RAIRO-Theor. Inf. Appl. 53 (2019) 1–17
https://doi.org/10.1051/ita/2018006

RAIRO - Theoretical Informatics and Applications
www.rairo-ita.org

THE COMPLEXITY OF WEAKLY RECOGNIZING MORPHISMS[☆]

Lukas Fleischer[1,*] and Manfred Kufleitner[2]

Abstract. Weakly recognizing morphisms from free semigroups onto finite semigroups are a classical way for defining the class of ω-regular languages, *i.e.*, a set of infinite words is weakly recognizable by such a morphism if and only if it is accepted by some Büchi automaton. We study the descriptional complexity of various constructions and the computational complexity of various decision problems for weakly recognizing morphisms. The constructions we consider are the conversion from and to Büchi automata, the conversion into strongly recognizing morphisms, as well as complementation. We also show that the fixed membership problem is NC^1-complete, the general membership problem is in L and that the inclusion, equivalence and universality problems are NL-complete. The emptiness problem is shown to be NL-complete if the input is given as a non-surjective morphism.

Mathematics Subject Classification. 20M35, 03D15, 68Q17, 68Q19

Received August 15, 2017. Accepted September 10, 2018.

1. Introduction

Büchi automata (BA) define the class of ω-regular languages. They were introduced by Büchi for deciding the monadic second-order theory of $(\mathbb{N}, <)$ [4]. Since then, ω-regular languages have become an important tool in formal verification, and many other automata models for this language class have been considered; see *e.g.* [18, 23]. Each automaton model has its merits and its disadvantages.

It is well-known that recognizing morphisms onto finite semigroups can be viewed as a two-sided algebraic counterpart to automata. When considered as acceptance models over infinite words, such morphisms come in two different flavors. Strongly recognizing morphisms admit efficient minimization and complementation, whereas weakly recognizing morphisms can be exponentially more succinct (but there is no minimal weak recognizer and there is no efficient complementation). The situation is similar to the behavior of deterministic and nondeterministic finite automata.

We first consider the descriptional complexity of various operations on weakly recognizing morphisms and conversions involving nondeterministic BA and strongly recognizing morphisms. In each case, we give asymptotically tight bounds. For the conversion of a BA into a weakly recognizing morphism, we give a lower bound which matches the naive upper bound. Our results are summarized in Table 1. For operations on BA, the number n

[☆]This work was supported by the DFG grants DI 435/5-2 and KU 2716/1-1.

Keywords and phrases: semigroup, regular language, automata, decision problem, descriptional complexity, computational complexity, operations

[1] FMI, University of Stuttgart, Stuttgart, Germany.
[2] Department of Computer Science, Loughborough University, Loughborough LE11 3TU, UK.

[*] Corresponding author: fleischer@fmi.uni-stuttgart.de

Article published by EDP Sciences

Table 1. Bounds for the descriptional complexity of various operations.

Operation	Lower bound	Upper bound
BA to weak recognition	2^{n^2} (Thm. 4.1)	2^{n^2} [17]
BA to weak recognition, binary alphabet	$2^{(n-1)^2/4}$ (Thm. 4.2)	2^{n^2} [17]
Weak recognition to BA	$(n-3)(n+1)/32$ (Thm. 5.1)	$n(n+1)$ [17]
Weak recognition to strong recognition	$n2^{n-1}$ (Prop. 6.1)	2^{n^2} [18]
Complementation of weak recognition	$n2^{n-1}$ (Prop. 6.1)	2^{n^2} [18]
Complementation for simple semigroups	$n2^{n-1}$ (Prop. 6.1)	$n2^n$ (Thm. 6.5)

refers to the number of states and for weakly or strongly recognizing morphisms, the number n is the cardinality of the underlying semigroup. The formulas in the table then describe lower and upper bounds on the maximum size of the objects resulting from the respective operations when the size of the input is n. For example, the standard conversion of weakly recognizing morphisms to BA [17] converts a semigroup of cardinality n to an automaton with at most $n(n+1)$ states, and we show later that there exist weakly recognizing morphisms to semigroups of cardinality n such that BA require at least $(n-3)(n+1)/32$ states to recognize the same language.

