On the Computational Complexity of Dominance Links in Grammatical Formalisms
Sylvain Schmitz

To cite this version:
ON THE COMPUTATIONAL COMPLEXITY OF DOMINANCE
LINKS IN GRAMMATICAL FORMALISMS

SYLVAIN SCHMITZ

Abstract. Dominance links were introduced in grammars to model long dis-
tance scrambling phenomena, motivating the definition of multiset-valued lin-
ear 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 restric-
tions.

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 illus-
tration 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 be-	ween the nominative Peter and the accusative den Kühlschrank; [Rambow (1994b)] intro-
duced 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 en-
forced 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][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 (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.

Originally published in the Proceedings of the 48th Annual Meeting of the Association for
Computational Linguistics (ACL 2010), pages 514–524, 2010. This is a revised version from
February 4, 2014, that makes the difference between leaf and root coverability explicit and provides
updated references to the literature.
Figure 1. A vector of productions for the verb \textit{repariert} together with its two complements.

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 minimalist grammars without shortest move constraint (Salvati, 2011),
- 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., 2012; 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 \textsc{ExpTime}-complete for MLIGs, and that the emptiness and membership problems are \textsc{ExpTime}-complete for $k$-bounded MLIGs but \textsc{PTime}-complete in the $k$-ranked case. This also implies an \textsc{ExpTime} lower bound for emptiness and membership in minimalist 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.

\textbf{Notations.} In the following, $\Sigma$ denotes a finite alphabet, $\Sigma^*$ the set of finite sentences over $\Sigma$, and $\varepsilon$ 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 $\Sigma^*$. Let $\mathbb{N}^n$ denote the set of vectors of positive integers of dimension $n$. The $i$-th component of a vector $\mathbf{x}$ in $\mathbb{N}^n$ is $x(i)$, $\mathbf{0}$ denotes the null vector, $\mathbf{T}$ the vector with 1 values, and $\mathbf{e}_i$ the vector with 1 as its $i$-th component and 0 everywhere else. The ordering $\leq$ on $\mathbb{N}^n$ is the componentwise ordering: $\mathbf{x} \leq \mathbf{y}$ iff $x(i) \leq y(i)$ for all $0 < i \leq n$. The size of a vector refers to the size of its binary encoding: $|\mathbf{x}| = \sum_{i=1}^n 1 + \max(0, \lfloor \log_2 x(i) \rfloor)$.

We refer the reader unfamiliar with complexity classes and notions such as hardness or \textsc{LogSpace} reductions to classical textbooks (e.g. Papadimitriou, 1994).
**THE COMPLEXITY OF DOMINANCE LINKS**

$$S, (0, 0, 0)$$

$$S, (1, 1, 1)$$

$$b \ S, (1, 0, 1)$$

$$S, (2, 1, 2)$$

$$c \ S, (2, 1, 1)$$

$$a \ S, (1, 1, 1)$$

$$a \ S, (0, 1, 1)$$

$$b \ S, (0, 0, 1)$$

$$c \ S, (0, 0, 0)$$

$$\varepsilon$$

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

## 2. Multiset-Valued Linear Indexed Grammars

**Definition 1** (Rambow, 1994b). An $n$-dimensional multiset-valued linear indexed grammar (MLIG) is a tuple $G = \langle N, \Sigma, P, (S, x_0) \rangle$ where $N$ is a finite set of non-terminal symbols, $\Sigma$ a finite alphabet disjoint from $N$, $V = (N \times N^n) \cup \Sigma$ the vocabulary, $P$ a finite set of productions in $(N \times N^n) \times V^*$, and $(S, x_0) \in N \times N^n$ the start symbol. Productions are more easily written as

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

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

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

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

if $\delta$ and $\delta'$ are in $V^*$, a production of form $\langle \rangle$ appears in $P$, $\vec{x} \leq \vec{y}$, for each $1 \leq i \leq m$, $\vec{x}_i \leq \vec{y}_i$, and $\vec{y} - \vec{x} = \sum_{i=1}^{m} \vec{y}_i - \vec{x}_i$.

