cup (N \times \{\overline{0}\}))^*$ form singleton vectors, and productions of form $(A, \overline{0}) \rightarrow (B, \overline{e}_i)$ with A, B in N and $0 < i \leq n$ need to be paired with a production of form $(C, \overline{e}_i) \rightarrow (D, \overline{0})$ for some C and D in N in order to form a vector with a dominance link between B and C.

The converse inclusion and its complexity are immediate when considering strict UVG-dls. □

The restrictions to k-ranked and k-bounded grammars find natural counterparts in strict UVG-dls by bounding the (total) number of pending dominance links in any derivation. Lexicalization has now its usual definition: for every vector $(\{p_{i,1}, \ldots, p_{i,k_i}\}, D_i)$ in W, at least one of the $p_{i,j}$ should contain at least one terminal in its right part—we have then $\mathcal{L}(\mathrm{UVG\text{-}dl}_\ell) \subseteq \mathcal{L}(\mathrm{MLIG}_\ell)$.

More on Dominance Links Dominance links are quite common in tree description formalisms, where they were already in use in D-theory (Marcus et al., 1983) and in quasi-tree semantics for fbTAGs (Vijay-Shanker, 1992). In particular, D-tree substitution grammars are essentially the same as UVG-dls (Rambow et al., 2001), and quite a few other tree description formalisms subsume them (Candito and Kahane, 1998; Kallmeyer, 2001; Guillaume and Perrier, 2010). Another class of grammars are *vector TAGs* (V-TAGs), which extend TAGs and MCTAGs using dominance links (Becker et al., 1991; Rambow, 1994a; Champollion, 2007), subsuming again UVG-dls.

4 Computational Complexity

We study in this section the complexity of several decision problems on MLIGs, prominently of emptiness and membership problems, in the general (Section 4.2),

9

k-bounded (Section 4.3), and lexicalized cases (Section 4.4). Table 1 sums up the known complexity results. Since by Theorem 16 we can translate between MLIGs and UVG-dls in logarithmic space, the complexity results on UVG-dls will be the same.

4.1 Decision Problems

Let us first review some decision problems of interest. In the following, $\mathcal{G}$ denotes a MLIG $\langle N, \Sigma, P, (S, \overline{x}_0) \rangle$:

boundedness given $\langle \mathcal{G} \rangle$, is $\mathcal{G}$ bounded? As seen in Section 2.2, this is equivalent to rankedness.

k-boundedness given $\langle \mathcal{G}, k \rangle$, k in $\mathbb{N}$, is $\mathcal{G}$ k-bounded? As seen in Section 2.2, this is the same as (kn)-rankedness. Here we will distinguish two cases depending on whether k is encoded in unary or binary.

coverability given $\langle \mathcal{G}, F \rangle$, $\mathcal{G}$ ε-free in ETF and F a finite subset of $N \times \mathbb{N}^n$, does there exist $\alpha = (A_1, \overline{y}_1) \cdots (A_m, \overline{y}_m)$ in $(N \times \mathbb{N}^n)^*$ such that $(S, \overline{x}_0) \Rightarrow^* \alpha$ and for each $0 < j \leq m$ there exists $(A_j, \overline{x}_j)$ in F with $\overline{x}_j \leq \overline{y}_j$?

reachability given $\langle \mathcal{G}, F \rangle$, $\mathcal{G}$ ε-free in ETF and F a finite subset of $N \times \mathbb{N}^n$, does there exist $\alpha = (A_1, \overline{y}_1) \cdots (A_m, \overline{y}_m)$ in F^* such that $(S, \overline{x}_0) \Rightarrow^* \alpha$?

non emptiness given $\langle \mathcal{G} \rangle$, is $L(\mathcal{G})$ non empty?

(uniform) membership given $\langle \mathcal{G}, w \rangle$, w in Σ^*, does w belong to $L(\mathcal{G})$?

Boundedness and k-boundedness are needed in order to prove that a grammar is bounded, and to apply the smaller complexities of Section 4.3. Coverability is often considered for Petri nets, and allows to derive lower bounds on reachability. Emptiness is the most basic static analysis one might want to perform on a grammar, and is needed for *parsing as intersection* approaches (Lang, 1994), while membership reduces to parsing. Note that we only consider uniform membership, since grammars for natural languages are typically considerably larger than input sentences, and their influence can hardly be neglected.

There are several obvious reductions between reachability, emptiness, and membership. Let $\rightarrow_{\log}$ denote LogSpace reductions between decision problems; we have:

Proposition 17.

$$\text{coverability} \rightarrow_{\log} \text{reachability} \tag{1}$$
$$\leftrightarrow_{\log} \text{non emptiness} \tag{2}$$
$$\leftrightarrow_{\log} \text{membership} \tag{3}$$

Proof sketch. For (1), construct a reachability instance $\langle \mathcal{G}', \{(E, \overline{0})\} \rangle$ from a coverability instance $\langle \mathcal{G}, F \rangle$ by adding to $\mathcal{G}$ a fresh nonterminal E and the productions

$$\{(A, \overline{x}) \rightarrow (E, \overline{0}) \mid (A, \overline{x}) \in F\}$$
$$\cup \{(E, \overline{e}_i) \rightarrow (E, \overline{0}) \mid 0 < i \leq n\} \, .$$

