Decidability and Complexity in Automatic Monoids *

Markus Lohrey

Lehrstuhl für Informatik I, RWTH Aachen, Germany
`lohrey@i1.informatik.rwth-aachen.de`

Abstract. We prove several complexity and decidability results for automatic monoids: (i) there exists an automatic monoid with a P-complete word problem, (ii) there exists an automatic monoid such that the first-order theory of the corresponding Cayley-graph is not elementary decidable, and (iii) there exists an automatic monoid such that reachability in the corresponding Cayley-graph is undecidable. Moreover, we show that for every hyperbolic group the word problem belongs to LOGCFL, which improves a result of Cai [4].

1 Introduction

Automatic groups attracted a lot of attention in combinatorial group theory during the last 15 years, see e.g. the textbook [11]. Roughly speaking, a finitely generated group $\mathcal{G}$, generated by the finite set Γ, is automatic, if the elements of $\mathcal{G}$ can be represented by words from a regular language over Γ, and the multiplication with a generator on the right can be recognized by a synchronized 2-tape automaton. This concept easily yields a quadratic time algorithm for the word problem of an automatic group.

It is straight forward to extend the definition of an automatic group to the monoid case; this leads to the class of *automatic monoids*, see e.g. [6, 13, 16, 26]. In the present paper, we study the complexity and decidability of basic algorithmic questions in automatic monoids. In Section 4 we consider the complexity of the word problem for automatic monoids. Analogously to the group case, it is easy to show that for every automatic monoid the word problem can be solved in quadratic time. Here, we prove that there exists a fixed automatic monoid with a P-complete word problem. Thus, unless P = NC, where NC is the class of all problems that can be solved in polylogarithmic time using a polynomial amount of hardware, there exist automatic monoids for which the word problem cannot be efficiently parallelized. Whether there exists an automatic *group* with a P-complete word problem was asked for the first time by Cai [4]. This problem remains open.

An important subclass of the class of automatic groups is the class of *hyperbolic groups*, which are defined via a geometric hyperbolicity condition on the

* This work was partly done while the author was at FMI, University of Stuttgart, Germany.

Cayley-graph. In [4], Cai has shown that for every hyperbolic group the word problem belongs to the parallel complexity class NC^2. Cai also asked, whether the upper bound of NC^2 can be improved. Using known results from formal language theory, we show in Section 4 that the word problem for every hyperbolic group belongs to the complexity class $\mathrm{LOGCFL} \subseteq \mathrm{NC}^2$. LOGCFL is the class of all problems that are logspace reducible to a context-free language [32]. We also present a class of automatic monoids, namely monoids that can be presented by finite, terminating, confluent, and left-basic semi-Thue systems [29], for which the complexity of the word problem captures the class LOGDCFL (the logspace closure of the *deterministic* context-free languages).

In Section 5 we study *Cayley-graphs* of automatic monoids. The Cayley-graph of a finitely generated monoid $\mathcal{M}$ wrt. a finite generating set Γ is a Γ-labeled directed graph with node set $\mathcal{M}$ and an a-labeled edge from a node x to a node y if $y = xa$ in $\mathcal{M}$. Cayley-graphs of groups are a fundamental tool in combinatorial group theory [23] and serve as a link to other fields like topology, graph theory, and automata theory, see, e.g., [24, 25]. Results on the geometric structure of Cayley-graphs of automatic monoids can be found in [30, 31]. Here we consider Cayley-graphs from a logical point of view, see [20, 21] for previous results in this direction. More precisely, we consider the first-order theory of the Cayley-graph of an automatic monoid $\mathcal{M}$. This theory contains all true statements of the Cayley-graph that result from atomic statements of the form "there is an a-labeled edge between two nodes" using Boolean connectives and quantification over nodes. From the definition of an automatic monoid it follows immediately that the Cayley-graph of an automatic monoid is an automatic graph in the sense of [1, 18]; hence, by a result from [18], its first-order theory is decidable. This allows to verify non-trivial properties for automatic monoids, like for instance right-cancellativity. Here, we prove that there exists an automatic monoid such that the first-order theory of the corresponding Cayley-graph is not elementary decidable. This result sharpens a corresponding statement for general automatic graphs [1]. We remark that, using a result from [22], the Cayley-graph of a right-cancellative automatic monoid has an elementarily decidable first-order theory. Finally we prove that there exists an automatic monoid $\mathcal{M}$ such that reachability in the Cayley-graph (i.e., the question whether for given monoid elements u and v there exists $x \in \mathcal{M}$ with $u = vx$ in $\mathcal{M}$) is undecidable.

2 Monoids and word problems

More details and references concerning the material in this section can be found in [3]. In the following, let Γ be always a *finite* alphabet of symbols. A semi-Thue system R over Γ is a (not necessarily finite) set $R \subseteq \Gamma^* \times \Gamma^*$; its elements are called rules. A rule (s, t) will be also written as $s \to t$. W.l.o.g. we may assume that every symbol from Γ appears in a rule of R; thus, Γ is given uniquely by R. Let $\mathrm{dom}(R) = \{\ell \mid \exists r : (\ell, r) \in R\}$ and $\mathrm{ran}(R) = \{r \mid \exists \ell : (\ell, r) \in R\}$. We define the binary relation $\to_R$ on Γ^* by: $x \to_R y$ if $\exists u, v \in \Gamma^* \, \exists (s, t) \in R : x = usv$ and $y = utv$. Let $\overset{*}{\leftrightarrow}_R$ by the smallest equivalence relation on Γ^* containing $\to_R$; it is a congruence wrt. the concatenation of words and called the *Thue-congruence* associated with R. Hence, we can define the quotient monoid $\Gamma^*/\overset{*}{\leftrightarrow}_R$, which is briefly denoted by Γ^*/R. Let $\pi_R : \Gamma^* \to \Gamma^*/R$ be the canonical surjective monoid homomorphism that maps a word $w \in \Gamma^*$ to its equivalence class wrt. $\overset{*}{\leftrightarrow}_R$. A monoid $\mathcal{M}$ is *finitely generated* if it is isomorphic to a monoid of the form Γ^*/R. In this case, we also say that $\mathcal{M}$ *is finitely generated by* Γ. If in addition to Γ also R is finite, then $\mathcal{M}$ is a *finitely presented monoid*. The *word problem of* $\mathcal{M} \simeq \Gamma^*/R$ *wrt.* R is the set $\{(u,v) \in \Gamma^* \times \Gamma^* \mid \pi_R(u) = \pi_R(v)\}$; it is undecidable in general. If a monoid $\mathcal{M}$ is isomorphic to both Γ^*/R and Σ^*/S for semi-Thue systems R and S, then the word problem of $\mathcal{M}$ wrt. R is logspace-reducible to the word problem of $\mathcal{M}$ wrt. S. Hence, since we are only interested in the decidability (resp. complexity) status of word problems, it makes sense to speak just of the word problem of $\mathcal{M}$.