The language of a MLIG is the set of terminal strings derived from $(S, x_0)$, i.e.

$$L(G) = \{ w \in \Sigma^* \mid (S, x_0) \Rightarrow^* w \}$$

and we denote by $L(\text{MLIG})$ 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

$$(S, \vec{y}) \rightarrow \varepsilon \mid (S, \vec{1}), (S, \vec{x}_1) \rightarrow a (S, \vec{y}),$$

$$(S, \vec{x}_2) \rightarrow b (S, \vec{y}), (S, \vec{x}_3) \rightarrow c (S, \vec{y})$$

and start symbol $(S, \vec{0})$. 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):

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

The size $|G|$ of a MLIG $G$ is essentially the sum of the sizes of each of its productions of form $\langle \rangle$:

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

A MLIG is in \textit{extended two form} (ETF) if all its productions are of form
\begin{align*}
\text{terminal: } & (A, \overline{a}) \rightarrow a \text{ or } (A, \overline{a}) \rightarrow \varepsilon, \\
\text{nonterminal: } & (A, \overline{x}) \rightarrow (B_1, \overline{x}_1)(B_2, \overline{x}_2) \text{ or } (A, \overline{x}) \rightarrow (B_1, \overline{x}_1),
\end{align*}
with \(a \in \Sigma, A, B_1, B_2 \in \mathbb{N}, \overline{x}, \overline{x}_1, \overline{x}_2 \in \mathbb{N}^n\). Using standard constructions, any MLIG can be put into ETF in linear time or logarithmic space.

A MLIG is in \textit{restricted index normal form} (RINF) if the productions in \(P\) are of form \((A, 0) \rightarrow \alpha, (A, 0) \rightarrow (B, e_i), (A, e_i) \rightarrow (B, 0)\), with \(A, B \in \mathbb{N}, 0 \leq i \leq n, \) and \(\alpha \in (\Sigma \cup (\mathbb{N} \times \{0\}))^*\). 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:

\textbf{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.

2.2.1. 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:

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

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

2.2.2. 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 (Segaard et al., 2007; Kallmeyer and Parmentier, 2008; Chiang and Schellfer, 2008). The \textit{shortest move constraint} (SMC) introduced by Stabler to enforce a strong form of minimality also falls into this category of restrictions.

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

A MLIG is \textit{k-ranked} (noted \(k\rMLIG\)) if any derivation starting with \(\alpha_0 = (S, \overline{x}_0)\) is of rank \(k\). It is \textit{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:

\footnote{This restriction is slightly stronger than that of \textit{linearly restricted} derivations (Rambow 1994b), but still allows to capture UVG-dl lexicalization.}
Lemma 6. Any \(n\)-dimensional \(k\)-ranked MLIG \(G\) can be transformed into an equivalent CFG \(G'\) in time \(O(|G| \cdot (n + 1)^{k^2})\).

Proof. We assume \(G\) 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 \((A, \bar{x})\) in \(N' \subseteq N \times N^n\) with \(\sum_{i=1}^{n} \bar{y}(i) \leq k\). Finally, for each production \((A, \bar{x}) \rightarrow (B_1, \bar{x}_1)(B_2, \bar{x}_2)\) of \(P\), at most \((n + 1)^{k^2}\) choices are possible for productions \((A, \bar{y}) \rightarrow (B_1, \bar{y}_1)(B_2, \bar{y}_2)\) with \((A, \bar{y}), (B_1, \bar{y}_1), (B_2, \bar{y}_2)\) in \(N'\). □

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 \(\alpha_j\), i.e. for all \(\bar{x} \in N^n\) such that there exist \(0 \leq j \leq p\), \(\delta, \delta' \in V^*\) and \(A \in N\) with \(\alpha_j = \delta(A, \bar{x})\delta'\), and for all \(1 \leq i \leq n\), one has \(\bar{x}(i) \leq k\).

A MLIG is \(k\)-bounded (noted \(kb\)-MLIG) if any derivation starting with \(\alpha_0 = (S, \bar{x}_0)\) is \(k\)-bounded. It is \(b\)-bounded if there exists \(k\) such that it is \(k\)-bounded.

The SMC in minimalist grammars translates exactly into \(1\)-boundedness of the corresponding MLIGs [Salvati, 2011].