There are some similarities between recognizing morphisms over finite and over infinite words. Strong recognition is the natural counterpart to recognition for finite words. Nevertheless, in order to prove lower bounds for the conversion of BA to weakly recognizing morphisms, we first show that the bounds for converting nondeterministic finite automata to recognizing morphisms over finite words (with some limitations) also hold for the conversion of BA to weakly recognizing morphisms. We then use techniques of Sakoda and Sipser [20] and of Yan [25] to obtain tight bounds for the conversion of nondeterministic finite automata to recognizing morphisms. This step is similar to the work of Holzer and König [9]. To the best of our knowledge, our lower bound over finite words for the conversion of a nondeterministic finite automaton into a recognizing morphism is also a new result.

In the second part of this paper, we study the computational complexity of the most common language-theoretic decision problems, namely *membership*, *emptiness*, *universality*, *inclusion* and *equivalence*, when the languages are given as weakly recognizing morphisms and accepting sets.

One usually considers two variants of the membership problem: the *general membership problem* asks for a given ω-regular language L (in this case represented by a semigroup with multiplication table, a morphism and an accepting set) and an ultimately periodic word uv^ω, whether $uv^\omega \in L$. In the *fixed membership problem*, the semigroup, the recognizing morphism and the accepting set are fixed and the input only consists of the word uv^ω. We show that the fixed membership problem is in NC^1 and NC^1-complete if the semigroup contains some non-solvable group as a subsemigroup. Note that for NC^1-hardness, as in the case of finite words, the additional condition is necessary: if the fixed semigroup has "too much structure" (*e.g.* is commutative, aperiodic or even trivial), the membership problem can be solved more efficiently. The general membership problem is shown to be in L (as well as NC^1-hard by a trivial reduction from fixed membership). This resembles the situation over finite words.

Emptiness, universality, inclusion and equivalence are NL-complete when inputs are not restricted, *i.e.*, morphisms given as part of the input are also allowed to be non-surjective. Hardness partly stems from the fact that the problem of testing a morphism for surjectivity or testing whether an element is in the image of a morphism is NL-complete already which, in turn, essentially follows from NL-completeness of the *associative generation problem* first investigated by Jones *et al.* [12]. When restricting the input to surjective morphisms, the emptiness problem can be decided in constant space because the language recognized by a surjective weakly recognizing morphism is non-empty whenever the corresponding accepting set is non-empty. However, universality, inclusion and equivalence remain NL-complete. Our computational complexity results are summarized in Table 2.

TABLE 2. The computational complexity of decision problems.

Decision problem	Morphisms	Surjective morph.
Fixed membership	NC^1-complete (Prop. 7.1, Thm. 7.2)	NC^1-complete
General membership	L (Thm. 7.3), NC^1-hard (Prop. 7.1)	L (NC^1-hard)
Emptiness	NL-complete (Prop. 7.4)	trivial
Universality	NL-complete (Thm. 7.5)	NL-complete
Inclusion	NL-complete (Thm. 7.5)	NL-complete
Equivalence	NL-complete (Thm. 7.5)	NL-complete

2. Preliminaries

This section gives a brief overview of some basic definitions from the fields of formal languages, finite automata, semigroup theory and complexity theory. We refer to [18, 19] for more detailed introductions on the algebraic foundations of automata theory and to [16, 24] for introductions to computational complexity.

Words. Let A be a finite *alphabet*. The elements of A are called *letters*. A *finite word* is a sequence $a_1a_2\cdots a_n$ of letters of A and an *infinite word* is an infinite sequence $a_1a_2\cdots$. The empty word is denoted by ε. Given an infinite word $\alpha = a_1a_2\cdots$, we let $\inf(\alpha) \subseteq A$ denote the set of letters in α which occur infinitely often. An infinite word α is *ultimately periodic* if $\alpha = uv^\omega$ for some $u, v \in A^+$.

Let K be a set of finite words and let L be a set of infinite words. We set $KL = \{u\alpha \mid u \in K, \alpha \in L\}$, $K^n = \{u_1u_2\cdots u_n \mid u_i \in K\}$, $K^+ = \bigcup_{n\geqslant 1} K^n$ and $K^* = K^+ \cup \{\varepsilon\}$. Moreover, if $\varepsilon \notin K$ we define the *infinite iteration* $K^\omega = \{u_1u_2\cdots \mid u_i \in K\}$. A natural extension to $K \subseteq A^*$ is $K^\omega = (K \setminus \{\varepsilon\})^\omega \cup \{\varepsilon\}$, *i.e.*, the set K^ω contains the empty word but does not contain any other finite words.

Automata. A *finite automaton* is a 5-tuple $\mathcal{A} = (Q, A, \delta, I, F)$ where Q is a finite set of *states* and A is a finite alphabet. The *transition relation* δ is a subset of $Q \times A \times Q$ and its elements are called *transitions*. The sets I and F are subsets of Q and are called *initial states* and *final states*, respectively.