The semi-Thue system R is *terminating* if there does not exist an infinite chain $s_1 \to_R s_2 \to_R s_3 \to_R \cdots$ in Γ^*. The set of *irreducible words* wrt. R is $\mathrm{IRR}(R) = \{s \in \Gamma^* \mid \neg\exists t \in \Gamma^* : s \to_R t\}$. The system R is *confluent* (resp. *locally confluent*) if for all $s,t,u \in \Gamma^*$ with $s \overset{*}{\to}_R t$ and $s \overset{*}{\to}_R u$ (resp. $s \to_R t$ and $s \to_R u$) there exists $w \in \Gamma^*$ with $t \overset{*}{\to}_R w$ and $u \overset{*}{\to}_R w$. If R is terminating, then by Newman's lemma R is confluent if and only if R is locally confluent. Using *critical pairs* [3] which result from overlapping left-hand sides of R, local confluence is decidable for finite terminating semi-Thue systems. The system R is *length-reducing* if $|s| > |t|$ for all $(s,t) \in R$, where $|w|$ is the length of a word w. The system R is called *length-lexicographic* if there exists a linear order $\succ$ on the alphabet Γ such that for every rule $(s,t) \in R$ either $|s| > |t|$ or ($|s| = |t|$ and there are $u,v,w \in \Gamma^*$ and $a,b \in \Gamma$ such that $s = uav$, $t = ubw$, and $a \succ b$). Clearly, every length-lexicographic semi-Thue system is terminating. In the case when R is terminating and confluent, then every word s has a unique *normal form* $\mathrm{NF}_R(s) \in \mathrm{IRR}(R)$ such that $s \overset{*}{\to}_R \mathrm{NF}_R(s)$ and moreover, the function $\pi_R \restriction \mathrm{IRR}(R)$ (i.e., π_R restricted to $\mathrm{IRR}(R)$) is bijective. Thus, if moreover R is finite, then the word problem of Γ^*/R is decidable: $\pi_R(s) = \pi_R(t)$ if and only if $\mathrm{NF}_R(s) = \mathrm{NF}_R(t)$.

3 Automatic monoids

Automatic monoids were investigated for instance in [6, 13, 14, 16, 26]. They generalize automatic groups, see [11]. Let us fix a finite alphabet Γ. Let $\# \notin \Gamma$ be an additional padding symbol and let $\Gamma_\# = \Gamma \cup \{\#\}$. We define two encodings $\nu_\ell, \nu_r : \Gamma^* \times \Gamma^* \to (\Gamma_\# \times \Gamma_\#)^*$ as follows: Let $u, v \in \Gamma^*$ and let $k = \max\{|u|, |v|\}$. Define $w = u\#^{k-|u|}$, $x = v\#^{k-|v|}$, $y = \#^{k-|u|}u$, and $z = \#^{k-|v|}v$. Let $w[i]$ denote the i-th symbol of w and similarly for x, y, and z. Then

$$\nu_r(u,v) = (w[1], x[1]) \cdots (w[k], x[k]) \text{ and } \nu_\ell(u,v) = (y[1], z[1]) \cdots (y[k], z[k]).$$

For instance, $\nu_r(aba, bbabb) = (a,b)(b,b)(a,a)(\#,b)(\#,b)$ and $\nu_\ell(aba, bbabb) = (\#,b)(\#,b)(a,a)(b,b)(a,b)$. In the following let $\alpha, \beta \in \{\ell, r\}$.

A relation $R \subseteq \Gamma^* \times \Gamma^*$ is called α-automatic if the language $\{\nu_\alpha(u,v) \mid (u,v) \in R\}$ is a regular language over the alphabet $\Gamma_\# \times \Gamma_\#$. The following simple lemma will turn out to be useful. Its simple proof is left to the reader. A relation $R \subseteq \Gamma^* \times \Gamma^*$ has *bounded length-difference* if there exists a constant γ such that for all $(u,v) \in R$, $|(|u| - |v|)| \leq \gamma$.

Lemma 1. *Let $R, S \subseteq \Gamma^* \times \Gamma^*$ have bounded length-difference. Then R is ℓ-automatic if and only if R is r-automatic. Moreover, if R and S are α-automatic, then $R \cdot S = \{(st, uv) \mid (s,u) \in R, (t,v) \in S\}$ is α-automatic as well.*