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 \(G\) can be transformed into an equivalent CFG \(G'\) in time \(O(|G| \cdot (k + 1)^{n^2})\).

Proof. We assume \(G\) 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 \((A, \bar{x})\) in \(N' = N \times \{0, \ldots, k\}^n\). Finally, for each production \((A, \bar{x}) \rightarrow (B_1, \bar{x}_1)(B_2, \bar{x}_2)\) of \(P\), each nonterminal \((A, \bar{y})\) of \(N'\) with \(\bar{x} \leq \bar{y}\), and each index \(0 < i \leq n\), there are at most \(k + 1\) ways to split \(\bar{y}(i) - \bar{x}(i) \leq k\) into \(\bar{y}_1(i) + \bar{y}_2(i)\) and span a production \((A, \bar{y}) \rightarrow (B_1, \bar{x}_1 + \bar{y}_1)(B_2, \bar{x}_2 + \bar{y}_2)\) of \(P'\). Overall, each production is mapped to at most \((k + 1)n^2\) context-free productions. □

One can check that the grammar of Example 2 is not \(b\)-bounded (to see this, repeatedly apply production \((S, \bar{0}) \rightarrow (S, \bar{1})\), 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]. \(L(\text{MLIG})\) is a substitution closed full abstract family of languages.

Proposition 10 [Rambow, 1994b]. \(L(\text{MLIG}_I)\) 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 [MacLaughlan and Rambow, 2002]. This includes the copy language

\(L_{\text{copy}} = \{ww \mid w \in \{a, b\}^*\}\),

which does not seem to be generated by any MLIG:

Conjecture 11 [Rambow, 1994b]. \(L_{\text{copy}}\) is not in \(L(\text{MLIG})\).
Corollary 12. \( L(\text{k-MLIG}) = L(\text{kb-MLIG}) = L(\text{kb-MLIG}_l) \) 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).

3.1. Petri Nets

Definition 13 (Petri, 1962). A marked Petri net is a tuple \( N = (S, T, f, m_0) \) where \( S \) and \( T \) are disjoint finite sets of places and transitions, \( f \) a flow function from \((S \times T) \cup (T \times S)\) to \( \mathbb{N} \), and \( m_0 \) an initial marking in \( \mathbb{N}^S \). A transition \( t \in T \) can be fired in a marking \( m \) in \( \mathbb{N}^S \) if \( f(p, t) \geq m(p) \) for all \( p \in S \), and reaches a new marking \( m' \) defined by \( m'(p) = m(p) - f(p, t) + f(t, p) \) for all \( p \in S \), written \( m \xrightarrow{t} m' \). Another view is that place \( p \) holds \( m(p) \) tokens, of which are first removed when firing \( t \), and then \( f(t, p) \) added back. Firings are extended to sequences \( \sigma \in T^* \) by \( m \xrightarrow{\varepsilon} m \), and \( m \xrightarrow{\sigma t} m' \) if there exists \( m'' \) with \( m \xrightarrow{\sigma} m'' \xrightarrow{t} m' \).

A labeled Petri net with reachability acceptance is endowed with a labeling homomorphism \( \varphi : T^* \rightarrow \Sigma^* \) and a finite acceptance set \( F \subseteq \mathbb{N}^S \), defining the language \( L(N, \varphi, F) = \{ \varphi(\sigma) \in \Sigma^* \mid \exists m \in F, m_0 \xrightarrow{\sigma} m \} \).

Labeled Petri nets (with acceptance set \( \{ \varnothing \} \)) are notational variants of right linear MLIGs, defined as having production in \((N \times N^0) \times (\Sigma^* \cup (\Sigma^* \cdot (N \times N^0)))\). 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 \( (S, e_3) \rightarrow c(S, \nothing) \) is represented by the rightmost, \( c \)-labeled transition, with \( f(S, t) = f(e_3, t) = f(t, S) = 1 \) and \( f(e_1, t) = f(e_2, t) = f(t, e_1) = f(t, e_2) = f(t, e_3) = 0 \).

\(^2\)Petri nets are also equivalent to vector addition systems (Karp and Miller, 1969) VAS) and vector addition systems with states (Hopcroft and Pansiot, 1979) VASS).
3.1.1. 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).