For (2), from a reachability instance $\langle \mathcal{G}, F \rangle$, remove all terminal productions from $\mathcal{G}$ and add instead the productions $\{(A, \overline{x}) \rightarrow \varepsilon \mid (A, \overline{x}) \in F\}$; the new

Problem	Lower bound	Upper bound
Petri net k-Boundedness	PSPACE (Jones et al., 1977)	PSPACE (Jones et al., 1977)
Petri net Boundedness	ExpSpace (Lipton, 1976)	ExpSpace (Rackoff, 1978)
Petri net {Emptiness, Membership}	ExpSpace (Lipton, 1976)	Decidable, not primitive recursive (Mayr, 1981; Kosaraju, 1982)
{MLIG, MLIG$_\ell$} k-Boundedness	ExpTime (Corollary 20)	ExpTime (Corollary 20)
{MLIG, MLIG$_\ell$} Boundedness	2ExpTime (Demri et al., 2009)	2ExpTime (Demri et al., 2009)
{MLIG, MLIG$_\ell$} Emptiness MLIG Membership	2ExpSpace (Lazić, 2010)	Not known to be decidable
{kb-MLIG, kb-MLIG$_\ell$} Emptiness kb-MLIG Membership	ExpTime (Theorem 19)	ExpTime (Theorem 19)
{MLIG$_\ell$, kb-MLIG$_\ell$} Membership	NPTime (Koller and Rambow, 2007)	NPTime (trivial)
kr-MLIG {Emptiness, Membership}	PTime (Jones and Laaser, 1976)	PTime (Lemma 6)

Table 1: Summary of complexity results.

grammar $\mathcal{G}'$ has a non empty language iff the reachability instance was positive. Conversely, from a non emptiness instance $\langle \mathcal{G} \rangle$, put the grammar in ETF and define F to match all terminal productions, i.e. $F = \{(A, \overline{x}) \mid (A, \overline{x}) \to a \in P, a \in \Sigma \cup \{\varepsilon\}\}$, and then remove all terminal productions in order to obtain a reachability instance $\langle \mathcal{G}', F \rangle$.

For (3), from a non emptiness instance $\langle \mathcal{G} \rangle$, replace all terminals in $\mathcal{G}$ by ε to obtain an empty word membership instance $\langle \mathcal{G}', \varepsilon \rangle$. Conversely, from a membership instance $\langle \mathcal{G}, w \rangle$, construct the intersection grammar $\mathcal{G}'$ with $L(\mathcal{G}') = L(\mathcal{G}) \cap \{w\}$ (Bar-Hillel et al., 1961), which serves as non emptiness instance $\langle \mathcal{G}' \rangle$. □

4.2 General Case

Verma and Goubault-Larrecq (2005) were the first to prove that coverability and boundedness were decidable for BVASS, using a covering tree construction à la Karp and Miller (1969), thus of non primitive recursive complexity. Demri et al. (2009, Theorems 7, 17, and 18) recently proved tight complexity bounds for these problems, extending earlier results by Rackoff (1978) and Lipton (1976) for Petri nets.

Theorem 18 (Demri et al., 2009). *Coverability and boundedness for MLIGs are* 2ExpTime-*complete.*

Regarding reachability, emptiness, and membership, decidability is still open. A 2ExpSpace lower bound was recently found by Lazić (2010). If a decision procedure exists, we can expect it to be quite complex, as already in the Petri net case, the complexity of the known decision procedures (Mayr, 1981; Kosaraju, 1982) is not primitive recursive (Cardoza et al., 1976, who attribute the idea to Hack).

4.3 k-Bounded and k-Ranked Cases

Since k-bounded MLIGs can be converted into CFGs (Lemma 8), emptiness and membership problems are decidable, albeit at the expense of an exponential

11

blowup. We know from the Petri net literature that coverability and reachability problems are PSPACE-complete for k-bounded right linear MLIGs (Jones et al., 1977) by a reduction from *linear bounded automaton* (LBA) membership. We obtain the following for k-bounded MLIGs, using a similar reduction from membership in polynomially space bounded *alternating Turing machines* (Chandra et al., 1981, ATM):

Theorem 19. *Coverability and reachability for k-bounded MLIGs are* ExpTime-*complete, even for fixed $k \geq 1$.*

The lower bound is obtained through an encoding of an instance of the membership problem for ATMs working in polynomial space into an instance of the coverability problem for 1-bounded MLIGs. The upper bound is a direct application of Lemma 8, coverability and reachability being reducible to the emptiness problem for a CFG of exponential size. Theorem 19 also shows the ExpTime-hardness of emptiness and membership in minimalist grammars with SMC.

Corollary 20. *Let $k \geq 1$; k-boundedness for MLIGs is* ExpTime-*complete.*

Proof. For the lower bound, consider an instance $\langle \mathcal{G}, F \rangle$ of coverability for a 1-bounded MLIG $\mathcal{G}$, which is ExpTime-hard according to Theorem 19. Add to the MLIG $\mathcal{G}$ a fresh nonterminal E and the productions

$$\{(A, \bar{\mathsf{x}}) \to (E, \bar{\mathsf{x}}) \mid (A, \bar{\mathsf{x}}) \in F\}$$
$$\cup \, \{(E, \bar{\mathsf{0}}) \to (E, \bar{\mathsf{e}}_i) \mid 0 < i \leq n\} \, ,$$

which make it non k-bounded iff the coverability instance was positive.