Let $\mathcal{M}$ be a monoid. A triple (Γ, R, L) is an *$\alpha\beta$-automatic presentation for $\mathcal{M}$* if: (i) R is a semi-Thue system over the finite alphabet Γ such that $\mathcal{M} \simeq \Gamma^*/R$, (ii) $L \subseteq \Gamma^*$ is a regular language such that $\pi_R \restriction L$ maps L *surjectively* to $\mathcal{M}$, (iii) the relation $\{(u,v) \in L \times L \mid \pi_R(u) = \pi_R(v)\}$ is α-automatic, and (iv) if $\beta = \ell$ (resp. $\beta = r$), then the relation $\{(u,v) \in L \times L \mid \pi_R(au) = \pi_R(v)\}$ (resp. $\{(u,v) \in L \times L \mid \pi_R(ua) = \pi_R(v)\}$) is α-automatic for every $a \in \Gamma$. The monoid $\mathcal{M}$ is *$\alpha\beta$-automatic* if there exists an $\alpha\beta$-automatic presentation for $\mathcal{M}$. Thus, we have four different basic notions of automaticity. Whereas for groups all these four variants are equivalent [13] (which allows to speak of automatic groups), one obtains 15 different notions of automaticity for monoids by combining the four basic variants of $\alpha\beta$-automaticity [13, 14]. For our lower bounds we will mostly work with the strongest possible notion of automaticity, i.e., simultaneous $\alpha\beta$-automaticity for all $\alpha, \beta \in \{\ell, r\}$ (which includes the notion of biautomaticity from the theory of automatic groups, see [11]). Note that a $\alpha\beta$-automatic monoid is by definition finitely generated. Various classes of semi-Thue systems that present automatic monoids can be found in [26].

4 Complexity of the word problem

The word problem for an automatic group can be solved in quadratic time [11]. Moreover, the same algorithm also works for $\alpha\beta$-automatic monoids [6]. Here we will show that P is also a lower bound for the monoid case.

Theorem 1. *There is a finite, length-lexicographic, and confluent semi-Thue system $R \subseteq \Gamma^* \times \Gamma^*$ such that the word problem for Γ^*/R is P-complete and $(\Gamma, R, \mathrm{IRR}(R))$ is an $\alpha\beta$-automatic presentation for Γ^*/R for all $\alpha, \beta \in \{\ell, r\}$.*

Proof. We start with a fixed deterministic Turing machine S that accepts a P-complete language. Let $p(n)$ be a polynomial such that S terminates on an input $w \in L(S)$ after exactly $p(|w|)$ steps (this exact time bound can be easily enforced). We may assume that the tape is restricted to size $p(|w|)$. It is straight forward to simulate S by a new deterministic Turing machine T that operates in a sequence of complete left/right sweeps over the whole tape (of size $p(|w|)$). During a right sweep, the head runs from the left tape end to the right tape end in a sequence of right moves. When reaching the right tape end, the head turns back and starts a left sweep. Let Σ be the tape alphabet of T, Q be the set of states, q_0 be the initial state, and q_f be the final state. With $¢ \in \Sigma$ we denote the blank symbol. We write $qa \Rightarrow_T bp$ ($aq \Rightarrow_T pb$), in case T writes b, moves right (left), and enters state p, when reading a in state q. The machine T terminates (and accepts its input) if and only if it finally reaches the final state q_f. Thus, T cannot make any transitions out of q_f. Moreover, we may assume that the tape is blank and that the tape head is scanning the first cell when T terminates in state q_f. Define $\Gamma = \Sigma \cup \overline{\Sigma} \cup Q \cup \overline{Q} \cup \{\$, \overline{\$}\}$, where $\overline{\Sigma} = \{\overline{a} \mid a \in \Sigma\}$ is a disjoint copy of Σ and similarly for $\overline{Q}$. Let R be the following semi-Thue system over Γ:

$$\boxed{\begin{array}{llll} qa \to \overline{b}p & \text{if } qa \Rightarrow_T bp & \overline{a}\,\overline{q} \to \overline{p}b & \text{if } aq \Rightarrow_T pb \\ q\$ \to \overline{q} & \text{for all } q \in Q & \overline{\$}\overline{q} \to q & \text{for all } q \in Q \end{array}}$$

R is length-lexicographic and confluent (because T is deterministic). Next, let $w \in \Sigma^*$ be an arbitrary input for T and let $m = p(|w|)$. Then w is accepted by T if and only if $\overline{\$}^m q_0 w ¢^{m-|w|} \$^m \xrightarrow{*}_R q_f ¢^m$ if and only if $\overline{\$}^m q_0 w ¢^{m-|w|} \$^m \xleftrightarrow{*}_R q_f ¢^m$. Thus, the word problem for Γ^*/R is P-hard.

Next, we show that for all $\alpha, \beta \in \{\ell, r\}$, $(\Gamma, R, \mathrm{IRR}(R))$ is an $\alpha\beta$-automatic presentation for Γ^*/R (then in particular, the word problem for Γ^*/R belongs to P). Due to the symmetry of R, we can restrict to $\beta = \ell$. Thus, we have to show that the relation $E_c = \{(u, v) \in \mathrm{IRR}(R) \times \mathrm{IRR}(R) \mid cu \xrightarrow{*}_R v\}$ is α-automatic for all $c \in \Gamma$ and $\alpha \in \{\ell, r\}$. Note that all relations that appear in the following consideration have bounded length-difference. This allows to make use of Lemma 1. First, note that the following relations are α-automatic:

$$\begin{aligned} A_q &= \{(u, \overline{v}p) \mid p \in Q, u \in \Sigma^*, \overline{v} \in \overline{\Sigma}^*, qu \xrightarrow{*}_R \overline{v}p\} \\ B_q &= \{(\overline{u}, \overline{p}v) \mid \overline{p} \in \overline{Q}, \overline{u} \in \overline{\Sigma}^*, v \in \Sigma^*, \overline{u}\,\overline{q} \xrightarrow{*}_R \overline{p}v\} \end{aligned}$$

