Size and computation of injective tree automatic presentations

Dietrich Kuske[1] and Thomas Weidner[2]

[1] Institut für Theoretische Informatik, TU Ilmenau, Germany
[2] Universität Leipzig, Institut für Informatik, Germany

Abstract. It has been shown that every tree automatic structure admits an injective tree automatic presentation, but no good size or time bounds are known. From an arbitrary tree automatic presentation, we construct an equivalent injective one in polynomial space that consequently has exponential size. Furthermore we also prove an exponential lower bound for the size of injective tree automatic presentations.

1 Introduction

Automatic structures allow us to represent relational structures using automata. This approach was introduced in [5, 7] for word automata and later generalized to tree automata in [1, 2]. It gained increasing interest over the last years [9, 10, 8, 16, 14, 15, 6, 12, 13].

Roughly speaking, a structure is tree automatic if it is the quotient structure of a structure, where the carrier set and the relations are given by tree automata, and a tree regular congruence. If this congruence is the identity relation, the structure is called injectively tree automatic. A presentation of a tree automatic structure is a tuple of tree automata that describe the carrier set, the relations, and the congruence; it is injective if the congruence is the identity.

It is known that every tree automatic structure admits an injective presentation [3]. That proof also shows that such an injective presentation can be computed effectively, but no good complexity bounds can be derived. This paper presents a construction of an (exponentially larger) injective presentation from an arbitrary presentation which can be carried out in polynomial space. Furthermore we show that this exponential blowup is unavoidable for some automatic structures.

The first order theory of tree automatic structures is decidable, but the complexity is in general non-elementary. Better complexity bounds are known for tree automatic structures of bounded degree [11]. Since these techniques only work for injective tree automatic structures, our upper bound provides the first upper bound for the uniform first-order theory of tree automatic structures of bounded degree.

2 Preliminaries

In this section we will collect some fundamental definitions and facts. Here the smallest natural number is one. Let Σ be an alphabet, i.e., a finite and nonempty set. We denote all finite words over Σ by Σ^* and for some non-negative number n let $\Sigma^{\leq n}$ be the set of all words of length less than or equal to n.

2.1 Trees

We will only consider binary trees in this article, the generalization to trees of bounded rank is easily possible.

Let $N \subseteq \{1,2\}^*$ be some set of positions. We call N *prefix-closed* if for all $x, y \in \{1,2\}^*$ whenever $x \,.\, y \in N$ holds also $x \in N$ is true. Furthermore, the *frontier of* N is the set $\mathrm{Fr}(N) \stackrel{\text{def}}{=} (N \,.\, \{1,2\}) \setminus N$. This set contains all positions directly below N.

A *tree over* Σ is a partial mapping $t \colon \{1,2\}^* \dashrightarrow \Sigma$ such that $\mathrm{dom}(t)$ is non-empty, finite, prefix-closed, and $x \,.\, 2 \in \mathrm{dom}(t)$ implies $x \,.\, 1 \in \mathrm{dom}(t)$ for all $x \in \{1,2\}^*$. Let T_Σ denote the set of all trees over Σ. The height of a tree is defined by $\mathrm{height}(t) \stackrel{\text{def}}{=} \max\{|x| : x \in \mathrm{dom}(t)\} + 1$. A position $x \in \mathrm{dom}(t)$ is called a leaf position if $x \,.\, \{1,2\} \cap \mathrm{dom}(t) = \emptyset$. Let $s, t \in \mathrm{T}_\Sigma$ and $x \in \mathrm{dom}(t)$. Then the subtree $t \restriction x$ of t rooted at x and the tree $t[x \mapsto s]$ obtained from t by replacing the subtree $t \restriction x$ rooted at x by s are defined as usual. Furthermore $f(t_1, \dots, t_d)$ is the tree t with root-symbol f and $t \restriction i = t_i$ for $1 \leq i \leq d$.

The convolution $t = t_1 \otimes \cdots \otimes t_d$ of the trees $t_1, \dots, t_d \in \mathrm{T}_\Sigma$ is defined by

$$\mathrm{dom}(t) \stackrel{\text{def}}{=} \bigcup_{i=1,\dots,d} \mathrm{dom}(t_i) \text{ and}$$

$$t(x) \stackrel{\text{def}}{=} (f_1, \dots, f_d) \quad \text{where} \quad f_i \stackrel{\text{def}}{=} \begin{cases} t_i(x) & \text{if } x \in \mathrm{dom}(t_i) \\ \square & \text{otherwise.} \end{cases}$$

Here $\square \notin \Sigma$ is a new symbol. We write $\Sigma_\square$ for $\Sigma \cup \{\square\}$ such that the convolution t becomes a tree over $\Sigma_\square^d$.

2.2 Tree automata

A (bottom-up) *tree automaton over* Σ is a tuple $A = (Q, \delta, F)$ where Q is a finite set of states and $F \subseteq Q$ a set of final states. The relation $\delta \subseteq Q \times \Sigma \times Q^{\leq 2}$ is called the transition relation. We also write $\mathrm{Q}(A)$, $\mathrm{F}(A)$ and $\delta(A)$ for Q, F and, respectively, δ.

A *run* of A on some tree $t \in \mathrm{T}_\Sigma$ is a tree $\rho \in \mathrm{T}_Q$ such that $\mathrm{dom}(\rho) = \mathrm{dom}(t)$ and for all position $x \in \mathrm{dom}(t)$, we have

$$(\rho(x), t(x), \rho(x \,.\, 1), \dots, \rho(x \,.\, k)) \in \delta\,,$$

where $0 \leq k \leq 2$ is the number of children of x. We write $\xrightarrow{t} q$ if there is a run ρ on the tree t with $\rho(\varepsilon) = q$. We further write $\xrightarrow{t} X$ for a set $X \subseteq Q$ if there is a $q \in X$ which satisfies $\xrightarrow{t} q$. This notation allows us to define the language of the tree automaton A as

$$\mathrm{L}(A) = \{t \in \mathrm{T}_\Sigma : \xrightarrow{t} F\}\,.$$

A tree automaton $A = (Q, \delta, F)$ is stored by saving all tuples in δ and the final states F. This requires space $\mathrm{O}(\log|Q| + \log|\Sigma|)$ for a single transition. Furthermore we assume that every state occurs as the left side of a transition. Therefore $|F| \leq |Q| \leq |\delta|$ always holds. The space needed to store the automaton A is hence given by $\mathrm{O}\big(|\delta| \cdot (\log|Q| + \log|\Sigma|)\big)$. Following these observations we define the size of A as $|A| \stackrel{\text{def}}{=} |\delta| \cdot (\log|Q| + \log|\Sigma|)$. The number of states $|A|_{\mathrm{Q}} \stackrel{\text{def}}{=} |Q|$ is another size measure.

