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; Rambow et al., 2001; Candito and Kahane, 1998; Kallmeyer, 2001; Guillaume and Perrier, 2010) and TAG extensions (Becker et al., 1991; Rambow, 1994a).

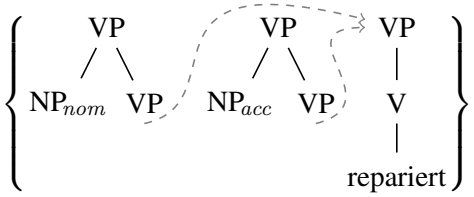

Figure 1: A vector of productions for the verb *repariert* together with its two complements.

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 minimalist grammars without

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 514–524,
Uppsala, Sweden, 11-16 July 2010.

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 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.

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 $\overline{\mathsf{x}}$ in $\mathbb{N}^n$ is $\overline{\mathsf{x}}(i)$, $\overline{0}$ denotes the null vector, $\overline{1}$ the vector with 1 values, and $\overline{\mathsf{e}}_i$ the vector with 1 as its i-th component and 0 everywhere else. The ordering $\leq$ on $\mathbb{N}^n$ is the componentwise ordering: $\overline{\mathsf{x}} \leq \overline{\mathsf{y}}$ iff $\overline{\mathsf{x}}(i) \leq \overline{\mathsf{y}}(i)$ for all $0 < i \leq n$. The size of a vector refers to the size of its binary encoding: $|\overline{\mathsf{x}}| = \sum_{i=1}^{n} 1 + \max(0, \lfloor \log_2 \overline{\mathsf{x}}(i) \rfloor)$.

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 $\mathcal{G} = \langle N, \Sigma, P, (S, \overline{\mathsf{x}}_0) \rangle$ 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 $(S, \overline{\mathsf{x}}_0) \in N \times \mathbb{N}^n$ the start symbol. Productions are more easily written as

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

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

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

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

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

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

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

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

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

and start symbol $(S, \overline{0})$. 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* $|\mathcal{G}|$ of a MLIG $\mathcal{G}$ is essentially the sum of the sizes of each of its productions of form $(\star)$:

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

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

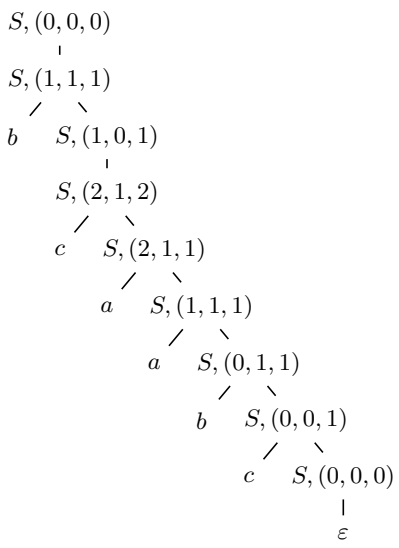

Figure 2: A derivation for $bcaabc$ in the grammar of Example 2.

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

with a in Σ, A, B_1, B_2 in N, and $\overline{\mathsf{x}}$, $\overline{\mathsf{x}}_1$, $\overline{\mathsf{x}}_2$ 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 $(A,\overline{0}) \rightarrow \alpha$, $(A,\overline{0}) \rightarrow (B,\overline{\mathsf{e}}_i)$, or $(A,\overline{\mathsf{e}}_i) \rightarrow (B,\overline{0})$, with A, B in N, $0 < i \leq n$, 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.

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 $(S,\overline{\mathsf{x}}_0)$ is c-lexicalized, and we denote by $\mathcal{L}(\mathrm{MLIG}_\ell)$ 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 (1997) 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 $\overline{\mathsf{x}}$ in $\mathbb{N}^n$ such that there exist $0 \leq j \leq p$, δ, δ' in V^* and A in N with $\alpha_j = \delta(A,\overline{\mathsf{x}})\delta'$, one has $\sum_{i=1}^n \overline{\mathsf{x}}(i) \leq k$.