The relation A_q (resp. B_q) describes a single right (resp. left) sweep over the whole tape started in state q, which is just a rational transduction. Since α-automatic relations are closed under composition, the relation

$$C_q = \{(u\$, \overline{p}v) \mid \overline{p} \in \overline{Q}, u, v \in \Sigma^*, qu\$ \xrightarrow{*}_R \overline{p}v\}$$

is α-automatic as well. Now the α-automaticity of the relations E_c for $c \in \Gamma$ follows easily: For $c \in \overline{Q} \cup \Sigma \cup \{\$\}$ we have $E_c = \{(u, cu) \mid u \in \mathrm{IRR}(R)\}$, which is clearly α-automatic. For $c = \overline{a} \in \overline{\Sigma}$ and $c = q \in Q$, respectively, we have:

$$\begin{aligned} E_{\overline{a}} &= \{(u, \overline{a}u) \mid u \in \mathrm{IRR}(R), u \notin \overline{Q}\Gamma^*\} \cup \\ &\quad \{(\overline{q}u, \overline{p}bu) \mid u \in \mathrm{IRR}(R), \overline{q}, \overline{p} \in \overline{Q}, b \in \Sigma, (\overline{a}\,\overline{q}, \overline{p}b) \in R\} \\ E_q &= \{(uw, vw) \mid (u, v) \in A_q, w \in \mathrm{IRR}(R), w \notin (\Sigma \cup \{\$\})\Gamma^*\} \cup \\ &\quad \{(uw, vw) \mid (u, v) \in C_q, w \in \mathrm{IRR}(R)\}. \end{aligned}$$

Finally, $E_{\overline{\$}} = \{(u, \overline{\$}u) \mid u \in \mathrm{IRR}(R), u \notin \overline{Q}\Gamma^*\} \cup \bigcup_{q \in Q} \{(\overline{q}u, v) \mid (u, v) \in E_q\}$. This concludes the proof of the α-automaticity of the relations E_c. □

Corollary 1. *There exists a fixed finitely presented monoid with a P-complete word problem, which is simultaneously $\alpha\beta$-automatic for all $\alpha, \beta \in \{\ell, r\}$.*

It is open, whether there exists an automatic group (or even cancellative automatic monoid) with a P-complete word problem. An important subclass of the class of automatic groups is the class of hyperbolic groups, which are defined via a geometric hyperbolicity condition on the Cayley-graph. The precise definition is not important for the purpose of this paper. In [4], Cai has shown that for every hyperbolic group the word problem belongs to the parallel complexity class NC^2, which is the class of all problems that can be recognized by a polynomial size family of Boolean circuits of depth $O(\log^2(n))$, where only Boolean gates of fan-in at most 2 are allowed. Cai also asked, whether the upper bound of NC^2 can be improved. Using known results from formal language theory, we will show that for every hyperbolic group the word problem belongs to $\mathrm{LOGCFL} \subseteq \mathrm{NC}^2$, which is the class of all problems that are logspace reducible to a context-free language [32]. For alternative characterizations of LOGCFL see [27, 33].

Theorem 2. *The word problem for every fixed hyperbolic group is in LOGCFL.*

Proof. By [8], a group $\mathcal{G}$ is hyperbolic if and only if $\mathcal{G} \cong \Gamma^*/R$, where R is finite, length-reducing, and $L := \{s \in \Gamma^* \mid s \xrightarrow{*}_R \varepsilon\} = \{s \in \Gamma^* \mid s \overset{*}{\leftrightarrow}_R \varepsilon\}$. Since $\mathcal{G}$ is a group, the word problem for $\mathcal{G}$ is logspace reducible to L. Since R is length-reducing, L is growing context-sensitive, i.e., it can be generated by a grammar, where every production is strictly length-increasing. Since every fixed growing context-sensitive language belongs to LOGCFL [9], the theorem follows. $\square$

In [10, 15], hyperbolic groups were generalized to *hyperbolic monoids*. It is not clear whether Theorem 2 can be extended to hyperbolic monoids. It is also open, whether the upper bound of LOGCFL from Theorem 2 can be further improved, for instance to LOGDCFL, which is the class of all problems that are logspace reducible to a deterministic context-free language [32]. For another class of automatic monoids, we can precisely characterize the complexity of the word problem using LOGDCFL: A semi-Thue system R over the alphabet Γ is called *left-basic* [29] if: (i) if $\ell \in \mathrm{dom}(R)$, $r \in \mathrm{ran}(R)$ and $r = u\ell v$ then $u = v = \varepsilon$ and (ii) if $\ell \in \mathrm{dom}(R)$, $r \in \mathrm{ran}(R)$, $ur = \ell v$, and $|\ell| > |u|$, then $v = \varepsilon$. Condition (i) means that a right-hand side does not strictly contain a left-hand side. Condition (ii) means that the following kind of overlapping is not allowed:

<table>
<tr><td>u</td><td colspan="2">$r \in \mathrm{ran}(R)$</td></tr>
<tr><td colspan="2">$\ell \in \mathrm{dom}(R)$</td><td>$v \neq \varepsilon$</td></tr>
</table>

Let us define the suffix-rewrite relation $\twoheadrightarrow_R$ by $s \twoheadrightarrow_R t$ if and only if $s = u\ell$ and $t = ur$ for some $u \in \Gamma^*$ and $(\ell, r) \in R$. The following lemma is obvious:

Lemma 2. *If R is left-basic, then for every $s \in \mathrm{IRR}(R)$ and $a \in \Gamma$ we have $sa \xrightarrow{*}_R t$ if and only if $sa \overset{*}{\twoheadrightarrow}_R t$.*

Left-basic semi-Thue systems generalize monadic semi-Thue systems. Systems that are finite, monadic, and confluent present monoids that are simultaneously rr- and $\ell\ell$-automatic, but in general neither $r\ell$- nor ℓr-automatic [26]. Using arguments similar to those from [26], we can show that for a finite, terminating, confluent, and left-basic semi-Thue system R over an alphabet Γ, the monoid Γ^*/R is rr-automatic.

Theorem 3. *The following problem is in LOGDCFL:*

INPUT: A finite, terminating, confluent, and left-basic semi-Thue system R over an alphabet Γ, and two words $s, t \in \Gamma^*$

QUESTION: $s \overset{*}{\leftrightarrow}_R t$?

Moreover, there exists a finite, length-reducing, confluent, and left-basic semi-Thue system R over an alphabet Γ such that the word problem for Γ^*/R is LOGDCFL-complete.