A tree automaton A over $\Sigma_{\square}^{d}$ is called *d-dimensional tree automaton over Σ*; it recognises the relation

$$\mathrm{R}(A) = \{(t_1, \ldots, t_d) \in \mathrm{T}_{\Sigma}^{d} : t_1 \otimes \cdots \otimes t_d \in \mathrm{L}(A)\} .$$

Note that the size of A is given by $|\delta(A)| \cdot (\log|\mathrm{Q}(A)| + d \cdot \log|\Sigma|)$.

3 An upper bound

Consider an equivalence relation $\sim$ over T_{Σ}. A *complete system of representatives* for ${}^{\mathrm{T}_{\Sigma}}/_{\sim}$ is a set $L \subseteq \mathrm{T}_{\Sigma}$ such that $[t]_{\sim} \cap L$ is a singleton set for all trees $t \in \mathrm{T}_{\Sigma}$. Here $[t]_{\sim} \subseteq \mathrm{T}_{\Sigma}$ is the equivalence class of t with respect to $\sim$.

The aim of this section is to construct in polynomial space from a 2-dimensional tree automaton recognizing an equivalence relation, a tree automaton which recognizes a complete system of representatives for this equivalence relation:

Theorem 3.1. *From a 2-dimensional tree automaton $A_{\sim}$ with $R(A_{\sim})$ an equivalence relation, one can compute in polynomial space a tree automaton B with $|B|_{\mathrm{Q}} = 2^{\mathrm{O}(|A_{\sim}|_{\mathrm{Q}})}$ and $|B| = |\Sigma|^{\mathrm{O}(1)} \cdot 2^{\mathrm{O}(|A_{\sim}|_{\mathrm{Q}})}$ which recognizes a complete system of representatives for ${}^{\mathrm{T}_{\Sigma}}/_{\mathrm{R}(A_{\sim})}$.*

The existence of the tree automaton B from the above theorem was shown in [3] and an analysis of Colcombet and Löding's proof also shows that it is effective, although of multi-exponential complexity. The new singly-exponential blowup is obtained by a refinement of the proof by Colcombet and Löding.

3.1 Shadow

For the rest of this section we fix a 2-dimensional tree automaton $A_{\sim}$ over some alphabet Σ such that $\sim \stackrel{\text{def}}{=} \mathrm{R}(A_{\sim})$ is an equivalence relation on T_{Σ}. We first introduce our notion of the shadow of an equivalence class. For a class $c \in {}^{\mathrm{T}_{\Sigma}}/_{\sim}$ we define the *shadow of c* as

$$\mathrm{S}(c) \stackrel{\text{def}}{=} \bigcap_{t \in c} \mathrm{dom}(t) .$$

Using this definition, we can define descriptions of an equivalence class.

Definition 3.2. *Let $c \in {}^{\mathrm{T}_{\Sigma}}/_{\sim}$. A tree $t \in c$ is a* description of c *if* $\mathrm{height}(t \restriction x) \leq |A_{\sim}|_{\mathrm{Q}}$ *holds for all $x \in \mathrm{dom}(t) \cap \mathrm{Fr}(\mathrm{S}(c))$. The set of all descriptions of c is denoted by* $\mathrm{Desc}(c)$.

Lemma 3.3. *Let $c \in {}^{\mathrm{T}_{\Sigma}}/_{\sim}$. Then* $0 < |\mathrm{Desc}(c)| < \infty$.

Proof. From the very definition, we learn that there are only finitely many descriptions of c. Let $t \in c$ be some tree. We prove the existence of some description by induction over

$$n_t \stackrel{\text{def}}{=} \left| \{x \in \mathrm{dom}(t) \cap \mathrm{Fr}(\mathrm{S}(c)) : \mathrm{height}(t \upharpoonright x) > |A_\sim|_{\mathrm{Q}}\} \right| .$$

If $n_t = 0$ then t is already a description of c. Now assume $n_t > 0$. Hence there is $x \in \mathrm{dom}(t) \cap \mathrm{Fr}(\mathrm{S}(c))$ such that $\mathrm{height}(t \upharpoonright x) > |A_\sim|_{\mathrm{Q}}$. As x is not in the shadow of c there exists a tree $t_0 \in c$ with $x \notin \mathrm{dom}(t_0)$. Let τ be a successful run of $A_\sim$ on $t \otimes t_0$. We consider the tree automaton $B = (\mathrm{Q}(A_\sim), \delta, \{\tau(x)\})$ with

$$\delta \stackrel{\text{def}}{=} \{(q, f, q_1, \ldots, q_k) : (q, (f, \Box), q_1, \ldots, q_k) \in \delta(A_\sim)\} .$$

Since $x \notin \mathrm{dom}(t_0)$ the tree $t \upharpoonright x$ is in the language of B and $\tau \upharpoonright x$ is an accepting run on it. From the pumping lemma [4, Corollary 1.2.3] it follows that there exists a tree $s \in \mathrm{L}(B)$ such that $\mathrm{height}(s) \leq |B|_{\mathrm{Q}} = |A_\sim|_{\mathrm{Q}}$. Let σ be an accepting run of B on s. Then $\sigma(\varepsilon) = \tau(x)$. We define trees $t_1 \stackrel{\text{def}}{=} t[x \mapsto s]$ and $\tau_1 \stackrel{\text{def}}{=} \tau[x \mapsto \sigma]$. One can show that τ_1 is an accepting run of $A_\sim$ on $t_1 \otimes t_0$. Therefore we obtain $t_1 \sim t_0 \sim t$ and from the definition of t_1 we get $n_{t_1} = n_t - 1$. This concludes the inductive proof. □

In the next step, we construct from $A_\sim$ in polynomial space a tree automaton that accepts the set of all descriptions. As an intermediate result we associate the shadow $\mathrm{S}([t])$ with a tree $t \in \mathrm{T}_\Sigma$. Thereto we encode a pair (t, N), where $N \subseteq \mathrm{dom}(t)$ is a set of positions, as a tree over $\Sigma_0 \stackrel{\text{def}}{=} \Sigma \times \{0, 1\}$. This tree is again denoted by (t, N). It is defined as

$$\mathrm{dom}\big((t, N)\big) \stackrel{\text{def}}{=} \mathrm{dom}(t) \qquad \text{and} \qquad (t, N)(x) \stackrel{\text{def}}{=} \big(t(x), \mathbb{1}_N(x)\big) ,$$

where $\mathbb{1}_N$ is the characteristic function of N. This definition allows us to specify the language L_{S} as

$$L_{\mathrm{S}} \stackrel{\text{def}}{=} \{(t, N) \in \mathrm{T}_{\Sigma_0} : N = \mathrm{S}([t])\} .$$