A MLIG is *k-ranked* (noted kr-MLIG) if any derivation starting with $\alpha_0 = (S,\overline{\mathsf{x}}_0)$ 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 $(A,\overline{\mathsf{y}})$ in $N' = N \times \mathbb{N}^n$ with $\sum_{i=1}^n \overline{\mathsf{y}}(i) \leq k$. Finally, for each production $(A,\overline{\mathsf{x}}) \rightarrow (B_1,\overline{\mathsf{x}}_1)(B_2,\overline{\mathsf{x}}_2)$ of P, at most $(n+1)^{k^3}$ choices are possible for productions $(A,\overline{\mathsf{y}}) \rightarrow (B_1,\overline{\mathsf{y}}_1)(B_2,\overline{\mathsf{y}}_2)$ with $(A,\overline{\mathsf{y}})$, $(B_1,\overline{\mathsf{y}}_1)$, and $(B_2,\overline{\mathsf{y}}_2)$ in N'. □

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

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

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

A MLIG is *k-bounded* (noted kb-MLIG) if any derivation starting with $\alpha_0 = (S, \overline{\mathsf{x}}_0)$ is k-bounded. It is *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, 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 $(A, \overline{\mathsf{y}})$ in $N' = N \times \{0, \ldots, k\}^n$. Finally, for each production $(A, \overline{\mathsf{x}}) \to (B_1, \overline{\mathsf{x}}_1)(B_2, \overline{\mathsf{x}}_2)$ of P, each nonterminal $(A, \overline{\mathsf{y}})$ of N' with $\overline{\mathsf{x}} \leq \overline{\mathsf{y}}$, and each index $0 < i \leq n$, there are at most $k+1$ ways to split $(\overline{\mathsf{y}}(i) - \overline{\mathsf{x}}(i)) \leq k$ into $\overline{\mathsf{y}}_1(i) + \overline{\mathsf{y}}_2(i)$ and span a production $(A, \overline{\mathsf{y}}) \to (B_1, \overline{\mathsf{x}}_1 + \overline{\mathsf{y}}_1)(B_2, \overline{\mathsf{x}}_2 + \overline{\mathsf{y}}_2)$ 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 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

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

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. *$\mathcal{L}(k\text{r-MLIG}) = \mathcal{L}(k\text{b-MLIG}) = \mathcal{L}(k\text{b-MLIG}_\ell)$ 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).

3.1 Petri Nets

Definition 13 (Petri, 1962)**.** A marked *Petri net*[2] is a tuple $\mathcal{N} = \langle \mathcal{S}, T, f, \overline{\mathsf{m}}_0 \rangle$ 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 $\overline{\mathsf{m}}_0$ an initial marking in $\mathbb{N}^{\mathcal{S}}$. A transition $t \in T$ can be fired in a *marking* $\overline{\mathsf{m}}$ in $\mathbb{N}^{\mathcal{S}}$ if $f(p,t) \geq \overline{\mathsf{m}}(p)$ for all $p \in \mathcal{S}$, and reaches a new marking $\overline{\mathsf{m}}'$ defined by $\overline{\mathsf{m}}'(p) = \overline{\mathsf{m}}(p) - f(p,t) + f(t,p)$ for all $p \in \mathcal{S}$, written $\overline{\mathsf{m}}\ [t\rangle\ \overline{\mathsf{m}}'$. Another view is that place p holds $\overline{\mathsf{m}}(p)$ *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 $\overline{\mathsf{m}}\ [\varepsilon\rangle\ \overline{\mathsf{m}}$, and $\overline{\mathsf{m}}\ [\sigma t\rangle\ \overline{\mathsf{m}}'$ if there exists $\overline{\mathsf{m}}''$ with $\overline{\mathsf{m}}\ [\sigma\rangle\ \overline{\mathsf{m}}''\ [t\rangle\ \overline{\mathsf{m}}'$.

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)

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

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

[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).

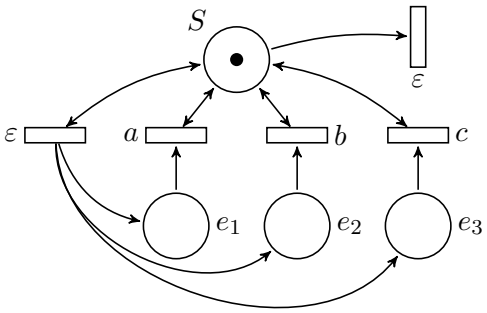

Figure 3: The labeled Petri net corresponding to the right linear MLIG of Example 2.

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 $(S,\overline{\mathsf{e}}_3) \to c\,(S,\overline{0})$ 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).

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]

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

[3] Adding terminal symbols c in each production would result in a lexicalized grammar, still with a non semilinear language.

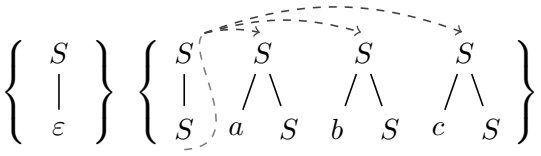

Figure 4: An UVG-dl for L_{mix}.