For the upper bound, apply Lemma 8 with $k' = k + 1$ to construct an $O(|\mathcal{G}| \cdot 2^{n^2 \log_2(k'+1)})$-sized CFG, reduce it in polynomial time, and check whether a nonterminal $(A, \bar{\mathsf{x}})$ with $\bar{\mathsf{x}}(i) = k'$ for some $0 < i \leq n$ occurs in the reduced grammar.

Note that the choice of the encoding of k is irrelevant, as $k = 1$ is enough for the lower bound, and k only logarithmically influences the exponent for the upper bound. $\square$

Corollary 20 also implies the ExpTime-completeness of k-rankedness, k encoded in unary, if k can take arbitrary values. On the other hand, if k is known to be small, for instance logarithmic in the size of $\mathcal{G}$, then k-rankedness becomes polynomial by Lemma 6.

Observe finally that k-rankedness provides the only tractable class of MLIGs for uniform membership, using again Lemma 6 to obtain a CFG of polynomial size—actually exponential in k, but k is assumed to be fixed for this problem. An obvious lower bound is that of membership in CFGs, which is PTime-complete (Jones and Laaser, 1976).

4.4 Lexicalized Case

Unlike the high complexity lower bounds of the previous two sections, NPTime-hardness results for uniform membership have been proved for a number of formalisms related to MLIGs, from the commutative CFG viewpoint Huynh (1983); Barton (1985); Esparza (1995), or from more specialized models (Søgaard et al., 2007; Champollion, 2007; Koller and Rambow, 2007). We focus here on this last

proof, which reduces from the *normal dominance graph configurability* problem (Althaus et al., 2003), as it allows to derive NPTime-hardness even in highly restricted grammars.

Theorem 21 (Koller and Rambow, 2007). *Uniform membership of $\langle \mathcal{G}, w \rangle$ for $\mathcal{G}$ a 1-bounded, lexicalized, UVG-dl with finite language is* NPTime-*hard, even for $|w| = 1$.*

Proof sketch. Set S as start symbol and add a production $S \rightarrow aA$ to the sole vector of the grammar $\mathcal{G}$ constructed by Koller and Rambow (2007) from a normal dominance graph, with dominance links to all the other productions. Then $\mathcal{G}$ becomes strict, lexicalized, with finite language $\{a\}$ or $\emptyset$, and 1-bounded, such that a belongs to $L(\mathcal{G})$ iff the normal dominance graph is configurable. $\square$

The fact that uniform membership is in NPTime in the lexicalized case is clear, as we only need to guess nondeterministically a derivation of size linear in $|w|$ and check its correctness.

The weakness of lexicalized grammars is however that their emptiness problem is not any easier to solve! The effect of lexicalization is indeed to break the reduction from emptiness to membership in Proposition 17, but emptiness is as hard as ever, which means that static checks on the grammar might even be undecidable.

5 Conclusion

Grammatical formalisms with dominance links, introduced in particular to model scrambling phenomena in computational linguistics, have deep connections with several open questions in an unexpected variety of fields in computer science. We hope this survey to foster cross-fertilizing exchanges; for instance, is there a relation between Conjecture 11 and the decidability of reachability in MLIGs? A similar question, whether the language L_{pal} of even 2-letters palindromes was a Petri net language, was indeed solved using the decidability of reachability in Petri nets (Jantzen, 1979), and shown to be strongly related to the latter (Lambert, 1992).

A conclusion with a more immediate linguistic value is that MLIGs and UVG-dls hardly qualify as formalisms for *mildly context-sensitive languages*, claimed by Joshi (1985) to be adequate for modeling natural languages, and "roughly" defined as the extensions of context-free languages that display

1. support for *limited cross-serial dependencies*: seems doubtful, see Conjecture 11,

2. constant growth, a requisite nowadays replaced by *semilinearity*: does not hold, as seen with Proposition 14, and

3. *polynomial recognition* algorithms: holds only for restricted classes of grammars, as seen in Section 4.

Nevertheless, variants such as k-ranked V-TAGs are easily seen to fulfill all the three points above.

13

References

Althaus, E., Duchier, D., Koller, A., Mehlhorn, K., Niehren, J., and Thiel, S., 2003. An efficient graph algorithm for dominance constraints. *Journal of Algorithms*, 48(1):194–219. doi:10.1016/S0196-6774(03)00050-6.

Bar-Hillel, Y., Perles, M., and Shamir, E., 1961. On formal properties of simple phrase structure grammars. *Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung*, 14:143–172.

Barton, G.E., 1985. The computational difficulty of ID/LP parsing. In *ACL'85*, pages 76–81. ACL Press. doi:10.3115/981210.981220.

Becker, T., Joshi, A.K., and Rambow, O., 1991. Long-distance scrambling and tree adjoining grammars. In *EACL'91*, pages 21–26. ACL Press. doi: 10.3115/977180.977185.

Bojańczyk, M., Muscholl, A., Schwentick, T., and Segoufin, L., 2009. Two-variable logic on data trees and XML reasoning. *Journal of the ACM*, 56(3): 1–48. doi:10.1145/1516512.1516515.

Candito, M.H. and Kahane, S., 1998. Defining DTG derivations to get semantic graphs. In *TAG+4*, pages 25–28.