Proof. Note that the upper bound in the first statement holds in a uniform setting, i.e., the semi-Thue system is part of the input. In order to prove this upper bound, we will use a machine-based characterization of LOGDCFL: A logspace bounded deterministic AuxPDA is a deterministic pushdown automaton that has an auxiliary read-write tape of size $\mathcal{O}(\log(n))$ (where n is the input size). A problem belongs to LOGDCFL if and only if it can be decided by a logspace bounded deterministic AuxPDA that moreover works in polynomial time [32].

Now, let us assume that the input consists of a tuple (Γ, R, s, t), where R is a finite, terminating, confluent, and left-basic semi-Thue system over the alphabet Γ and $s, t \in \Gamma^*$. Let n be the length of the binary coding of this input. We will construct a logspace bounded deterministic AuxPDA that checks in polynomial time, whether $\mathrm{NF}_R(s) = \mathrm{NF}_R(t)$. For this, we will first show how to calculate $\mathrm{NF}_R(s)$ on a deterministic AuxPDA in logspace and polynomial time. The basic idea of how to do this appeared many times in the literature, see e.g. [3, Thm. 4.2.7]. The only slight complication in our situation results from the fact that the semi-Thue system R belongs to the input. To overcome this, we need the logspace bounded auxiliary store of our AuxPDA. The correctness of the following procedure follows from Lemma 2. Our algorithm for computing $\mathrm{NF}_R(s)$ works in stages. At the beginning of a stage the pushdown contains a word from $\mathrm{IRR}(R)$ and the auxiliary store contains a pointer to a position i in the input word s. Note that a symbol $a \in \Gamma$ can be represented as a bit string of length $\mathcal{O}(\log(n))$, thus the pushdown content is a sequence of blocks of length $\mathcal{O}(\log(n))$, where every block represents a symbol from Γ. The stage begins by pushing the i-th symbol of s onto the pushdown (which is a bit string of length $\mathcal{O}(\log(n))$) and incrementing the pointer to position $i + 1$ in s. Now we have to check whether the pushdown content is of the form $\Gamma^*\mathrm{dom}(R)$. For this we have to scan every left-hand side of R using a second pointer to the input. Every $\ell \in \mathrm{dom}(R)$ is scanned in reverse order and thereby compared with the top of the push-down. During this phase, symbols are popped from the pushdown. If it turns out that the left-hand side that is currently scanned is not a suffix of the pushdown content, then these symbols must be "repushed". This can be done, since the suffix of the pushdown content that was popped so far is a suffix of the currently scanned left-hand side $\ell \in \mathrm{dom}(R)$, which is still available on the read-only input tape. If a left-hand side ℓ is found on top of the pushdown, then the corresponding right-hand side is pushed on the pushdown and we try to find again a left-hand side on top of the pushdown. If finally no left-hand side matches a suffix of the pushdown content, then we know that the pushdown content belongs to $\mathrm{IRR}(R)$ and we can proceed with the next stage. Finally, if the first pointer has reached the end of the input word s (or more precisely points to the first position following s), then the pushdown content equals $\mathrm{NF}_R(s)$.

Claim: In the above procedure, after the i-th stage the pushdown has length at most $i \cdot \alpha$, where $\alpha = \max(\{1\} \cup \{|r| \mid r \in \mathrm{ran}(R)\})$. Moreover, every stage needs only polynomial time.

The first statement can be shown by induction on i. Since R is left-basic, it follows that if w is the pushdown content at the end of the $(i-1)$-th stage, then the pushdown content at the end of the i-th stage either belongs to $w\Gamma$ or is of the form ur for some $r \in \mathrm{ran}(R)$ and some prefix u of w. Moreover, the i-th stage simulates at most $|w| \cdot |R|$ rewrite steps of R.

In order to check whether $\mathrm{NF}_R(s) = \mathrm{NF}_R(t)$, we have to solve one more problem: If we would calculate $\mathrm{NF}_R(t)$ in the same way as above, then the pushdown would finally contain the word $\mathrm{NF}_R(s)\mathrm{NF}_R(t)$. But now there seems to be no way of checking, whether $\mathrm{NF}_R(s) = \mathrm{NF}_R(t)$. Thus, we have to apply another strategy. Note that for a fixed binary coded number $1 \leq i \leq \alpha \cdot |s|$, it is easy to modify our algorithm for calculating $\mathrm{NF}_R(s)$ such that some specified auxiliary storage cell S contains always the i-th symbol of the pushdown content (or some special symbol if the pushdown is shorter than i). For this we have to store the length of the pushdown, for which we need only space $\mathcal{O}(\log(n))$. Moreover, also S only needs space $\mathcal{O}(\log(n))$. Thus, at the end of our modified algorithm for computing $\mathrm{NF}_R(s)$, S contains the symbol $\mathrm{NF}_R(s)[i]$ (the i-th symbol of $\mathrm{NF}_R(s)$) or some special symbol in case $|\mathrm{NF}_R(s)| < i$. Next, we flush the pushdown and repeat the same procedure with the other input word t and the same i, using another storage cell T. In this way we can check, whether $\mathrm{NF}_R(s)[i] = \mathrm{NF}_R(t)[i]$. Finally, we repeat this step for every $1 \leq i \leq \max\{\alpha \cdot |s|, \alpha \cdot |t|\}$. The latter bound is the maximal pushdown-length that may occur, which follows from the above claim. Note that also i needs only space $\mathcal{O}(\log(n))$. This concludes the description of our LOGDCFL-algorithm.