To show the regularity of L_{S}, we prove that the following two languages are regular:

$$L_{\mathrm{S}}^1 \stackrel{\text{def}}{=} \{(t, N) \in \mathrm{T}_{\Sigma_0} : N \subseteq \mathrm{S}([t])\} \text{ and } L_{\mathrm{S}}^2 \stackrel{\text{def}}{=} \{(t, N) \in \mathrm{T}_{\Sigma_0} : N \supseteq \mathrm{S}([t])\} .$$

Lemma 3.4. *From a tree automaton $A_\sim$ with $R(A_\sim)$ an equivalence relation, one can compute in polynomial space a tree automaton recognizing L_{S}^1 with $2^{\mathrm{O}(|A_\sim|_{\mathrm{Q}})}$ many states.*

Proof. Let $t \in \mathrm{T}_\Sigma$ be a tree and $N \subseteq \{1, 2\}^*$ be some set. Then we have $N \subseteq \mathrm{S}([t])$ iff $N \subseteq \mathrm{dom}(s)$ for all trees $s \in \mathrm{T}_\Sigma$ with $s \sim t$. Consider the relation

$$R \stackrel{\text{def}}{=} \{\big((t, N), s\big) \in \mathrm{T}_{\Sigma_0} \times \mathrm{T}_\Sigma : N \not\subseteq \mathrm{dom}(s) \text{ and } s \sim t\}$$

such that L_{S}^1 is the complement of the projection of R on the first component. A tree automaton B for the projection of R whose number of states is linear in $|A_\sim|_{\mathrm{Q}}$ can easily be computed. The complementation of this automaton can be carried out in space polynomial in $|B|$ and therefore polynomial in $|A_\sim|$. □

The other direction is more involved as the proof of the following lemma indicates.

Lemma 3.5. *From a tree automaton $A_\sim$ with $R(A_\sim)$ an equivalence relation, one can compute in polynomial space a tree automaton recognizing the tree language L_{S}^2 with $2^{\mathrm{O}(|A_\sim|_{\mathrm{Q}})}$ many states.*

Proof. Let $A_\sim = (Q, \delta, F)$. We will construct an automaton B with state set 2^Q such that for $t \in \mathrm{T}_\Sigma$, $N \subseteq \mathrm{dom}(t)$ and $\rho \in \mathrm{T}_{2^Q}$ with $\mathrm{dom}(t) = \mathrm{dom}(\rho)$, the following are equivalent:

1. ρ is an accepting run of B on (t, N)
2. – there are trees $s_1, \dots, s_n \in [t]$ and accepting runs $\rho_1, \dots \rho_n$ of $A_\sim$ on $s_i \otimes t$ such that $\rho(x) = \{\rho_1(x), \dots, \rho_n(x)\}$ for all $x \in \mathrm{dom}(t)$ and
   – $\mathrm{dom}(t) \setminus N \subseteq \bigcup_{1 \le i \le n} \mathrm{dom}(t) \setminus \mathrm{dom}(s_i)$.

First, let $\emptyset$ denote the "empty tree" and $Q_\emptyset = \{p \in Q \mid \exists s \in \mathrm{T}_\Sigma : \xrightarrow{\emptyset \otimes s} p\}$. A tuple $(M, (a, x), M_1, \dots, M_k)$ is a transition of B if and only if

- for all $p \in M$, there is $b_p \in \Sigma \cup \{\Box\}$ and $\bar{p} \in \left(\prod_{1 \le i \le k} M_i\right) \times Q_\emptyset^{\le 2-k}$ such that $(p, (a, b_p), \bar{p}) \in \delta$,
- for all $1 \le i \le k$ and $p_i \in M_i$, there are $b \in \Sigma \cup \{\Box\}$, $p_j \in M_j$ for $1 \le j \le k$ and $j \neq i$, and $p \in M$ such that $(p, (a, b), p_1, \dots, p_k) \in \delta$, and
- if $x = 0$, then there exists $p \in M$ so that one can choose $b_p = \Box$ in the first condition.

Finally a set of states $M \subseteq Q$ is accepting if and only if $M \subseteq F$. Now let $(t, N) \in \mathrm{T}_{\Sigma_0}$. Then

$$\begin{aligned}
& B \text{ accepts the pair } (t, N) \in \mathrm{T}_{\Sigma_0} \\
\iff & \text{there exist trees } s_1, \dots, s_n \in [t] \text{ with } \mathrm{dom}(t) \setminus N \subseteq \textstyle\bigcup_{1 \le i \le n} \mathrm{dom}(t) \setminus \mathrm{dom}(s_i) \\
\iff & \mathrm{dom}(t) \setminus N \subseteq \textstyle\bigcup_{s \sim t} \mathrm{dom}(t) \setminus \mathrm{dom}(s) \\
\iff & N \subseteq \textstyle\bigcap_{s \sim t} \mathrm{dom}(s) = S([t]).
\end{aligned}$$

□

Now the regularity of the tree language L_{S} is immediate from Lemmas 3.4 and 3.5:

Corollary 3.6. *From a tree automaton $A_\sim$ with $R(A_\sim)$ an equivalence relation, one can compute in polynomial space a tree automaton recognizing the tree language L_{S} with $2^{\mathrm{O}(|A_\sim|_{\mathrm{Q}})}$ many states.*

3.2 Final steps

In this section we finish the construction of an automaton which recognizes a complete system of representatives for $\mathrm{T}_\Sigma/\!\sim$. For this, fix some linear ordering $\le_\Sigma$ on the finite alphabet Σ. Let s and t be two distinct trees and let p be the lexicographically minimal position where they differ (i.e., from the set $(\mathrm{dom}(s) \setminus \mathrm{dom}(t)) \cup (\mathrm{dom}(t) \setminus \mathrm{dom}(s)) \cup \{q \in \mathrm{dom}(s) \cap \mathrm{dom}(t) : s(q) \neq t(q)\}$). Then set $s <_{\mathrm{T}} t$ provided $p \notin \mathrm{dom}(t)$ or $p \in \mathrm{dom}(s) \cap \mathrm{dom}(t)$ and $s(p) <_\Sigma t(p)$. Then the reflexive closure $\le_{\mathrm{T}}$ of $<_{\mathrm{T}}$ is a linear order that can be accepted by a 2-dimensional tree automaton.