Cardoza, E., Lipton, R.J., and Meyer, A.R., 1976. Exponential space complete problems for Petri nets and commutative semigroups: Preliminary report. In *STOC'76*, pages 50–54. ACM Press. doi:10.1145/800113.803630.

Champollion, L., 2007. Lexicalized non-local MCTAG with dominance links is NP-complete. In *MOL 10*.

Chandra, A.K., Kozen, D.C., and Stockmeyer, L.J., 1981. Alternation. *Journal of the ACM*, 28(1):114–133. doi:10.1145/322234.322243.

Chiang, D. and Scheffler, T., 2008. Flexible composition and delayed tree-locality. In *TAG+9*.

Cremers, A.B. and Mayer, O., 1974. On vector languages. *Journal of Computer and System Sciences*, 8(2):158–166. doi:10.1016/S0022-0000(74)80053-X.

de Groote, P., Guillaume, B., and Salvati, S., 2004. Vector addition tree automata. In *LICS'04*, pages 64–73. IEEE Computer Society. ISBN 0-7695-2192-4. doi:10.1109/LICS.2004.51.

Demri, S., Jurdziński, M., Lachish, O., and Lazić, R., 2009. The covering and boundedness problems for branching vector addition systems. In Kannan, R. and Narayan Kumar, K., editors, *FSTTCS'09*, volume 4 of *Leibniz International Proceedings in Informatics*, pages 181–192. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSTTCS.2009.2317.

Dufourd, C. and Finkel, A., 1999. A polynomial λ-bisimilar normalization for reset Petri nets. *Theoretical Computer Science*, 222(1–2):187–194. doi: 10.1016/S0304-3975(98)00351-X.

Esparza, J., 1995. Petri nets, commutative context-free grammars, and basic parallel processes. In Reichel, H., editor, *FCT'95*, volume 965 of *Lecture Notes in Computer Science*, pages 221–232. Springer. doi:10.1007/3-540-60249-6_54.

Guillaume, B. and Perrier, G., 2010. Interaction grammars. *Research on Language and Computation*. doi:10.1007/s11168-010-9066-x. To appear.

Haddad, S. and Poitrenaud, D., 2007. Recursive Petri nets. *Acta Informatica*, 44(7–8):463–508. doi:10.1007/s00236-007-0055-y.

Hopcroft, J. and Pansiot, J.J., 1979. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8(2): 135–159. doi:10.1016/0304-3975(79)90041-0.

Huynh, D.T., 1983. Commutative grammars: the complexity of uniform word problems. *Information and Control*, 57(1):21–39. doi:10.1016/S0019-9958(83)80022-9.

Jantzen, M., 1979. On the hierarchy of Petri net languages. *RAIRO Theoretical Informatics and Applications*, 13(1):19–30.

Jones, N.D. and Laaser, W.T., 1976. Complete problems for deterministic polynomial time. *Theoretical Computer Science*, 3(1):105–117. doi: 10.1016/0304-3975(76)90068-2.

Jones, N.D., Landweber, L.H., and Lien, Y.E., 1977. Complexity of some problems in Petri nets. *Theoretical Computer Science*, 4(3):277–299. doi: 10.1016/0304-3975(77)90014-7.

Joshi, A.K., 1985. Tree-adjoining grammars: How much context sensitivity is required to provide reasonable structural descriptions? In Dowty, D.R., Karttunen, L., and Zwicky, A.M., editors, *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*, chapter 6, pages 206–250. Cambridge University Press.

Joshi, A.K., Becker, T., and Rambow, O., 2000. Complexity of scrambling: A new twist to the competence-performance distinction. In Abeillé, A. and Rambow, O., editors, *Tree Adjoining Grammars. Formalisms, Linguistic Analysis and Processing*, chapter 6, pages 167–181. CSLI Publications.

Kallmeyer, L., 2001. Local tree description grammars. *Grammars*, 4(2):85–137. doi:10.1023/A:1011431526022.

Kallmeyer, L. and Parmentier, Y., 2008. On the relation between multicomponent tree adjoining grammars with tree tuples (TT-MCTAG) and range concatenation grammars (RCG). In Martín-Vide, C., Otto, F., and Fernau, H., editors, *LATA'08*, volume 5196 of *Lecture Notes in Computer Science*, pages 263–274. Springer. doi:10.1007/978-3-540-88282-4_25.

15

Karp, R.M. and Miller, R.E., 1969. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195. doi:10.1016/S0022-0000(69)80011-5.

Koller, A. and Rambow, O., 2007. Relating dominance formalisms. In *FG'07*.

Kosaraju, S.R., 1982. Decidability of reachability in vector addition systems. In *STOC'82*, pages 267–281. ACM Press. doi:10.1145/800070.802201.

Kroch, A.S. and Santorini, B., 1991. The derived constituent structure of the West Germanic verb-raising construction. In Freidin, R., editor, *Principles and Parameters in Comparative Grammar*, chapter 10, pages 269–338. MIT Press.

Lambert, J.L., 1992. A structure to decide reachability in Petri nets. *Theoretical Computer Science*, 99(1):79–104. doi:10.1016/0304-3975(92)90173-D.

Lang, B., 1994. Recognition can be harder than parsing. *Computational Intelligence*, 10(4):486–494. doi:10.1111/j.1467-8640.1994.tb00011.x.