3.1.2. 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:

\[
\begin{align*}
(S, e_2) &\rightarrow (S, e_1), \\
(A, e_1) &\rightarrow (A, 2e_2), \\
(B, e_1) &\rightarrow b (B, e_2) \mid b,
\end{align*}
\]

and \((S, e_2)\) as start symbol, that generates the non semilinear language

\[L_{nsm} = \{a^nb^m \mid 0 \leq n, 0 < m \leq 2n\}.\]

**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

3.2.1. **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 \(G = (N, \Sigma, W, S)\) where \(N\) and \(\Sigma\) 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 \(P\) 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 \(\Sigma^*\) 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 \(p_1p_2 \cdots p_p\) is a permutation of a member of \(\Sigma^*\) and the dominance relations of \(W\) hold in the associated derivation tree. The language \(L(G)\) of an UVG-dl \(G\) is the set of sentences \(w\) with some terminal derivation. We write \(L(UVG-dl)\) 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 \(S \rightarrow S\) in the second vector has to spawn eventually one

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

occurrence of each $S \rightarrow aS$, $S \rightarrow bS$, and $S \rightarrow cS$, which corresponds exactly to the MLIG of Example 2.

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’s proof of the inclusion $L(\text{MLIG}) \subseteq L(\text{UVG-dl})$ 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 $(A, 0) \rightarrow \alpha$ with $A$ in $N$ and $\alpha$ in $(\Sigma \cup (N \times \{0\}))^*$ form singleton vectors, and productions of form $(A, 0) \rightarrow (B, e_i)$ with $A$, $B$ in $N$ and $0 < i \leq n$ need to be paired with a production of form $(C, e_i) \rightarrow (D, 0)$ 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 $L(\text{UVG-dl}_l) \subseteq L(\text{MLIG}_l)$.

3.2.2. 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 (Candido 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), $k$-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 \( \langle N, \Sigma, P, (S, \alpha_0) \rangle \):

- **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.
- **root coverability**: given \( \langle \mathcal{G}, F \rangle \), \( \mathcal{G} \) \( \varepsilon \)-free in ETF and \( F \) a finite subset of \( N \times \mathbb{N}^n \), does there exist \( \bar{y}_0 \geq \bar{x}_0 \) and \( \alpha = (A_1, \bar{y}_1) \cdots (A_m, \bar{y}_m) \) in \( F^* \) such that \( (S, \bar{y}_0) \Rightarrow^* \alpha \)?
- **leaf coverability**: given \( \langle \mathcal{G}, F \rangle \), \( \mathcal{G} \) \( \varepsilon \)-free in ETF and \( F \) a finite subset of \( N \times \mathbb{N}^n \), does there exist \( \alpha = (A_1, \bar{y}_1) \cdots (A_m, \bar{y}_m) \) in \( (N \times \mathbb{N}^n)^* \) such that \( (S, \bar{y}_0) \Rightarrow^* \alpha \) and for each \( 0 < j \leq m \) there exists \( (A_j, \bar{x}_j) \) with \( \bar{x}_j \leq \bar{y}_j \)?
- **reachability**: given \( \langle \mathcal{G}, F \rangle \), \( \mathcal{G} \) \( \varepsilon \)-free in ETF and \( F \) a finite subset of \( N \times \mathbb{N}^n \), does there exist \( \alpha = (A_1, \bar{y}_1) \cdots (A_m, \bar{y}_m) \) in \( F^* \) such that \( (S, \bar{x}_0) \Rightarrow^* \alpha \)?
- **non emptiness**: given \( \langle \mathcal{G} \rangle \), is \( L(\mathcal{G}) \) non empty?
- **(uniform) membership**: given \( \langle \mathcal{G}, w \rangle \), \( w \in \Sigma^* \), does \( w \) belong to \( L(\mathcal{G}) \)?