Lemma 3.7. *From a tree automaton $A_\sim$ with $R(A_\sim)$ an equivalence relation, one can compute in polynomial space a tree automaton B with $2^{\mathrm{O}(|A_\sim|_\mathrm{Q})}$ many states that accepts* $\{\min_{\leq_\mathrm{T}} \mathrm{Desc}([t]) : t \in \mathrm{T}_\Sigma\}$.

Proof. First note that $s \in \mathrm{Desc}([t])$ iff $s \sim t$ and $\mathrm{dom}(s) \subseteq \mathrm{S}([t]).\{1,2\}^{<|A_\sim|_\mathrm{Q}}$. Hence $t = \min_{\leq_\mathrm{T}} \mathrm{Desc}([t])$ iff there exists $N \subseteq \{1,2\}^*$ such that

(*1*) $(t, N) \in L_\mathrm{S}$,
(*2*) $\mathrm{dom}(t) \subseteq N \cdot \{1,2\}^{<|A_\sim|_\mathrm{Q}}$, and
(*3*) there is no tree $s \sim t$ with $\mathrm{dom}(s) \subseteq N \cdot \{1,2\}^{<|A_\sim|_\mathrm{Q}}$ and $s <_\mathrm{T} t$.

Classical constructions allow us to build the tree automaton B using the tree automaton from Corollary 3.6. □

Since, by Lemma 3.3, $\min_{\leq_\mathrm{T}} \mathrm{Desc}([t])$ exists for all trees t, the language from the previous lemma is a complete system of representatives for ${}^{\mathrm{T}_\Sigma}/_\sim$. Note that for a tree automaton A the condition $|A|_\mathrm{Q} = 2^{\mathrm{O}(n)}$ implies $|A| = |\Sigma|^{\mathrm{O}(1)} \cdot 2^{\mathrm{O}(n)}$. This is due to $|\delta(A)| \leq |\Sigma| \cdot |\mathrm{Q}(A)|^{\mathrm{O}(1)}$. Therefore Lemma 3.7 also proves Theorem 3.1.

3.3 Application to tree automatic structures

In this part we introduce (injective) tree automatic structures and presentations. We show that every tree automatic presentation can be transformed into an equivalent injective tree automatic presentation of exponential size.

Before we can introduce tree automatic structures we need to define signatures and structures. We call a finite set $\mathcal{S}$ of relation symbols together with their arities $(m_r)_{r\in\mathcal{S}}$ a signature. A $\mathcal{S}$-structure $\mathcal{A}$ is a tuple $(U, (r^\mathcal{A})_{r\in\mathcal{S}})$ where U is an arbitrary set and $r^\mathcal{A} \subseteq U^{m_r}$ ($r \in \mathcal{S}$) are relations. A congruence on $\mathcal{A}$ is an equivalence relation $\sim$ such that

$$(u_1, \ldots, u_{m_r}) \in r^\mathcal{A} \iff (v_1, \ldots, v_{m_r}) \in r^\mathcal{A}$$

for every $r \in \mathcal{S}$ and $u_1 \sim v_1, \ldots, u_{m_r} \sim v_{m_r}$ in U.

A *tree automatic presentation* (over $\mathcal{S}$) is a tuple $\mathcal{P} = (\Sigma, A, A_\sim, (A_r)_{r\in\mathcal{S}})$ such that Σ is an alphabet, A is a tree automaton, A_r are m_r-dimensional tree automata with $\mathrm{R}(A_r) \subseteq \mathrm{L}(A)^{m_r}$ and $A_\sim$ is a 2-dimensional tree automaton such that $\mathrm{R}(A_\sim) \subseteq \mathrm{L}(A)^2$ is a congruence on $(\mathrm{L}(A), (\mathrm{R}(A_r))_{r\in\mathcal{S}})$. The $\mathcal{S}$-structure defined by $\mathcal{P}$ is given by

$$\mathcal{A}(\mathcal{P}) \stackrel{\mathrm{def}}{=} (\mathrm{L}(A), (\mathrm{R}(A_r))_{r\in\mathcal{S}})/\,\mathrm{R}(A_\sim)\,.$$

We say $\mathcal{P}$ is injective if $\mathrm{R}(A_\sim)$ is the identity relation on $\mathrm{L}(A)$. A structure $\mathcal{A}$ is called *(injective) tree automatic structure* if there is an (injective) tree automatic presentation $\mathcal{P}$ such that $\mathcal{A}(\mathcal{P}) \cong \mathcal{A}$.

The size and number of states of $\mathcal{P}$ are the sums of the corresponding values for the automata, i.e.,

$$|\mathcal{P}| \stackrel{\mathrm{def}}{=} |A| + |A_\sim| + \sum_{r\in\mathcal{S}} |A_r| \quad \text{and} \quad |\mathcal{P}|_\mathrm{Q} \stackrel{\mathrm{def}}{=} |A|_\mathrm{Q} + |A_\sim|_\mathrm{Q} + \sum_{r\in\mathcal{S}} |A_r|_\mathrm{Q}\,.$$

We will apply Theorem 3.1 to automatic structures to obtain the next theorem.

Theorem 3.8. *From a tree automatic presentation $\mathcal{P}$ over some alphabet Σ and $M = \max\{m_r : r \in \mathcal{S}\}$, one can compute in polynomial space an injective tree automatic presentation $\mathcal{I}$ such that $\mathcal{A}(\mathcal{P}) \cong \mathcal{A}(\mathcal{I})$, $|\mathcal{I}| = |\Sigma|^{\mathrm{O}(M)} \cdot 2^{\mathrm{O}(M \cdot |\mathcal{P}|_{\mathrm{Q}})}$ and $|\mathcal{I}|_{\mathrm{Q}} = 2^{\mathrm{O}(M \cdot |\mathcal{P}|_{\mathrm{Q}})}$.*