Lazić, R., 2010. The reachability problem for branching vector addition systems requires doubly-exponential space. Manuscript.

Lichte, T., 2007. An MCTAG with tuples for coherent constructions in German. In *FG'07*.

Lipton, R., 1976. The reachability problem requires exponential space. Technical Report 62, Yale University.

Maclachlan, A. and Rambow, O., 2002. Cross-serial dependencies in Tagalog. In *TAG+6*, pages 100–107.

Marcus, M.P., Hindle, D., and Fleck, M.M., 1983. D-theory: talking about talking about trees. In *ACL'83*, pages 129–136. ACL Press. doi:10.3115/981311.981337.

Mayr, E.W., 1981. An algorithm for the general Petri net reachability problem. In *STOC'81*, pages 238–246. ACM Press. doi:10.1145/800076.802477.

Mayr, R., 1999. Process rewrite systems. *Information and Computation*, 156 (1–2):264–286. doi:10.1006/inco.1999.2826.

Papadimitriou, C.H., 1994. *Computational Complexity*. Addison-Wesley. ISBN 0-201-530821.

Parikh, R.J., 1966. On context-free languages. *Journal of the ACM*, 13(4): 570–581. doi:10.1145/321356.321364.

Peterson, J.L., 1981. *Petri Net Theory and the Modeling of Systems*. Prentice Hall.

Petri, C.A., 1962. *Kommunikation mit Automaten*. PhD thesis, University of Bonn.

Rackoff, C., 1978. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223–231. doi:10.1016/0304-3975(78)90036-1.

Rambow, O., 1994a. *Formal and Computational Aspects of Natural Language Syntax*. PhD thesis, University of Pennsylvania.

Rambow, O., 1994b. Multiset-valued linear index grammars: imposing dominance constraints on derivations. In *ACL'94*, pages 263–270. ACL Press. doi:10.3115/981732.981768.

Rambow, O., Vijay-Shanker, K., and Weir, D., 1995. D-tree grammars. In *ACL'95*, pages 151–158. ACL Press. doi:10.3115/981658.981679.

Rambow, O., Weir, D., and Vijay-Shanker, K., 2001. D-tree substitution grammars. *Computational Linguistics*, 27(1):89–121. doi:10.1162/089120101300346813.

Salvati, S., 2010. Minimalist grammars in the light of logic. Manuscript.

Shieber, S.M., 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8(3):333–343. doi:10.1007/BF00630917.

Søgaard, A., Lichte, T., and Maier, W., 2007. The complexity of linguistically motivated extensions of tree-adjoining grammar. In *RANLP'07*, pages 548–553. http://www.sfs.uni-tuebingen.de/~wmaier/pub/ranlp07.pdf.

Stabler, E.P., 1997. Derivational minimalism. In Retoré, C., editor, *LACL'96*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer. doi:10.1007/BFb0052147.

Verma, K.N. and Goubault-Larrecq, J., 2005. Karp-Miller trees for a branching extension of VASS. *Discrete Mathematics and Theoretical Computer Science*, 7(1):217–230. http://www.dmtcs.org/volumes/abstracts/dm070113.abs.html.

Vijay-Shanker, K., 1992. Using descriptions of trees in a tree adjoining grammar. *Computational Linguistics*, 18(4):481–517. http://www.aclweb.org/anthology/J92-4004.

Yoshinaka, R. and Kanazawa, M., 2005. The complexity and generative capacity of lexicalized abstract categorial grammars. In Blache, P., Stabler, E., Busquets, J., and Moot, R., editors, *LACL'05*, volume 3492 of *Lecture Notes in Computer Science*, pages 330–346. Springer. doi:10.1007/11422532_22.

A Complements to Section 2

This section details the proof of the following proposition, which was omitted from the main text:

Proposition 3. *For any MLIG, one can construct an equivalent MLIG in RINF in logarithmic space.*

As explained in Section 2, the difficulty lies in avoiding an exponential blowup when constructing the MLIG in RINF. The idea is to proceed in two steps, first by constructing a grammar in *ordinary form* (OF) (Lemma 23), and then by translating this grammar in OF into a grammar in RINF (Lemma 24). This construction is akin to the normalization presented by Dufourd and Finkel (1999) for reset Petri nets.

Definition 22. A MLIG is in *ordinary form* if, for any production of form $(\star)$ in P, for any vector $\overline{\mathsf{y}}$ in $\{\overline{\mathsf{x}}_0\} \cup \{\overline{\mathsf{x}}\} \cup \{\overline{\mathsf{x}}_j \mid 1 \leq j \leq m\}$ that appears in the start symbol or in this production, and for any index $0 < i \leq n$, $\overline{\mathsf{y}}(i) \leq 1$.

Lemma 23. *For any MLIG, one can construct an equivalent MLIG in OF in logarithmic space.*

Proof. Let us fix a MLIG $\mathcal{G} = \langle N, \Sigma, P, (S, \overline{\mathsf{x}}_0) \rangle$. We first define the *maximal vector value* of $\mathcal{G}$ as the minimum integer $\max_{\mathcal{G}}$ such that, for any production of form $(\star)$ in P, for any vector $\overline{\mathsf{y}}$ in $\{\overline{\mathsf{x}}_0\} \cup \{\overline{\mathsf{x}}\} \cup \{\overline{\mathsf{x}}_j \mid 1 \leq j \leq m\}$ that appears in the start symbol or in this production, and for any index $0 < i \leq n$, $\overline{\mathsf{y}}(i) \leq \max_{\mathcal{G}}$. Thus a MLIG in OF is one where $\max_{\mathcal{G}} \leq 1$.