Boundness 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 \textsc{LogSpace} reductions between decision problems; we have:

**Proposition 17.**

(1) root coverability \( \rightarrow_{\log} \) reachability
(2) leaf coverability \( \rightarrow_{\log} \) reachability
(3) reachability \( \leftrightarrow_{\log} \) non emptiness
(4) \( \leftrightarrow_{\log} \) membership

**Proof sketch.** For (1), construct a reachability instance \( \langle \mathcal{G}', F \rangle \) from a root coverability instance \( \langle \mathcal{G}, F \rangle \) by adding to \( \mathcal{G} \) a fresh nonterminal \( S' \) and the productions
\[
\{(S', \overline{0}) \rightarrow (S', \overline{1}) \mid 0 < i \leq n\} \\
\cup \{(S', \overline{0}) \rightarrow (S, \overline{0})\}.
\]

For (2), construct a reachability instance \( \langle \mathcal{G}', \{(E, \overline{0})\} \rangle \) from a leaf coverability instance \( \langle \mathcal{G}, F \rangle \) by adding to \( \mathcal{G} \) a fresh nonterminal \( E \) and the productions
\[
\{(A, \overline{x}) \rightarrow (E, \overline{0}) \mid (A, \overline{x}) \in F\} \\
\cup \{(E, \overline{0}) \rightarrow (E, \overline{0}) \mid 0 < i \leq n\}.
\]

For (3), 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 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. \( F = \{(A, \overline{x}) \mid (A, \overline{x}) \rightarrow a \in P, a \in \Sigma \cup \{\varepsilon\}\} \), and then remove all terminal productions in order to obtain a reachability instance \( \langle \mathcal{G}', F \rangle \).
Table 1. Summary of complexity results.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Petri net $k$-Boundedness</td>
<td>PSpace-c. (Jones et al., 1977)</td>
</tr>
<tr>
<td>Petri net Boundedness</td>
<td>ExpSpace-c. (Lipton, 1976; Rackoff, 1978)</td>
</tr>
<tr>
<td>Petri net {Emptiness, Membership}</td>
<td>ExpSpace-hard (Lipton, 1976), $\Delta_1$-easy (Mayr, 1981; Kosaraju, 1982)</td>
</tr>
<tr>
<td>{MLIG, MLIG}_ℓ $k$-Boundedness</td>
<td>ExpTime-c. (Corollary 21)</td>
</tr>
<tr>
<td>{MLIG, MLIG}_ℓ Boundedness</td>
<td>2ExpTime-c. (Demri et al., 2012)</td>
</tr>
<tr>
<td>{MLIG, MLIG}_ℓ Emptiness</td>
<td>Tower-hard (Lazić and Schmitz, 2014), $\Sigma_1^n$-easy</td>
</tr>
<tr>
<td>MLIG Membership</td>
<td>ExpTime-c. (Theorem 20)</td>
</tr>
<tr>
<td>{kb-MLIG, kb-MLIG}_ℓ Emptiness</td>
<td>ExpTime-c. (Theorem 20)</td>
</tr>
<tr>
<td>kb-MLIG Membership</td>
<td>NPTIME-c. (Koller and Rambow, 2007)</td>
</tr>
<tr>
<td>{MLIG}_ℓ, kb-MLIG}_ℓ Membership</td>
<td>PTIME-c. (Jones and Laaser, 1976) and Lemma 6</td>
</tr>
</tbody>
</table>

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

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. (2012) proved tight complexity bounds for these problems, extending earlier results by Rackoff (1978) and Lipton (1976) for Petri nets.

Theorem 18 (Demri et al., 2012). Root coverability and boundedness for MLIGs are 2ExpTime-complete.

More recently, Lazić and Schmitz (2014) proved a Tower lower bound on leaf coverability: here the tower function is defined by tower(0) = 0 and tower($n + 1$) = $2^{\text{tower}(n)}$, and

$$\text{Tower} = \bigcup_{e \in \text{FElem}} \text{DTIME}(\text{tower}(e(n)))$$

is the class of problems that can be solved by a deterministic Turing machine operating in time tower of some elementary function $e$ of the input. Complete problems for Tower are understood relative to elementary many-one reductions.

Theorem 19 (Lazić and Schmitz, 2014). Leaf coverability for MLIGs is Tower-complete.