Proof. Let $\mathcal{P} = (\Sigma, A, A_\sim, (A_r)_{r \in \mathcal{S}})$ and $\sim = \mathrm{R}(A_\sim) \subseteq \mathrm{L}(A)^2$. Let $\sim'$ be the least equivalence relation on the whole set T_Σ that contains $\sim$, i.e.,

$$s \sim' t \iff s \sim t \text{ or } s = t$$

for all $s, t \in \mathrm{T}_\Sigma$. Clearly $\sim'$ can be recognized by a tree automaton $A_{\sim'}$ with $\mathrm{O}(|A_\sim|_{\mathrm{Q}})$ many states. Now, we apply Theorem 3.1 to $A_{\sim'}$ and obtain a tree automaton B' such that $\mathrm{L}(B')$ is a complete system of representatives for ${}^{\mathrm{T}_\Sigma}/_{\sim'}$ and $|B'|_{\mathrm{Q}} = 2^{\mathrm{O}(|A_\sim|_{\mathrm{Q}})}$. By standard constructions we compute an automaton B such that $\mathrm{L}(B) = \mathrm{L}(B') \cap \mathrm{L}(A)$ and $|B|_{\mathrm{Q}} = 2^{\mathrm{O}(|\mathcal{P}|_{\mathrm{Q}})}$. By definition the set $\mathrm{L}(A)$ is $\sim$-closed, and therefore $\sim'$-closed. Hence $\mathrm{L}(B)$ is a complete system of representatives for ${}^{\mathrm{L}(A)}/_{\sim}$. Finally, we construct for every $r \in \mathcal{S}$ a tree automaton B_r by intersecting every component of $\mathrm{R}(A_r)$ with $\mathrm{L}(B)$. Then $|B_r|_{\mathrm{Q}} = \mathrm{O}(|A_r|_{\mathrm{Q}} \cdot |B|_{\mathrm{Q}}^{m_r})$. Let $\mathcal{I} = (B, (B_r)_{r \in \mathcal{S}})$. As $\sim$ is a congruence, $\mathcal{I}$ is the wanted injective tree automatic presentation. □

One may want an estimate for the size of $\mathcal{I}$ which only depends on $|\mathcal{P}|$. It is easy to check that for any tree automaton A, we have $|A|_{\mathrm{Q}} \cdot \log |A| = \mathrm{O}(|A|)$. We may also assume that a tree automatic presentation actually uses every letter of its alphabet. Therefore we can assume $|\Sigma| \leq |A| \leq |\mathcal{P}|$. In the proof of Theorem 3.8 we now have $\log |B| = \mathrm{O}(\log |\Sigma| + {}^{|A_\sim|}/_{\log |A_\sim|}) = \mathrm{O}({}^{|\mathcal{P}|}/_{\log |\mathcal{P}|})$. Hereby we get

$$|\mathcal{I}| = |B| + \sum_{r \in \mathcal{S}} |B_r| = \sum_{r \in \mathcal{S}} 2^{\mathrm{O}(M \cdot |\mathcal{P}| / \log |\mathcal{P}|)} \cdot |A_r| = 2^{\mathrm{O}(M \cdot |\mathcal{P}| / \log |\mathcal{P}|)} .$$

This result allows us to answer an open question from [11] that asks for the complexity of the uniform model checking of tree automatic structures of bounded degree.[3] A structure $\mathcal{A} = (U, (r^{\mathcal{A}})_{r \in \mathcal{S}})$ has *bounded degree* if there exists a natural number d such that any $x \in U$ belongs to at most d tuples from $\bigcup_{r \in \mathcal{S}} r^{\mathcal{A}}$. Then one obtains

Corollary 3.9. *The set of pairs $(\mathcal{P}, \varphi)$ with $\mathcal{P}$ a tree automatic presentation of a structure of bounded degree and φ a first-order sentence such that $\mathcal{A}(\mathcal{P}) \models \varphi$ is decidable in 5-fold exponential time.*

Proof. In the given time bound, $\mathcal{P}$ can be transformed into an equivalent injective presentation $\mathcal{P}'$ of exponential size. Then the result follows from [11, Cor. 3.8]. □

4 A lower bound

From Section 3 we know that we can construct both, a tree automaton recognizing a complete system of representatives and an injective tree automatic presentation of

[3] This answer is incorporated in the journal version of [11] that will appear in the Journal of Symbolic Logic.

exponential size. In this section we show that there is also an exponential lower bound in both cases. To show this, we construct a tree automatic structure and show that every injective tree automatic presentation of this structure or tree automaton recognizing a complete system of representatives for the carrier set is at least of exponential size.

4.1 State complexity of complementation revisited

In this section, we will construct "small" automata A_n such that any automaton B accepting a language of size $m = |\Sigma^* \setminus \mathrm{L}(A_n)|$ is "large". Choosing an appropriate alphabet of size m, two states would suffice for the automaton B. Therefore, we will not consider the number of states $|B|_{\mathrm{Q}}$, but the value $|B|_{\mathrm{Q}} + \log |\Gamma|$ where Γ is the alphabet of B.

We now fix the alphabet $\Sigma \stackrel{\text{def}}{=} \{0, 1\}$. For some non-empty word $w = x_0 \dots x_{k-1}$ in Σ^+ let $\mathrm{val}(w) = \sum_{0 \le i < k} x_i 2^i$ be the value of w as binary number, where the lowest bit is at the first position. Vice versa let $\mathrm{bin}_k(n)$ be the unique word $w \in \Sigma^*$ of length k such that $\mathrm{val}(w) = n$ (if $n \ge 2^k$, then $\mathrm{bin}_k(n)$ is undefined). We now consider word languages over Σ^3 and we will view words in $(\Sigma^3)^*$ as convolutions of three words over Σ of equal length.[4]

Definition 4.1. *Let $n \in \mathbb{N}$. The language L_n^{W} consists of all words $w \in (\Sigma^3)^*$ that satisfy*

(i) $|w| = k \cdot n$ *for some* $k > 0$*. Therefore* $w = (u_1 \otimes v_1 \otimes x_1) \cdots (u_k \otimes v_k \otimes x_k)$ *for some* $u_i, v_i, x_i \in \Sigma^n$*.*
(ii) $\mathrm{val}(u_1) = \mathrm{val}(v_1) = 0$ *and* $\mathrm{val}(u_k) = 2^n - 1$*, i.e.,* $u_1 = v_1 = 0^n$ *and* $u_k = 1^n$*.*
(iii) $\mathrm{val}(u_i) = \mathrm{val}(v_i) + 1$ *for all* $i \in \{2, \dots, k\}$*.*
(iv) $v_{i+1} = u_i$ *for all* $i \in \{1, \dots, k-1\}$*.*

Furthermore, the tree language L_n^{T} consists of all trees $t \in \mathrm{T}_{\Sigma^3}$ satisfying

(1) $u \,.\, 1 \in \mathrm{dom}(t)$ *implies* $u \,.\, 2 \in \mathrm{dom}(t)$*, i.e.,* t *is a complete binary tree and*
(2) $t(\varepsilon)t(x_1)t(x_1x_2) \cdots t(x_1 \cdots x_\ell) \in L_n^{\mathrm{W}}$ *for all leaf positions* $x_1 \cdots x_\ell \in \mathrm{dom}(t)$*.*

The languages we are really interested in are the complements of the sets L_n^{W} and L_n^{T}. Therefore, we next show that these complements can be recognized by "small" automata.