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

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

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 $S \overset{p_1}{\Rightarrow} \alpha_1 \overset{p_2}{\Rightarrow} \alpha_2 \cdots \alpha_{p-1} \overset{p_p}{\Rightarrow} w$ 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.

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 $S \to S$ 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 $\mathcal{L}(\text{MLIG}) \subseteq \mathcal{L}(\text{UVG-dl})$ 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 $(A,\overline{0}) \to \alpha$ with A in N and α in $(\Sigma \cup (N \times \{\overline{0}\}))^*$ form singleton vectors, and productions of form $(A,\overline{0}) \to (B,\overline{\mathsf{e}}_i)$ with A, B in N and $0 < i \leq n$ need to be paired with a production of form $(C,\overline{\mathsf{e}}_i) \to (D,\overline{0})$ 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 $\mathcal{L}(\text{UVG-dl}_\ell) \subseteq \mathcal{L}(\text{MLIG}_\ell)$.

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 fb-TAGs (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), 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 $\langle N, \Sigma, P, (S, \overline{\mathsf{x}}_0)\rangle$:

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 $\alpha = (A_1, \overline{\mathsf{y}}_1) \cdots (A_m, \overline{\mathsf{y}}_m)$ in $(N \times \mathbb{N}^n)^*$ such that $(S, \overline{\mathsf{x}}_0) \Rightarrow^* \alpha$ and for each $0 < j \leq m$ there exists $(A_j, \overline{\mathsf{x}}_j)$ in F with $\overline{\mathsf{x}}_j \leq \overline{\mathsf{y}}_j$?

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 $\alpha = (A_1, \overline{\mathsf{y}}_1) \cdots (A_m, \overline{\mathsf{y}}_m)$ in F^* such that $(S, \overline{\mathsf{x}}_0) \Rightarrow^* \alpha$?

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.

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

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

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

For (2), from a reachability instance $\langle \mathcal{G}, F\rangle$, remove all terminal productions from $\mathcal{G}$ and add instead the productions $\{(A, \overline{\mathsf{x}}) \to \varepsilon \mid (A, \overline{\mathsf{x}}) \in F\}$; 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. $F = \{(A, \overline{\mathsf{x}}) \mid (A, \overline{\mathsf{x}}) \to a \in P, a \in \Sigma \cup \{\varepsilon\}\}$, 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 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

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

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

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.

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 $(A, \bar{\mathrm{x}})$ with $\bar{\mathrm{x}}(i) = k'$ 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. □

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 $S \to aA$ 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. □

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.

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

Ernst Althaus, Denys Duchier, Alexander Koller, Kurt Mehlhorn, Joachim Niehren, and Sven Thiel. 2003. An efficient graph algorithm for dominance constraints. *Journal of Algorithms*, 48(1):194–219.

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

G. Edward Barton. 1985. The computational difficulty of ID/LP parsing. In *ACL'85*, pages 76–81. ACL Press.

Tilman Becker, Aravind K. Joshi, and Owen Rambow. 1991. Long-distance scrambling and tree adjoining grammars. In *EACL'91*, pages 21–26. ACL Press.

Mikołaj Bojańczyk, Anca Muscholl, Thomas Schwentick, and Luc Segoufin. 2009. Two-variable logic on data trees and XML reasoning. *Journal of the ACM*, 56(3):1–48.

Marie-Hélène Candito and Sylvain Kahane. 1998. Defining DTG derivations to get semantic graphs. In *TAG+4*, pages 25–28.

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

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

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the ACM*, 28(1):114–133.

David Chiang and Tatjana Scheffler. 2008. Flexible composition and delayed tree-locality. In *TAG+9*.

Armin B. Cremers and Otto Mayer. 1974. On vector languages. *Journal of Computer and System Sciences*, 8(2):158–166.

Philippe de Groote, Bruno Guillaume, and Sylvain Salvati. 2004. Vector addition tree automata. In *LICS'04*, pages 64–73. IEEE Computer Society.

Stéphane Demri, Marcin Jurdziński, Oded Lachish, and Ranko Lazić. 2009. The covering and boundedness problems for branching vector addition systems. In Ravi Kannan and K. Narayan Kumar, editors, *FSTTCS'09*, volume 4 of *Leibniz International Proceedings in Informatics*, pages 181–192. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

Catherine Dufourd and Alain Finkel. 1999. A polynomial λ-bisimilar normalization for reset Petri nets. *Theoretical Computer Science*, 222(1–2):187–194.