Regarding reachability, emptiness, and membership, decidability is still open. A 2ExpSpace lower bound was found by Lazić (2010), and improved to Tower by Lazić and Schmitz (2014). 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. \textit{k}-Bounded and \textit{k}-Ranked Cases

First observe that, in the \textit{k}-bounded and \textit{k}-ranked cases, root coverability, leaf coverability, and reachability are all interreducible (using complement counters).

Since \textit{k}-bounded MLIGs can be converted into CFGs (Lemma 8), emptiness and membership problems are decidable, albeit at the expense of an exponential blowup. We know from the Petri net literature that coverability and reachability problems are $\text{PSpace}$-complete for \textit{k}-bounded right linear MLIGs (Jones et al., 1977) by a reduction from linear bounded automaton (LBA) membership. We obtain the following for \textit{k}-bounded MLIGs, using a similar reduction from membership in polynomially space bounded alternating Turing machines (Chandra et al., 1981):

\textbf{Theorem 20.} Reachability for \textit{k}-bounded MLIGs is $\text{ExpTime}$-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 leaf coverability problem for 1-bounded MLIGs. The upper bound is a direct application of Lemma 8, reachability being reducible to the emptiness problem for a CFG of exponential size. Theorem 20 also shows the $\text{ExpTime}$-hardness of emptiness and membership in minimalist grammars with SMC.

\textbf{Corollary 21.} Let $k \geq 1$; \textit{k}-boundedness for MLIGs is $\text{ExpTime}$-complete.

\textit{Proof.} For the lower bound, consider an instance $\langle G, F \rangle$ of leaf coverability for a 1-bounded MLIG $G$, which is $\text{ExpTime}$-hard according to Theorem 20. Add to the MLIG $G$ a fresh nonterminal $E$ and the productions

$$\{(A, \overline{x}) \rightarrow (E, \overline{x}) \mid (A, \overline{x}) \in F\}$$

$$\cup \{(E, \overline{0}) \rightarrow (E, \overline{e_i}) \mid 0 < i \leq n\},$$

which make it non-\textit{k}-bounded iff the coverability instance was positive.

For the upper bound, apply Lemma 8 with $k' = k + 1$ to construct an $O(|G| \cdot 2^{n^2 \log_2 (k' + 1)})$-sized CFG, reduce it in polynomial time, and check whether a non-terminal $(A, \overline{x})$ with $\overline{x}(i) = k'$ 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. \hfill \Box

Corollary 21 also implies the $\text{ExpTime}$-completeness of $\textit{k}$-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 $G$, then $\textit{k}$-rankedness becomes polynomial by Lemma 8.

Observe finally that $\textit{k}$-rankedness provides the only tractable class of MLIGs for uniform membership, using again Lemma 8 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 $\text{PTime}$-complete (Jones and Laaser, 1976).

4.4. Lexicalized Case

Unlike the high complexity lower bounds of the previous two sections, $\text{NPTime}$-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 (with closed leaves) configurability problem \cite{althaus2003}, as it allows to derive \textsc{Nptime}-hardness even in highly restricted grammars.

**Theorem 22** \cite{koller2007}. Uniform membership of \( \langle G, w \rangle \) for \( G \) a 1-bounded, lexicalized, UVG-dl with finite language is \textsc{Nptime}-hard, even for \(|w| = 1\).

\textit{Proof sketch.} Set \( S \) as start symbol and add a production \( S \rightarrow aA \) to the sole vector of the grammar \( G \) constructed by \cite{koller2007} from a normal dominance graph, with dominance links to all the other productions. Then \( G \) becomes strict, lexicalized, with finite language \( \{a\} \) or \( \emptyset \), and 1-bounded, such that \( a \) belongs to \( L(G) \) iff the normal dominance graph is configurable. \( \square \)

The fact that uniform membership is in \textsc{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 \( L_{\text{pal}} \) of even 2-letters palindromes was a Petri net language, was indeed solved using the decidability of reachability in Petri nets \cite{jantzen1979}, and shown to be strongly related to the latter \cite{lambert1992}.