It remains to construct a finite, length-reducing, confluent, and left-basic semi-Thue system R such that the corresponding word problem is LOGDCFL-hard. In [32], Sudborough has shown that there exists a fixed deterministic context-free language $L \subseteq \Sigma^*$ with a LOGDCFL-complete membership problem. Let $\mathcal{A} = (Q, \Delta, \Sigma, \delta, q_0, \bot)$ be a deterministic pushdown automaton with $L = L(\mathcal{A})$, where Q is the set of states, $q_0 \in Q$ is the initial state, Δ is the pushdown alphabet, $\bot \in \Delta$ is the bottom symbol, and $\delta : \Delta \times Q \times \Sigma \to \Delta^* \times Q$ is the transition function. By [32, Lem. 7] we may assume that $\mathcal{A}$ makes no ε-moves and that $\mathcal{A}$ accepts L by empty store in state q_0. Let $m = \max\{|\gamma| \mid \delta(A, q, a) = (\gamma, p),\ q, p \in Q, A \in \Delta, a \in \Sigma\}$; thus, m is the maximal length of a sequence that is pushed on the pushdown in one step. Let $\# \notin \Delta \cup Q \cup \Sigma$ be an additional symbol and let $\Gamma = \Delta \cup Q \cup \Sigma \cup \{\#\}$. Define the semi-Thue system R by $R = \{Aqa^m\# \to \gamma p \mid \delta(A, q, a) = (\gamma, p)\}$; it is length-reducing, confluent, and left-basic. Moreover, if $h : \Sigma^* \to (\Sigma \cup \{\#\})^*$ denotes the homomorphism defined by $h(a) = a^m\#$, which can be computed in logspace, then $w \in L$ if and only if $\bot q_0 h(w) \xrightarrow{*}_R q_0$ if and only if $\bot q_0 h(w) \xleftrightarrow{*}_R q_0$. $\square$

5 Cayley-graphs

Let $\mathcal{M}$ be a monoid, which is finitely generated by Γ, and let $\circ$ denote the monoid operation of $\mathcal{M}$. The *right Cayley-graph* of $\mathcal{M}$ wrt. Γ is the Γ-labeled directed graph $\mathcal{C}(\mathcal{M}, \Gamma) = (\mathcal{M}, (\{(u, v) \mid u \circ a = v\})_{a\in\Gamma})$. Thus, edges are defined via multiplication with generators on the right. The graph that is defined analogously via multiplication with generators on the left is called the *left Cayley-graph* of $\mathcal{M}$ wrt. Γ. In the following, we will always refer to the *right* Cayley-graph when just speaking of the Cayley-graph. Cayley-graphs were mainly investigated for groups, in particular they play an important role in combinatorial group theory [23] (see also the survey of Schupp [28]). Combinatorial properties of Cayley-graphs of monoids are studied in [17]. In [30, 31], Cayley-graphs of automatic monoids are investigated. The work of Calbrix and Knapik on Thue-specifications [5, 19] covers Cayley-graphs of monoids that are presented by terminating and confluent semi-Thue systems as a special case.

In [21], an investigation of Cayley-graphs from a logical point of view was initiated. For a given Cayley-graph $\mathcal{C} = (\mathcal{M}, (E_a)_{a\in\Gamma})$ we consider first-order formulas over the structure $\mathcal{C}$. Atomic formulas are of the form $x = y$ and $E_a(x, y)$, (there is an a-labeled edge from x to y) where x and y are variables that range over the monoid $\mathcal{M}$. Instead of $(x, y) \in E_a$ we write $x \circ a = y$, or briefly $xa = y$. First-order formulas are built from atomic formulas using Boolean connectives and quantifications over variables. The notion of a free variable is defined as usual. A first-order formula without free variables is called a *first-order sentence*. For a first-order sentence φ, we write $\mathcal{C} \models \varphi$ if φ evaluates to true in $\mathcal{C}$. The *first-order theory* of the Cayley-graph $\mathcal{C}$, denoted by $\mathrm{FOTh}(\mathcal{C})$, is the set of all first-order sentences φ such that $\mathcal{C} \models \varphi$. For a detailed introduction into first-order logic over arbitrary structures see [12].

If the monoid $\mathcal{M}$ is finitely generated both by Γ and Σ, then $\mathrm{FOTh}(\mathcal{C}(\mathcal{M}, \Gamma))$ is logspace reducible to $\mathrm{FOTh}(\mathcal{C}(\mathcal{M}, \Sigma))$ and vice versa [20]. Thus, analogously to word problems, the decidability (resp. complexity) status of the first-order theory of a Cayley-graph does not depend on the chosen set of generators. From the definition of an αr-automatic monoid $\mathcal{M}$ it follows immediately that $\mathcal{C}(\mathcal{M}, \Gamma)$ is an automatic graph in the sense of [1, 18] (but the converse is even false for groups, see e.g. [2]). Thus, since every automatic graph has a decidable first-order theory [18], $\mathrm{FOTh}(\mathcal{C}(\mathcal{M}, \Gamma))$ is decidable in case $\mathcal{M}$ is αr-automatic ($\alpha = \ell$ or $\alpha = r$). If $\mathcal{M}$ is an $\alpha\ell$-automatic monoid ($\alpha = \ell$ or $\alpha = r$), then the first-order theory of the left Cayley-graph of $\mathcal{M}$ is decidable.

A problem is elementary decidable if it can be solved in time $\mathcal{O}(2^{\cdot^{\cdot^{2^n}}})$, where the height of this tower of exponents is constant. By [1], there exists an automatic graph with a nonelementary first-order theory. This complexity is already realized by Cayley-graphs of automatic monoids:

Theorem 4. *There is a finite, length-lexicographic, and confluent semi-Thue system $R \subseteq \Gamma^* \times \Gamma^*$ such that $(\Gamma, R, \mathrm{IRR}(R))$ is an $\alpha\beta$-automatic presentation for Γ^*/R for all $\alpha, \beta \in \{\ell, r\}$ and $\mathrm{FOTh}(\mathcal{C}(\Gamma^*/R, \Gamma))$ is nonelementary.*

Proof. Let $\Gamma = \{a, b, \bar{a}, \bar{b}, \$_1, \$_2, \$_a\}$ and let the semi-Thue system R over Γ consist of the following rules, where $c \in \{a, b\}$:

$c\,\$_1 \to \$_1 c$	$c\,\$_2 \to \$_2 c$	$\bar{a}\,\$_a \to a$	$c\,\$_a \to \$_a c$
$\bar{c}\,\$_1 \to c$	$\bar{c}\,\$_2 \to \$_1 \bar{c}$	$\bar{b}\,\$_a \to \$_a \bar{b}$	

R is length-lexicographic and confluent. Arguments similar to those from the proof of Theorem 1 show that $(\Gamma, R, \mathrm{IRR}(R))$ is an $\alpha\beta$-automatic presentation of $\mathcal{M} = \Gamma^*/R$. Let $\mathcal{C} = \mathcal{C}(\mathcal{M}, \Gamma)$. It remains to show that $\mathrm{FOTh}(\mathcal{C})$ is not elementary decidable. For this we reduce the *first-order theory of finite words* over $\{a, b\}$ to $\mathrm{FOTh}(\mathcal{C})$. The former theory is defined as follows: A word $w = a_1 a_2 \cdots a_n \in \{a, b\}^*$ of length n is identified with the relational structure $S_w = (\{1, \ldots, n\}, <, Q_a)$, where $<$ is the usual order on natural numbers and Q_a is the unary predicate $\{i \in \{1, \ldots, n\} \mid a_i = a\}$. Then the first-order theory of finite words over $\{a, b\}$ consists of all first-order sentences ϕ that are built up from the atomic formulas $x < y$ and $Q_a(x)$ such that $S_w \models \phi$ for every word $w \in \{a, b\}^*$. It is known that the first-order theory of finite words is decidable but not elementary, see e.g. [7, Example 8.1] for a simplified proof.

For our reduction first notice that $\mathrm{IRR}(R) = \{\$_1, \$_2, \$_a\}^* \{a, b, \bar{a}, \bar{b}\}^*$. Hence, the latter set can be identified with the monoid $\mathcal{M}$. For $x \in \mathrm{IRR}(R)$ we have $x \in \{\$_1, \$_2, \$_a\}^* \{a, b\}^*$ if and only if $x\$_2\$_1 \neq x\$_1\$_1$ in $\mathcal{M}$. This allows us to represent all words from $\{a, b\}^*$ in $\mathcal{C}$. The fact that a word $w \in \{a, b\}^*$ is represented by infinitely many nodes of $\mathcal{C}$, namely by all elements from $\{\$_1, \$_2, \$_a\}^* w$ does not cause any problems; it is only important that every word $w \in \{a, b\}^*$ is represented at least once. In the sequel let us fix $x = vw$ with $v \in \{\$_1, \$_2, \$_a\}^*$ and $w \in \{a, b\}^*$. The set of all positions within the word w is in one-to-one correspondence with the set of all y such that $y\$_1 = x$ in $\mathcal{M}$: the latter holds if and only if $\exists w_1, w_2 \in \{a, b\}^* \, \exists c \in \{a, b\} : w = w_1 c w_2$ and $y = v w_1 \bar{c} w_2$. Thus, we can quantify over positions of the word w by quantifying in $\mathcal{C}$ over all those nodes y such that $y\$_1 = x$ in $\mathcal{M}$. Next, assume that $y = v w_1 \bar{c} w_2$ and $w = w_1 c w_2$, i.e., y represents the position $|w_1| + 1$ of w. Then $c = a$ if and only if $y\$_a = x$ in $\mathcal{M}$; thus we can express that a position is labeled with the symbol a. It remains to express that a position is smaller than another one. Assume that $y = v w_1 \bar{c} w_2$, $y' = v w_1' \bar{d} w_2'$, $w_1 c w_2 = w_1' d w_2' = w$, and $w_1 \neq w_1'$, i.e., the two positions represented by y and y' are different. Then $|w_1| < |w_1'|$ if and only if $\exists z \in \mathcal{M} : z\$_1 = y \wedge z\$_2 = y'$ in $\mathcal{M}$.

From the preceding discussion it follows that for every first-order sentence ψ over the signature $(<, Q_a)$ we can construct in polynomial time a first-order formula $\phi(x)$ over the Cayley-graph $\mathcal{C}$ such that ψ belongs to the first-order theory of finite words if and only if $\mathcal{C} \models \forall x : \phi(x)$. This proves the theorem. □

Corollary 2. *There exists a finitely presented monoid $\mathcal{M}$ such that $\mathcal{M}$ is simultaneously $\alpha\beta$-automatic for all $\alpha, \beta \in \{\ell, r\}$ and $\mathrm{FOTh}(\mathcal{C}(\mathcal{M}, \Gamma))$ is not elementary decidable.*

Since the word problem of an automatic group can be solved in time $\mathcal{O}(n^2)$, the results from [20] imply that the nonelementary lower bound from Corollary 2 cannot be realized by an automatic group. This fact even holds for automatic monoids of *finite geometric type*: A finitely generated monoid $\mathcal{M}$ has finite geometric type if for some (and hence every) finite generating set Γ, the Cayley-graph $\mathcal{C}(\mathcal{M}, \Gamma)$ has bounded degree [30], i.e., the number of neighbors of any node is bounded by a fixed constant. Every *right-cancellative monoid* has finite geometric type, but for instance the bicyclic monoid $\{a, b\}^*/\{(ab, \varepsilon)\}$ is not right-cancellative but has finite geometric type. Since the Cayley-graph of an αr-automatic monoid of finite geometric type is an automatic graph of bounded degree, and the first-order theory of every automatic graph of bounded degree belongs to $\mathrm{DSPACE}(2^{2^{2^{\mathcal{O}(n)}}})$ [22], we obtain:

Theorem 5. *Let $\mathcal{M}$ be an αr-automatic monoid ($\alpha \in \{r, \ell\}$) of finite geometric type. Then* $\mathrm{FOTh}(\mathcal{C}(\mathcal{M}, \Gamma))$ *belongs to* $\mathrm{DSPACE}(2^{2^{2^{\mathcal{O}(n)}}})$.