Let $n' = \lfloor \max_{\mathcal{G}} \rfloor$ (thus of logarithmic size); the idea in the following is to increase the dimension to $n(n'+1)$ and use the additional indices to encode vector values in binary.

Let us fix some notation: each index $0 < i \leq n$ of $\mathcal{G}$ is associated with $n'+1$ indices in the constructed grammar $\mathcal{G}'$. The index (i,j) denotes the jth such index, $0 \leq j \leq n'$, with the convention $(i,0) = i$. For every nonterminal A of N, and every $0 < i \leq n$, and every $0 < j \leq n'$, we add the nonterminals $A_{i,j}$ and $A'_{i,j}$: let

$$N' = N \cup \{A_{i,j} \mid 0 < i \leq n, 0 < j \leq n'\}$$
$$\cup \{A'_{i,j} \mid 0 < i \leq n, 0 < j \leq n'\}.$$

These nonterminals will handle the conversions to and from binary: we define the productions

$$P_{A,i} = \{(A, \overline{\mathsf{e}}_{i,0}) \to (A_{i,1}, \overline{\mathsf{0}})\}$$
$$\cup \{(A_{i,j}, \overline{\mathsf{e}}_{i,j}) \to (A_{i,j+1}, \overline{\mathsf{0}}) \mid 0 < j \leq n'\}$$
$$\cup \{(A_{i,j}, \overline{\mathsf{0}}) \to (A, \overline{\mathsf{e}}_{i,j}) \mid 0 < j \leq n'\}$$
$$P'_{A,i} = \{(A, \overline{\mathsf{e}}_{i,j}) \to (A'_{i,j}, \overline{\mathsf{0}}) \mid 0 < j \leq n'\}$$
$$\cup \{(A'_{i,j}, \overline{\mathsf{0}}) \to (A'_{i,j-1}, \overline{\mathsf{e}}_{i,j-1}) \mid 1 < j \leq n'\}$$
$$\cup \{(A'_{i,1}, \overline{\mathsf{0}}) \to (A, \overline{\mathsf{e}}_{i,0})\}$$

for all A in N and $0 < i \leq n$. We want to prove that this set of productions performs a binary encoding of the contents of the ith index, i.e. that

$$\overline{\mathsf{y}}(i,0) + \sum_{j=1}^{n'} \overline{\mathsf{y}}(i,j)2^{j-1} = \overline{\mathsf{y}}'(i,0) + \sum_{j=1}^{n'} \overline{\mathsf{y}}'(i,j)2^{j-1} \tag{4}$$

i

holds whenever $(A, \overline{y}) \Rightarrow^* (A, \overline{y}')$ using productions from $P_{A,i} \cup P'_{A,i}$.

Claim 23.1. If $(A, \overline{y}) \Rightarrow^p (A, \overline{y}')$, for some $p \geq 0$ and using only productions from $P_{A,i} \cup P'_{A,i}$, then (4) holds.

We prove the claim by induction on p, using productions from $P_{A,i}$ solely; the case of $P'_{A,i}$ is symmetric—and there is no possible interference between the two sets of productions.

The claim holds vacuously for $p = 0$. For $p > 0$, we can split the derivation into

$$(A, \overline{y}) \Rightarrow^{p-1} (A_{i,j}, \overline{y}_{p-1}) \Rightarrow (A, \overline{y}')$$

with

$$\overline{y}_{p-1}(i, j) + 1 = \overline{y}'(i, j)$$

for some $0 < j \leq n'$—using the last ruleset of $P_{A,i}$—, and we can distinguish two cases:

1. $(A, \overline{y}) \Rightarrow^{p-2} (A, \overline{y}_{p-2}) \Rightarrow (A_{i,j}, \overline{y}_{p-1})$, which enforces $j = 1$, and then

$$\begin{aligned}
\overline{y}_{p-2}(i, 0) &= \overline{y}_{p-1}(i, 0) + 1 \\
&= \overline{y}'(i, 0) + 1 \text{ and} \\
\overline{y}_{p-2}(i, 1) + 1 &= \overline{y}_{p-1}(i, 1) + 1 \\
&= \overline{y}'(i, 1) ,
\end{aligned}$$

thus (4) holds between $\overline{y}'$ and $\overline{y}_{p-2}$, and using the induction hypothesis on derivation $(A, \overline{y}) \Rightarrow^{p-2} (A, \overline{y}_{p-2})$, it also holds for the entire derivation.