A conclusion with a more immediate linguistic value is that MLIGs and UVG-dls hardly qualify as formalisms for \textit{mildly context-sensitive languages}, claimed by \cite{joshi1985} to be adequate for modeling natural languages, and “roughly” defined as the extensions of context-free languages that display

1. support for \textit{limited cross-serial dependencies}: seems doubtful, see Conjecture \[11\]
2. \textit{constant growth}, a requisite nowadays replaced by \textit{semilinearity}: does not hold, as seen with Proposition \[14\] and
3. \textit{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.

Acknowledgements

Thanks to Pierre Chambart, Stéphane Demri, and Alain Finkel for helpful discussions, and to Sylvain Salvati for pointing out the relation with minimalist grammars.

REFERENCES


APPENDIX 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 24), and then by translating this grammar in OF into a grammar in RINF (Lemma 25). This construction is akin to the normalization presented by Dufourd and Finkel (1999) for reset Petri nets.

**Definition 23.** A MLIG is in ordinary form if, for any production of form \( P \) in \( P \), for any vector \( y \in \{x_0\} \cup \{x_i\} \cup \{x_j \mid 1 \leq j \leq m\} \) that appears in the start symbol or in this production, and for any index \( 0 < i \leq n \), \( y(i) \leq 1 \).

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

**Proof.** Let us fix a MLIG \( G = \langle N, \Sigma, P, (S,x_0) \rangle \). We first define the maximal vector value of \( G \) as the minimum integer \( \text{max}_G \) such that, for any production of form \( \langle e \rangle \) in \( P \), for any vector \( y \in \{x_0\} \cup \{x\} \cup \{x_j \mid 1 \leq j \leq m\} \) that appears in the start symbol or in this production, and for any index \( 0 < i \leq n \), \( y(i) \leq \text{max}_G \). Thus a MLIG in OF is one where \( \text{max}_G \leq 1 \).

Let \( n' = \text{max}_G \) (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 \( G \) is associated with \( n' + 1 \) indices in the constructed grammar \( G' \). The index \( (i,j) \) denotes the \( j \)th 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

\[
P_{A,i} = \{(A, e_{i,0}) \rightarrow (A_{i,1}, \overline{0})\} \\
\cup \{(A_{i,j}, e_{i,j}) \rightarrow (A_{i,j+1}, \overline{0}) \mid 0 < j \leq n'\} \\
\cup \{(A_{i,j}, \overline{0}) \rightarrow (A, e_{i,j}) \mid 0 < j \leq n'\}
\]

\[
P'_{A,i} = \{(A, e_{i,j}) \rightarrow (A'_{i,j}, \overline{0}) \mid 0 < j \leq n'\} \\
\cup \{(A'_{i,j}, \overline{0}) \rightarrow (A'_{i,j-1}, e_{i,j-1}) \mid 1 < j \leq n'\} \\
\cup \{(A'_{i,n'}, \overline{0}) \rightarrow (A, e_{i,n'})\}
\]

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 \( i \)th index, i.e. that

\[
y(i,0) + \sum_{j=1}^{n'} y(i,j)2^{j-1} = y'(i,0) + \sum_{j=1}^{n'} y'(i,j)2^{j-1}
\]

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

**Claim 24.1.** If \( (A,y) \Rightarrow^P (A,y') \), for some \( p \geq 0 \) and using only productions from \( P_{A,i} \cup P'_{A,i} \), then \( \text{(6)} \) 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

\[
(A, \varphi) \Rightarrow^{p-1} (A, \varphi_{p-1}) \Rightarrow (A, \varphi')
\]

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

1. \((A, \varphi) \Rightarrow^{p-2} (A, \varphi_{p-2}) \Rightarrow (A_{i,j}, \varphi_{p-1})\), which enforces \( j = 1 \), and then

\[
\varphi_{p-2}(i, 0) = \varphi_{p-1}(i, 0) + 1 = \varphi'(i, 0) + 1
\]

and

\[
\varphi_{p-2}(i, 1) + 1 = \varphi_{p-1}(i, 1) + 1 = \varphi'(i, 1),
\]

thus \([1]\) holds between \( \varphi' \) and \( \varphi_{p-2} \), and using the induction hypothesis on derivation \((A, \varphi) \Rightarrow^{p-2} (A, \varphi_{p-2})\), it also holds for the entire derivation.