Lemma 4.2. *Let $n \in \mathbb{N}$. There is a word automaton A and a tree automaton B such that the following hold:*

$$|A| = \mathrm{O}(n \cdot \log n) \qquad |A|_{\mathrm{Q}} = \mathrm{O}(n) \qquad \mathrm{L}(A) = (\Sigma^3)^* \setminus L_n^{\mathrm{W}}$$
$$|B| = \mathrm{O}(n \cdot \log n) \qquad |B|_{\mathrm{Q}} = \mathrm{O}(n) \qquad \mathrm{L}(B) = \mathrm{T}_{\Sigma^3} \setminus L_n^{\mathrm{T}}$$

[4] Convolutions of words are analogously defined to convolutions of trees.

Proof. The negations of the conditions (*i*) to (*iii*) can be checked by deterministic word automata with $\mathrm{O}(n)$ many states and of size $\mathrm{O}(n \cdot \log n)$. The negation of condition (*iv*) is "there is a position $j \in \{1, \dots, |w| - n\}$ such that the first component of w_j does not equal the second one of w_{j+n}". Certainly, this property can be checked by a non-deterministic word automaton with the same size and number of states. Therefore $\left(\Sigma^3\right)^* \setminus L_n^{\mathrm{W}}$ can be recognized by a word automaton with $\mathrm{O}(n)$ many states and of size $\mathrm{O}(n \cdot \log n)$.

The complement of L_n^{T} consists of all trees which contain a leaf position x such that the word of the labels from the root to x is in $(\Sigma^3)^* \setminus L_n^{\mathrm{W}}$. Again, this can be checked by a non-deterministic tree automaton with $\mathrm{O}(n)$ many states and of size $\mathrm{O}(n \cdot \log n)$. □

To show that no tree language of size $\left|L_n^{\mathrm{T}}\right|$ can be accepted by a "small" automaton, we next estimate the size of the languages L_n^{W} and L_n^{T}. For these estimations, define $\exp(0, n) \overset{\mathrm{def}}{=} n$ and $\exp(k+1, n) \overset{\mathrm{def}}{=} 2^{\exp(k,n)}$ for $n, k \in \mathbb{N}_0$.

Lemma 4.3. *Let* $n \in \mathbb{N}$ *be at least* 2*. Then* $\exp(2, n) \leq \left|L_n^{\mathrm{W}}\right| < \infty$ *and* $\exp(3, n) \leq \left|L_n^{\mathrm{T}}\right| < \infty$.

Proof. Let $w = (u_1 \otimes v_1 \otimes x_1) \cdots (u_k \otimes v_k \otimes x_k) \in L_n^{\mathrm{W}}$ with $u_i, v_i, x_i \in \Sigma^n$. By induction it follows that $\mathrm{val}(u_i) = i - 1$ and $\mathrm{val}(v_i) = \max\{i - 2, 0\}$ for all $i \in \{1, \dots, k\}$. By definition we have $\mathrm{val}(u_k) = 2^n - 1$ and therefore $k = 2^n$. Hence the word w is of length $n \cdot 2^n$ and L_n^{W} is finite. Vice versa every word $w \in (\Sigma^3)^*$ which is defined as above is in the language L_n^{W}. As we can still choose the x_i arbitrary, there are at least $|\Sigma|^{n \cdot 2^n} \geq \exp(2, n)$ many words in L_n^{W}.

From the definition of L_n^{T} and the observations for L_n^{W}, we obtain that L_n^{T} consists of full binary trees of height $n \cdot 2^n$. Since also here, the third components of the labelings can be chosen freely, L_n^{T} contains the set of Σ-labeled full binary trees of height $n \cdot 2^n$. But this set contains (for $n \geq 2$) $\exp(1, \exp(1, n \cdot 2^n) - 1) \geq \exp(3, n)$ elements. □

Proposition 4.4. *Let* $n \in \mathbb{N}$ *and* B *be a tree automaton over some alphabet* Γ *recognizing a finite language of size at least* $\exp(3, n)$*. Then* $|B|_{\mathrm{Q}} + \log(\Gamma) \geq 2^n$.

Proof. We first verify $|\mathrm{L}(B)| \leq \exp(2, |B|_{\mathrm{Q}} + \log|\Gamma|)$ by contradiction: Assume $\mathrm{L}(B)$ contains more than $\exp(2, |B|_{\mathrm{Q}} + \log|\Gamma|)$ elements. Then $\mathrm{L}(B)$ contains at least one tree of height greater than $|B|_{\mathrm{Q}}$. Hence, by the pumping lemma for regular tree languages [4, Corollary 1.2.3], $\mathrm{L}(B)$ is infinite, a contradiction.

Hence we have $\exp(3, n) \leq |\mathrm{L}(B)| \leq \exp(2, |B|_{\mathrm{Q}} + \log|\Gamma|)$ and therefore, as claimed, $|B|_{\mathrm{Q}} + \log|\Gamma| \geq 2^n$. □

4.2 A tree automatic structure

Let $L \subseteq \mathrm{T}_\Sigma$ be some tree language. Our idea is to express the set complement $\mathrm{T}_\Sigma \setminus L$, or at least a set with the same cardinality, in terms of some injective tree automatic presentation.