2. $(A, \overline{y}) \Rightarrow^{p-2} (A_{i,j-1}, \overline{y}_{p-2}) \Rightarrow (A_{i,j}, \overline{y}_{p-1})$ with $\overline{y}_{p-2}(i, j-1) = \overline{y}_{p-1}(i, j-1) + 1$, then $(A, \overline{y}) \Rightarrow^{p-2} (A_{i,j-1}, \overline{y}_{p-2}) \Rightarrow (A, \overline{y}'')$ with $\overline{y}_{p-2}(i, j-1) + 1 = \overline{y}''(i, j-1)$ when applying the last ruleset of $P_{A,i}$ to $(A_{i,j-1}, \overline{y}_{p-2})$, thus

$$\begin{aligned}
\overline{y}_{p-1}(i, j-1) + 2 &= \overline{y}''(i, j-1) \\
&= \overline{y}'(i, j-1) + 2 \\
\overline{y}_{p-1}(i, j) + 1 &= \overline{y}''(i, j) + 1 \\
&= \overline{y}'(i, j) ,
\end{aligned}$$

and therefore (4) holds between $\overline{y}'$ and $\overline{y}''$. Applying the induction hypothesis to $(A, \overline{y}) \Rightarrow^{p-2} (A_{i,j-1}, \overline{y}_{p-2}) \Rightarrow (A, \overline{y}'')$ yields the claim.

It remains to modify the productions of P in order to use the new indices. Let $\overline{x}$ be a vector of $\mathbb{N}^n$: its *binary encoding* is the vector $b\overline{x}$ in $\mathbb{N}^{n(n'+1)}$ such that, for all $0 < i \leq n$,

$$\overline{x}(i) = \sum_{j=1}^{n'} b\overline{x}(i, j) 2^{j-1},$$

$$b\overline{x}(i, 0) = 0, \text{ and}$$

$$b\overline{x}(i, j) \leq 1 \text{ for all } 0 < j \leq n' ,$$

the point being that $|b\overline{x}|$ is polynomial in $|\overline{x}|$. We construct a new set of productions accordingly, with a production

$$(A, b\overline{x}) \rightarrow u_0(B_1, b\overline{x}_1) u_1 \cdots u_m(B_m, b\overline{x}_m) u_{m+1}$$

for each production of form $(\star)$ in P. Let us dub P' the set of productions that gathers these binary encodings and the productions of $P_{A,i} \cup P'_{A,i}$ for each A in N and $0 < i \le n$.

Claim 23.2. The $(n(n'+1))$-dimensional MLIG $\mathcal{G}' = \langle N', \Sigma, P', (S, b\overline{\mathsf{x}}_0) \rangle$ is in OF and equivalent to $\mathcal{G}$.

The fact that $\mathcal{G}'$ is in OF is immediate by definition of the binary encoding $b\overline{\mathsf{x}}$ and of the productions of P'. The equivalence of $\mathcal{G}$ and $\mathcal{G}'$ stems from Claim 23.1 and the properties of $b\overline{\mathsf{x}}$.

We can conclude by noting that, indeed, $\mathcal{G}'$ can be constructed from $\mathcal{G}$ in logarithmic space. $\qquad\square$

Lemma 24. *For any MLIG in OF, one can construct an equivalent MLIG in RINF in logarithmic space.*

Proof. The construction presented by Rambow (1994a, Theorem 3) fits in the OF case. $\qquad\square$

B Complements to Section 4

B.1 Complexity for k-Bounded MLIGs

This section contains the proof of the following result:

Theorem 19. *Coverability and reachability for k-bounded MLIGs are* ExpTime-*complete, even for fixed $k \ge 1$.*

B.1.1 Lower Bound

We reduce the membership problem for an alternating Turing machine operating in polynomial space to the coverability problem for a 1-bounded MLIG, which yields its ExpTime-hardness (Chandra et al., 1981).

Formally, we are given an ATM $\mathcal{M} = \langle Q, \Sigma, \Gamma, \delta, q_0, F \rangle$, an input string w in Σ^*, and the insurance that $\mathcal{M}$ will never visit more than $p(|w|)$ cells of its tape. Wlog., we consider $\delta(q, Z)$ for a state q in Q and a tape content Z in Γ to be (q_1, Z_1, d_1) op (q_2, Z_2, d_2) with q_1, q_2 in Q, $Z_1 \ne Z$ and $Z_2 \ne Z$ in Γ, d_1, d_2 in $\{-1, +1\}$ (standing for a move to the left or to the right), and op in $\{\vee, \wedge\}$ (standing for disjunction or conjunction).

Encoding ATM Configurations The total number of different tape contents of $\mathcal{M}$ is bounded by $|\Gamma|^{p(|w|)}$, which we cannot afford to represent explicitly. Instead, we store the current tape contents of $\mathcal{M}$ as a vector of dimension $c = |\Gamma| \cdot p(|w|)$, and maintain it throughout the simulation by our MLIG. A difficulty arises with conjunctive transitions $\delta(q, Z) = (q_1, Z_1, d_1) \wedge (q_2, Z_2, d_2)$, which cannot be directly simulated by MLIG derivations of form $(A, \overline{\mathsf{y}}) \Rightarrow (B_1, \overline{\mathsf{y}}_1)(B_2, \overline{\mathsf{y}}_2)$ with $(A, \overline{\mathsf{y}})$ encoding the configuration matched by (q, Z), and each $(B_j, \overline{\mathsf{y}}_j)$ encoding the new configuration corresponding to the (q_j, Z_j, d_j) action. Vector values from $\overline{\mathsf{y}}$, encoding the current tape configuration, are scattered nondeterministically between $\overline{\mathsf{y}}_1$ and $\overline{\mathsf{y}}_2$. The solution is to construct a 1-bounded MLIG with enough redundancy to recover "clean" tape configurations after the simulation of a conjunctive transition.