We conclude this paper with an undecidability result for automatic monoids. Note that for an αr-automatic monoid $\mathcal{M}$ ($\alpha \in \{r, \ell\}$) it is decidable whether for given $u, v \in \mathcal{M}$ there exists $x \in \mathcal{M}$ such that $xu = v$ in $\mathcal{M}$, because this is a first-order property of the Cayley-graph. On the other hand, the reverse question ($\exists x : ux = v$, i.e., reachability in the Cayley-graph) is undecidable in general:

Theorem 6. *There exists a finitely presented monoid $\mathcal{M}$ that is simultaneously ℓr- and rr-automatic such that for given $u, v \in \mathcal{M}$ it is undecidable whether $\exists x \in \mathcal{M} : ux = v$ in $\mathcal{M}$.*

The proof of this result uses the same techniques as the proof of Theorem 1.

References

1. A. Blumensath and E. Grädel. Automatic structures. In *Proc. LICS'2000*, pages 51–62. IEEE Computer Society Press, 2000.
2. A. Blumensath and E. Grädel. Finite presentations of infinite structures: Automata and interpretations. In *Proc. CiAD 2002*, 2002.
3. R. V. Book and F. Otto. *String–Rewriting Systems*. Springer, 1993.
4. J.-y. Cai. Parallel computation over hyperbolic groups. In *Proc. STOC 92*, pages 106–115. ACM Press, 1992.
5. H. Calbrix and T. Knapik. A string-rewriting characterization of Muller and Schupp's context-free graphs. In *Proc. FSTTCS 1998*, LNCS 1530, pages 331–342. Springer, 1998.
6. C. M. Campbell, E. F. Robertson, N. Ruškuc, and R. M. Thomas. Automatic semigroups. *Theor. Comput. Sci.*, 250(1-2):365–391, 2001.
7. K. J. Compton and C. W. Henson. A uniform method for proving lower bounds on the computational complexity of logical theories. *Ann. Pure Appl. Logic*, 48:1–79, 1990.
8. M. Coornaert, T. Delzant, and A. Papadopoulos. *Géométrie et théorie des groupes*. Number 1441 in Lecture Notes in Mathematics. Springer, 1990.

9. E. Dahlhaus and M. K. Warmuth. Membership for growing context-sensitive grammars is polynomial. *J. Comput. Syst. Sci.*, 33:456–472, 1986.
10. A. Duncan and R. H. Gilman. Word hyperbolic semigroups. *Math. Proc. Cambridge Philos. Soc.*, 136:513–524, 2004.
11. D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, and W. P. Thurston. *Word processing in groups.* Jones and Bartlett, Boston, 1992.
12. W. Hodges. *Model Theory.* Cambridge University Press, 1993.
13. M. Hoffmann. *Automatic semigroups.* PhD thesis, University of Leicester, Department of Mathematics and Computer Science, 2000.
14. M. Hoffmann and R. M. Thomas. Notions of automaticity in semigroups. *Semigroup Forum*, 66(3):337–367, 2003.
15. M. Hoffmann and D. Kuske and F. Otto and R. M. Thomas. Some relatives of automatic and hyperbolic groups. In *Workshop on Semigroups, algorithms, automata and languages 2001*, pages 379–406. World Scientific, 2002.
16. J. F. P. Hudson. Regular rewrite systems and automatic structures. In *Semigroups, Automata and Languages*, pages 145–152. World Scientific, 1998.
17. A. V. Kelarev and S. J. Quinn. A combinatorial property and Cayley graphs of semigroups. *Semigroup Forum*, 66(1):89–96, 2003.
18. B. Khoussainov and A. Nerode. Automatic presentations of structures. In *LCC: International Workshop on Logic and Computational Complexity*, LNCS 960, pages 367–392, Springer 1994.
19. T. Knapik and H. Calbrix. Thue specifications and their monadic second-order properties. *Fundam. Inform.*, 39:305–325, 1999.
20. D. Kuske and M. Lohrey. Logical aspects of Cayley-graphs: the group case. to appear in *Ann. Pure Appl. Logic.*
21. D. Kuske and M. Lohrey. Decidable theories of Cayley-graphs. In *Proc. STACS 2003*, LNCS 2607, pages 463–474. Springer, 2003.
22. M. Lohrey. Automatic structures of bounded degree. In *Proc. LPAR 2003*, LNAI 2850, pages 344–358, Springer 2003.
23. R. C. Lyndon and P. E. Schupp. *Combinatorial Group Theory.* Springer, 1977.
24. D. E. Muller and P. E. Schupp. Groups, the theory of ends, and context-free languages. *J. Comput. Syst. Sci.*, 26:295–310, 1983.
25. D. E. Muller and P. E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theor. Comput. Sci.*, 37(1):51–75, 1985.
26. F. Otto and N. Ruškuc. Confluent monadic string-rewriting systems and automatic structures. *J. Autom. Lang. Comb.*, 6(3):375–388, 2001.
27. W. L. Ruzzo. Tree–size bounded alternation. *J. Comput. Syst. Sci.*, 21:218–235, 1980.
28. P. E. Schupp. Groups and graphs: Groups acting on trees, ends, and cancellation diagrams. *Math. Intell.*, 1:205–222, 1979.
29. G. Sénizergues. Formal languages and word-rewriting. In *Term Rewriting, French Spring School of Theoretical Computer Science*, LNCS 909, pages 75–94. Springer, 1993.
30. P. V. Silva and B. Steinberg. A geometric characterization of automatic monoids. Technical Report CMUP 2000-03, University of Porto, 2001.
31. P. V. Silva and B. Steinberg. Extensions and submonoids of automatic monoids. *Theor. Comput. Sci.*, 289:727–754, 2002.
32. I. H. Sudborough. On the tape complexity of deterministic context–free languages. *J. Assoc. Comput. Mach.*, 25(3):405–414, 1978.
33. H. Venkateswaran. Properties that characterize LOGCFL. *J. Comput. Syst. Sci.*, 43:380–404, 1991.