Definition 4.5. *Let $n \in \mathbb{N}$ and let* X *and* Y *be new symbols not in* $\Sigma = \{0,1\}^3$. *The tree automatic structure* $\mathcal{A}_n \stackrel{\text{def}}{=} {}^{(U,R)}/_{\sim_n}$ *is then given by*

$$U \stackrel{\text{def}}{=} \{\mathrm{X}(t), \mathrm{Y}(t) \in \mathrm{T}_{\Sigma\cup\{\mathrm{X},\mathrm{Y}\}} : t \in \mathrm{T}_\Sigma\}$$
$$R \stackrel{\text{def}}{=} \{\big(x(t), y(t)\big) \in U^2 : t \in \mathrm{T}_\Sigma,\ \{x,y\} = \{\mathrm{X},\mathrm{Y}\}\}, \textit{ and}$$
$$\sim_n \stackrel{\text{def}}{=} \{\big(x(t), y(t)\big) \in U^2 : t \in \mathrm{T}_\Sigma,\ x, y \in \{\mathrm{X},\mathrm{Y}\} \textit{ and } (t \in L_n^{\mathrm{T}} \Rightarrow x = y)\}\,.$$

Since the relation R is symmetric, we can think of the structures (U,R) and therefore $\mathcal{A}_n$ as an undirected graph. Then (U,R) is the disjoint union of infinitely many disjoint edges and $\mathcal{A}_n$ that of $|L_n^{\mathrm{T}}|$ disjoint edges and infinitely many isolated nodes (with a self-loop). Note that U and R can be accepted by a (2-dimensional) tree automaton and do not depend on n. To also accept the relation $\sim_n$, we modify the automaton B from Lemma 4.2 in the obvious way. Hence $\sim_n$ can be accepted by a two-dimensional automaton $A_{\sim_n}$ with $|A_{\sim_n}|_{\mathrm{Q}} = \mathrm{O}(n)$ and $|A_{\sim_n}| = \mathrm{O}(n \cdot \log n)$. This proves the next lemma.

Lemma 4.6. *There is $C_1 > 0$ such that for all $n \in \mathbb{N}$ there is a tree automatic presentation $\mathcal{P}_n$ of $\mathcal{A}_n$ with $|\mathcal{P}_n| \le C_1 \cdot n \cdot \log n$ and $|\mathcal{P}_n|_{\mathrm{Q}} \le C_1 \cdot n$.*

We now come to the lower bound for injective tree automatic presentations of $\mathcal{A}_n$:

Lemma 4.7. *There exist constants $c, d \ge 1$ such that for any $n \in \mathbb{N}$ with $n \ge 2$ and any injective tree automatic presentation $\mathcal{I}$ of $\mathcal{A}_n$, we have $2^n \le c \cdot |\mathcal{I}|_{\mathrm{Q}} + \log|\Gamma|$ and $2^n \le d \cdot |\mathcal{I}|$.*

Proof. Let $\mathcal{I} = (\Gamma, B, B_R)$ and consider the set

$$L = \{t \in \mathrm{L}(B) : (s,t) \in \mathrm{R}(B_R) \text{ for some } s \in \mathrm{L}(B) \setminus \{t\}\}\,.$$

The relation $\{(s,t) \in \mathrm{T}_\Gamma^2 : s \ne t\}$ can be accepted by a tree automaton with two states. Running this automaton in parallel with B_R, we get an automaton with $2 \cdot |B_R|_{\mathrm{Q}}$ states that accepts the relation

$$\{(s,t) \in \mathrm{R}(B_R) : s \ne t\}\,.$$

Then L is the projection of this relation to the first component. Hence L can be accepted by an automaton C with $|C|_{\mathrm{Q}} \le c \cdot |B_R|_{\mathrm{Q}}$. Note that L is the set of nodes of $\mathcal{A}_n$ that are connected to some other node, hence $|L| = 2 \cdot \left|L_n^{\mathrm{T}}\right|$. From Lemma 4.3 and Proposition 4.4, we therefore get $|C|_{\mathrm{Q}} + \log|\Gamma| \ge 2^n$. Hence $2^n \le |C|_{\mathrm{Q}} + \log|\Gamma| \le c \cdot |B_R|_{\mathrm{Q}} + \log|\Gamma| \le c \cdot |\mathcal{I}|_{\mathrm{Q}} + \log|\Gamma|$.

We can assume that every symbol from Γ actually appears in a tree in $\mathrm{L}(A)$ and hence in some transition in $\delta(A)$ such that $\log|\Gamma| \le |\Gamma| \le |\delta(A)| \le |A| \le |\mathcal{I}|$. Similarly, we can assume $|Q| \le |\delta|$ for any automaton implying $|\mathcal{I}|_{\mathrm{Q}} \le |\mathcal{I}|$. Therefore for the size of $\mathcal{I}$ we get $2^n \le c \cdot |\mathcal{I}|_{\mathrm{Q}} + \log|\Gamma| \le (c+1) \cdot |\mathcal{I}|$. □

Now we are able to prove the lower bounds for injective tree automatic presentations.

Theorem 4.8. *There are $C_1, C_2, C_3 > 0$ such that the following hold:*

1. *for every $n \in \mathbb{N}$ there exists a tree automatic presentation $\mathcal{P}$ such that $|\mathcal{P}|_{\mathrm{Q}} \leq C_1 \cdot n$, $|\mathcal{P}| \leq C_1 \cdot n \cdot \log n$, and for every injective tree automatic presentation $\mathcal{I}$ of $\mathcal{A}(\mathcal{P})$ over some alphabet Γ it holds that $|\mathcal{I}|_{\mathrm{Q}} \geq C_2 \cdot 2^n - \log |\Gamma|$.*
2. *for every $m \in \mathbb{N}$ there exists a tree automatic presentation $\mathcal{P}$ such that $|\mathcal{P}| \leq C_1 \cdot m$, and for every injective tree automatic presentation $\mathcal{I}$ of $\mathcal{A}(\mathcal{P})$ it holds that $|\mathcal{I}| \geq C_2 \cdot 2^{\frac{m}{\log m}}$.*

Proof. Let c and d be the constants from Lemma 4.7. Then the first claim follows immediately from Lemmas 4.6 and 4.7 with $C_2 = 1/c$.

Choose $n = \lfloor m/\log m \rfloor$. Then $|\mathcal{P}_n| \leq C_1 \cdot \frac{m}{\log m} \cdot \log \frac{m}{\log m} \leq C_1 \cdot m$. Let $\mathcal{I}$ be some injective tree automatic presentation of $\mathcal{A}(\mathcal{P}_n)$. Then $|\mathcal{I}| \geq \frac{1}{d} \cdot 2^n$. Since $n > \frac{m}{\log m} - 1$, we get $|\mathcal{I}| \geq C_3 \cdot 2^{\frac{m}{\log m}}$ with $C_3 = 1/d$. □

4.3 Transfer to word automatic structures

Word automatic structures have been introduced in [5, 7]. Essentially they are defined as tree automatic structures, but with word automata instead of tree automata. Every word automatic presentation admits an injective word automatic presentation of exponential size. This has already been shown in [7]. In this section we will transfer our lower bound result for tree automatic structures to word automatic ones.