$$\{([q,i], \langle Z,i\rangle_1 + \langle Z,i\rangle_2 + \overline{\langle Z_j,i\rangle}_1 + \overline{\langle Z_j,i\rangle}_2) \to$$
$$([q_j, i+d_j], \overline{\langle Z,i\rangle}_1 + \overline{\langle Z,i\rangle}_2 + \langle Z_j,i\rangle_1 + \langle Z_j,i\rangle_2) \mid i \leq p(|w|), j \in \{1,2\}\}$$

Figure 5: The productions encoding a transition $\delta(q,Z) = (q_1,Z_1,d_1) \vee (q_2,Z_2,d_2)$.

$$\{([q,i], \langle Z,i\rangle_1 + \langle Z,i\rangle_2 + \overline{\langle Z_1,i\rangle}_1 + \overline{\langle Z_2,i\rangle}_2) \to$$
$$\bigodot_{j\in\{1,2\}} \left([q_j, i+d_j]_j, \overline{\langle Z,i\rangle}_j + \langle Z_j,i\rangle_j + \sum_{Z'\in\Gamma, m\leq p(|w|)} \langle Z',m\rangle_c \right) \mid i \leq p(|w|)\}$$
$$\cup \{([q,i]_j, \langle Z,i\rangle_j + \langle Z,i\rangle_c) \to ([q,i]_j, \langle Z,i\rangle_j + \langle Z,i\rangle_{3-j} + \overline{\langle Z,i\rangle}_c \mid q \in Q, i \leq p(|w|), j \in \{1,2\}\}$$
$$\cup \{([q,i]_j, \overline{\langle Z,i\rangle}_j + \langle Z,i\rangle_c) \to ([q,i]_j, \overline{\langle Z,i\rangle}_j + \overline{\langle Z,i\rangle}_{3-j} + \overline{\langle Z,i\rangle}_c \mid q \in Q, i \leq p(|w|), j \in \{1,2\}\}$$
$$\cup \{([q,i]_j, \sum_{Z\in\Gamma, i\leq p(|w|)} \overline{\langle Z,i\rangle}_c) \to ([q,i], \bar{0}) \mid q \in Q, i \leq p(|w|), j \in \{1,2\}\}$$

Figure 6: The productions encoding a transition $\delta(q,Z) = (q_1,Z_1,d_1) \wedge (q_2,Z_2,d_2)$.

Accordingly, we set our dimension as $n = 6c$: each (Z,i) pair in $\Gamma \times \{1,\ldots,p(|w|)\}$ is associated with a left and a right coordinate (whose unit vectors are denoted as $\langle Z,i\rangle_1$ and $\langle Z,i\rangle_2$), their complements (denoted as $\overline{\langle Z,i\rangle}_1$ and $\overline{\langle Z,i\rangle}_2$), and two counts (denoted as $\langle Z,i\rangle_c$ and $\overline{\langle Z,i\rangle}_c$).

We also define our set of nonterminals as

$$N = \bigcup_{q\in Q, i\leq p(|w|)} \{[q,i], [q,i]_1, [q,i]_2\},$$

recording the current state and current head position on the tape. Hence a pair in $N \times \mathbb{N}^n$ represents the current configuration of $\mathcal{M}$.

Encoding Disjunctive Transitions The translation of a disjunctive rule $\delta(q,Z) = (q_1,Z_1,d_1) \vee (q_2,Z_2,d_2)$ is a set of productions shown in Figure 5. The productions check that the current configuration allows to apply the rule, and update the configuration accordingly. Nothing in these productions requires more than c as a dimension (the other coordinates are either redundant or unused).

Encoding Conjunctive Transitions We exploit the extra indices for conjunctive rules $\delta(q,Z) = (q_1,Z_1,d_1) \wedge (q_2,Z_2,d_2)$, see Figure 6. The effect of the first set of productions is to update the left and right configurations with the new configurations for each of the two conjuncts. The next three sets ensure that the nondeterministic splitting between left and right was correct, and copy back the configuration in the left part into the right one and vice versa. Once this copy step is completed, and only then—which enforces a verification at the same time—, the last production set switches back to a normal nonterminal, thereby allowing the application of more rules.

Start Symbol and Covering Set Define $S = ([q_0, 1])$ as start symbol, while the initial vector encodes the initial tape contents $\gamma_0 = w\#^{p(|w|)-|w|}$ (where $\#$ denotes the blank tape symbol):

$$\bar{x}_0 = \sum_{Z \in \Gamma, i \leq p(|w|), j \in \{1,2\}} \begin{cases} \overline{\langle Z, i \rangle}_j & \text{if } \gamma_0(i) \neq Z \\ \langle Z, i \rangle_j & \text{otherwise.} \end{cases}$$

Our covering set C is the union of all pairs $([q_f, i], \overline{0})$ for q_f an accepting state of $\mathcal{M}$ and $i \leq p(|w|)$ a tape position.

B.1.2 Upper Bound

By Proposition 17, coverability and reachability can be reduced to language non emptiness. By Lemma 8, a k-bounded MLIG can be converted into an equivalent CFG in exponential time (and thus of exponential size). Emptiness in CFGs can be checked in polynomial time, and we have overall an exponential time algorithm for coverability and reachability in k-bounded MLIGs.