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

\[
\varphi_{p-1}(i, j - 1) + 2 = \varphi''(i, j - 1)
\]

\[
= \varphi'(i, j - 1) + 2
\]

\[
\varphi_{p-1}(i, j) + 1 = \varphi''(i, j) + 1 = \varphi'(i, j),
\]

and therefore \([2]\) holds between \( \varphi' \) and \( \varphi'' \). Applying the induction hypothesis to \((A, \varphi) \Rightarrow^{p-2} (A_{i,j-1}, \varphi_{p-2}) \Rightarrow (A, \varphi'')\) yields the claim.

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

\[
\mathbf{x}(i) = \sum_{j=1}^{n'} b\mathbf{x}(i, j)2^{j-1},
\]

\( b\mathbf{x}(i, 0) = 0 \), and

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

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

\[
(A, b\mathbf{x}) \Rightarrow u_0(B_1, b\mathbf{x}_1)u_1 \cdots u_m(B_m, b\mathbf{x}_m)u_{m+1}
\]

for each production of form \([3]\) 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 \( 0 < i \leq n \).

Claim 24.2. The \((n(n'+1))-\)dimensional MLIG \( G' = \langle N', \Sigma, P', (S, b\mathbf{x}_0) \rangle \) is in OF and equivalent to \( G \).

The fact that \( G' \) is in OF is immediate by definition of the binary encoding \( b\mathbf{x} \) and of the productions of \( P' \). The equivalence of \( G \) and \( G' \) stems from Claim 24.1 and the properties of \( b\mathbf{x} \).

We can conclude by noting that, indeed, \( G' \) can be constructed from \( G \) in logarithmic space. \qed
\{\langle q, i \rangle, \langle Z, i \rangle_1 + \langle Z, i \rangle_2 + \langle Z_j, i \rangle_1 + \langle Z_j, i \rangle_2 \}
\{(q, i + d_1), \langle Z, i \rangle_1 + \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 \(\delta(q, Z) = (q_1, Z_1, d_1) \lor (q_2, Z_2, d_2)\).

**Lemma 25.** 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. \(\square\)

**APPENDIX B. COMPLEMENTS TO SECTION 4**

This section contains the proof of the following result:

**Theorem 20.** Reachability for \(k\)-bounded MLIGs is ExpTime-complete, even for fixed \(k \geq 1\).

**B.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 \(\Sigma^*\), 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 \(\Gamma\) to be \((q_1, Z_1, d_1) \lor (q_2, Z_2, d_2)\) with \(q_1, q_2\) in \(Q\), \(Z_1 \neq Z\) and \(Z_2 \neq Z\) in \(\Gamma\), \(d_1, d_2\) in \([-1, +1]\) (standing for a move to the left or to the right), and \(op\) in \(\{\lor, \land\}\) (standing for disjunction or conjunction).

**B.1.1. 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 \(\delta(q, Z) = (q_1, Z_1, d_1) \land (q_2, Z_2, d_2)\), which cannot be directly simulated by MLIG derivations of form \((A, \vec{y}) \Rightarrow (B_1, \vec{y}_1)(B_2, \vec{y}_2)\) with \((A, \vec{y})\) encoding the configuration matched by \((q, Z)\), and each \((B_j, \vec{y}_j)\) encoding the new configuration corresponding to the \((q_j, Z_j, d_j)\) action. Vector values from \(\vec{y}\), encoding the current tape configuration, are scattered nondeterministically between \(\vec{y}_1\) and \(\vec{y}_2\). The solution is to construct a 1-bounded MLIG with enough redundancy to recover “clean” tape configurations after the simulation of a conjunctive transition.

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 \(\langle Z, i \rangle_d\)).

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 \(N \times \mathbb{N}^8\) represents the current configuration of \(\mathcal{M}\).
be checked in polynomial time, and we have overall an exponential time algorithm
CFG in exponential time (and thus of exponential size). Emptiness in CFGs can
emptiness. By Lemma 8, a

By Proposition 17, coverability and reachability can be reduced to language non

B.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.

LSV, ENS CACHAN & CNRS, FRANCE
E-mail address: schmitz@lsv.ens-cachan.fr