We use the languages L_n^{W} from Definition 4.1 to replace the languages L_n^{T} from the tree automatic case. Let $n \in \mathbb{N}$ and again $\Sigma = \{0,1\}^3$, and let $\mathrm{X}, \mathrm{Y} \notin \Sigma$ be two symbols. We define the structure $\mathcal{A}_n^{\mathrm{W}}$ by $\mathcal{A}_n^{\mathrm{W}} \stackrel{\text{def}}{=} (U^{\mathrm{W}}, R^{\mathrm{W}})/\sim_n^{\mathrm{W}}$ with

$$U^{\mathrm{W}} \stackrel{\text{def}}{=} \{w \,.\, X, w \,.\, Y : w \in \Sigma^*\}\,,$$

$$R^{\mathrm{W}} \stackrel{\text{def}}{=} \{(w \,.\, x, w \,.\, y) : w \in \Sigma^*,\ \{x, y\} = \{\mathrm{X}, \mathrm{Y}\}\}, \text{ and}$$

$$\sim_n^{\mathrm{W}} \stackrel{\text{def}}{=} \{(w \,.\, x, w \,.\, y) : w \in \Sigma^*,\ x, y \in \{\mathrm{X}, \mathrm{Y}\} \text{ and } (w \in L_n^{\mathrm{W}} \Rightarrow x = y)\}\,.$$

Now fix an injective word automatic presentation $\mathcal{I} = (\Gamma, B, B_R)$ with word automata B and B_R over Γ. By Lemma 4.3 and similar arguments as in Proposition 4.4, and Lemma 4.7 we obtain $c \cdot |\mathcal{I}|_{\mathrm{Q}} + \log |\Gamma| \geq 2^n$ and $d \cdot |\mathcal{I}| \geq 2^n$ for constants $c, d > 0$. Together with corresponding arguments as before Lemma 4.6 and as in Theorem 4.8 we obtain the following theorem.

Theorem 4.9. *There are constants $C_1, C_2, C_3 > 0$ such that for every $n \in \mathbb{N}$ with $n \geq 2$ there exists a word automatic structure $\mathcal{P}$ such that $|\mathcal{P}| \leq C_1 \cdot n \cdot \log n$ and for every injective word automatic presentation $\mathcal{I}$ of $\mathcal{A}(\mathcal{P})$ it holds that $|\mathcal{I}| \geq C_2 \cdot 2^{\frac{n}{\log n}}$, and $|\mathcal{I}|_{\mathrm{Q}} \geq C_3 \cdot 2^n$.*

Remark 4.10. We define an injective tree automatic presentation of the word automatic structure $\mathcal{A}_n^{\mathrm{W}}$ as follows. Let $\Sigma = \{0,1\}$ and let L comprise all trees $t \in \mathrm{T}_\Sigma$ with $\mathrm{dom}(t) = 0^{<n} \,.\, 1 \,.\, \{0,1\}^{<n}$ implying $|\,\mathrm{dom}(t)| = n + n \cdot (2^n - 1) = n \cdot 2^n$ for $t \in L$. Furthermore, define $U \stackrel{\text{def}}{=} \{x(t) : x \in \{X, Y\}, t \in \mathrm{T}_\Sigma, (t \in L \Rightarrow x = X)\}$. Finally, let $R = \{(x(t), y(t)) \in U^2 : x, y \in \{X, Y\}, (x = y \Rightarrow t \in L)\}$. Then $\mathcal{A}_n^{\mathrm{W}} \cong (U, R)$ and U and R can be accepted by tree automata with $\mathrm{O}(n)$ many states.

So the following open question remains: Does the exponential blowup also occur when we move from an arbitrary *word* automatic presentation to an equivalent injective *tree* automatic presentation?

References

1. A. Blumensath. Automatic structures. Diplomarbeit, RWTH Aachen, 1999.
2. A. Blumensath and E. Grädel. Automatic Structures. In *LICS'00*, pages 51–62. IEEE Computer Society Press, 2000.
3. Th. Colcombet and Ch. Löding. Transforming structures by set interpretations. *Logical Methods in Computer Science*, 3(2), 2007.
4. H. Comon, M. Dauchet, R. Gilleron, Ch. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata`, 2007. release October, 12th 2007.
5. B.R. Hodgson. On direct products of automaton decidable theories. *Theor. Comput. Sci.*, 19:331–335, 1982.
6. A. Kartzow. Collapsible pushdown graphs of level 2 are tree-automatic. In *STACS '10*, Leibniz International Proceedings in Informatics (LIPIcs) vol. 5, pages 501–512. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2010.
7. B. Khoussainov and A. Nerode. Automatic presentations of structures. In *LCC'94*, Lecture Notes in Computer Science vol. 960, pages 367–392. Springer, 1994.
8. B. Khoussainov, A. Nies, S. Rubin, and F. Stephan. Automatic structures: Richness and limitations. *Logical Methods in Computer Science*, 3(2), 2007.
9. B. Khoussainov, S. Rubin, and F. Stephan. Definability and regularity in automatic structures. In *STACS'04*, Lecture Notes in Computer Science vol. 2996, pages 440–451. Springer, 2004.
10. B. Khoussainov, S. Rubin, and F. Stephan. Automatic linear orders and trees. *ACM Trans. Comput. Log.*, 6(4):675–700, 2005.
11. D. Kuske and M. Lohrey. Automatic structures of bounded degree revisited. http://arxiv.org/abs/0810.4998, 2008. To appear in *Journal of Symbolic Logic* in slightly extended form.
12. D. Kuske and M. Lohrey. Some natural problems in automatic graphs. *J. Symbolic Logic*, 75(2):678-710, 2010.
13. D. Kuske, J. Liu, and M. Lohrey. The isomorphism problem on classes of automatic structures. In *LICS'10*, pages 160–169. IEEE Computer Society, 2010.
14. L. Libkin and A. W. To. Recurrent reachability analysis in regular model checking. In *LPAR '08*, Lecture Notes in Comp. Science vol. 5330, pages 198–213. Springer, 2008.
15. L. Libkin and A. W. To. Algorithmic metatheorems for decidable LTL model checking over infinite systems. In *FOSSACS '10*, Lecture Notes in Comp. Science vol. 6014, pages 221–236. Springer, 2010.
16. S. Rubin. Automata presenting structures: A survey of the finite string case. *Bulletin of Symbolic Logic*, 14:169–209, 2008.