Javier Esparza. 1995. Petri nets, commutative context-free grammars, and basic parallel processes. In Horst Reichel, editor, *FCT'95*, volume 965 of *Lecture Notes in Computer Science*, pages 221–232. Springer.

Bruno Guillaume and Guy Perrier. 2010. Interaction grammars. *Research on Language and Computation*. To appear.

Serge Haddad and Denis Poitrenaud. 2007. Recursive Petri nets. *Acta Informatica*, 44(7–8):463–508.

John Hopcroft and Jean-Jacques Pansiot. 1979. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8(2):135–159.

Dung T. Huynh. 1983. Commutative grammars: the complexity of uniform word problems. *Information and Control*, 57(1):21–39.

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

Neil D. Jones and William T. Laaser. 1976. Complete problems for deterministic polynomial time. *Theoretical Computer Science*, 3(1):105–117.

Neil D. Jones, Lawrence H. Landweber, and Y. Edmund Lien. 1977. Complexity of some problems in Petri nets. *Theoretical Computer Science*, 4(3):277–299.

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

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

Laura Kallmeyer and Yannick Parmentier. 2008. On the relation between multicomponent tree adjoining grammars with tree tuples (TT-MCTAG) and range concatenation grammars (RCG). In Carlos Martín-Vide, Friedrich Otto, and Henning Fernau, editors, *LATA'08*, volume 5196 of *Lecture Notes in Computer Science*, pages 263–274. Springer.

Laura Kallmeyer. 2001. Local tree description grammars. *Grammars*, 4(2):85–137.

Richard M. Karp and Raymond E. Miller. 1969. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195.

Alexander Koller and Owen Rambow. 2007. Relating dominance formalisms. In *FG'07*.

S. Rao Kosaraju. 1982. Decidability of reachability in vector addition systems. In *STOC'82*, pages 267–281. ACM Press.

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

Jean-Luc Lambert. 1992. A structure to decide reachability in Petri nets. *Theoretical Computer Science*, 99(1):79–104.

Bernard Lang. 1994. Recognition can be harder than parsing. *Computational Intelligence*, 10(4):486–494.

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

Timm Lichte. 2007. An MCTAG with tuples for coherent constructions in German. In *FG'07*.

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

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

Mitchell P. Marcus, Donald Hindle, and Margaret M. Fleck. 1983. D-theory: talking about talking about trees. In *ACL'83*, pages 129–136. ACL Press.

Ernst W. Mayr. 1981. An algorithm for the general Petri net reachability problem. In *STOC'81*, pages 238–246. ACM Press.

Richard Mayr. 1999. Process rewrite systems. *Information and Computation*, 156(1–2):264–286.

Christos H. Papadimitriou. 1994. *Computational Complexity*. Addison-Wesley.

Rohit J. Parikh. 1966. On context-free languages. *Journal of the ACM*, 13(4):570–581.

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

Carl A. Petri. 1962. *Kommunikation mit Automaten*. Ph.D. thesis, University of Bonn.

Charles Rackoff. 1978. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223–231.

Owen Rambow, K. Vijay-Shanker, and David Weir. 1995. D-tree grammars. In *ACL'95*, pages 151–158. ACL Press.

Owen Rambow, David Weir, and K. Vijay-Shanker. 2001. D-tree substitution grammars. *Computational Linguistics*, 27(1):89–121.

Owen Rambow. 1994a. *Formal and Computational Aspects of Natural Language Syntax*. Ph.D. thesis, University of Pennsylvania.

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

Sylvain Salvati. 2010. Minimalist grammars in the light of logic. Manuscript.

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

Anders Søgaard, Timm Lichte, and Wolfgang Maier. 2007. The complexity of linguistically motivated extensions of tree-adjoining grammar. In *RANLP'07*, pages 548–553.

Edward P. Stabler. 1997. Derivational minimalism. In Christian Retoré, editor, *LACL'96*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer.

Kumar Neeraj Verma and Jean Goubault-Larrecq. 2005. Karp-Miller trees for a branching extension of VASS. *Discrete Mathematics and Theoretical Computer Science*, 7(1):217–230.

K. Vijay-Shanker. 1992. Using descriptions of trees in a tree adjoining grammar. *Computational Linguistics*, 18(4):481–517.

Ryo Yoshinaka and Makoto Kanazawa. 2005. The complexity and generative capacity of lexicalized abstract categorial grammars. In Philippe Blache, Edward Stabler, Joan Busquets, and Richard Moot, editors, *LACL'05*, volume 3492 of *Lecture Notes in Computer Science*, pages 330–346. Springer.