A *finite run* of a word $a_1a_2\cdots a_n$ on $\mathcal{A}$ is a sequence $q_0a_1q_1a_1\cdots q_{n-1}a_nq_n$ such that $q_0 \in I$ and $(q_i, a_{i+1}, q_{i+1}) \in \delta$ for all $i \in \{0, \ldots, n-1\}$. The run is said to *start* in q_0 and *end* in q_n. The word $a_1a_2\cdots a_n$ is the *label* of the run. A finite run is called *accepting* if it ends in a final state. A finite word u is said to be *accepted by* $\mathcal{A}$ if there exists an accepting finite run of u on $\mathcal{A}$ and the language *accepted by* $\mathcal{A}$ is the set of all finite words over A^* accepted by $\mathcal{A}$. It is denoted by $L_{\mathrm{NFA}}(\mathcal{A})$.

Analogously, an *infinite run* of a word $a_1a_2\cdots$ on $\mathcal{A}$ is an infinite sequence $q_0a_1q_1a_1\cdots$ such that $q_0 \in I$ and $(q_i, a_{i+1}, q_{i+1}) \in \delta$ for all $i \geqslant 0$. It is called *accepting* if $\inf(q_0q_1q_2\cdots) \cap F \neq \emptyset$. An infinite word α is said to be *Büchi-accepted by* $\mathcal{A}$ if there exists an accepting infinite run of α on $\mathcal{A}$. The language *Büchi-accepted by* $\mathcal{A}$ is the set of all infinite words Büchi-accepted by $\mathcal{A}$ and it is denoted by $L_{\mathrm{BA}}(\mathcal{A})$.

We use the term *run* for both finite and infinite runs if the reference is clear from the context. A language $L \subseteq A^*$ (resp. $L \subseteq A^\omega$) is *regular* (resp. ω*-regular*) if it is accepted (resp. Büchi-accepted) by some finite automaton.

Finite semigroups. A *semigroup morphism* is a mapping $h\colon S \to T$ between two (not necessarily finite) semigroups S and T such that $h(s)h(t) = h(st)$ for all $s, t \in S$. Since we do not consider morphisms of other objects, we use the term *morphism* synonymously. A *subsemigroup* of a semigroup S is a subset that is closed under multiplication. We say that a semigroup T *divides* a semigroup S if there exists a surjective morphism from a subsemigroup of S onto T.

Green's relations are an important tool in the study of semigroups. For the remainder of this subsection, let S be a finite semigroup. We let S^1 denote the monoid that is obtained by adding a new neutral element 1 to S.

For $s, t \in S$ let

$$\begin{array}{ll} s \mathrel{\mathcal{R}} t & \text{if there exist } q, q' \in S^1 \text{ such that } sq = t \text{ and } tq' = s, \\ s \mathrel{\mathcal{L}} t & \text{if there exist } p, p' \in S^1 \text{ such that } ps = t \text{ and } p't = s, \\ s \mathrel{\mathcal{J}} t & \text{if there exist } p, q, p', q' \in S^1 \text{ such that } psq = t \text{ and } p'tq' = s, \\ s \mathrel{\mathcal{H}} t & \text{if } s \mathrel{\mathcal{R}} t \text{ and } s \mathrel{\mathcal{L}} t. \end{array}$$

These relations are equivalence relations. The equivalence classes of $\mathcal{R}$ (resp. $\mathcal{L}$, $\mathcal{J}$, $\mathcal{H}$) are called *$\mathcal{R}$-classes* (resp. *$\mathcal{L}$-classes*, *$\mathcal{J}$-classes*, *$\mathcal{H}$-classes*). For $s \in S$, we denote the $\mathcal{R}$-class (resp. $\mathcal{L}$-class) of s by R_s (resp. L_s) and we let $S/\mathcal{R} = \{R_s \mid s \in S\}$ as well as $S/\mathcal{L} = \{L_s \mid s \in S\}$.

A semigroup is called *$\mathcal{J}$-trivial* if each of its $\mathcal{J}$-classes contains exactly one element. A semigroup is called *simple* if it consists of a single $\mathcal{J}$-class. In a finite simple semigroup, the relations $s \mathrel{\mathcal{R}} st \mathrel{\mathcal{L}} t$ hold for all $s, t \in S$. Moreover, each $\mathcal{H}$-class forms a group and all such groups are isomorphic [19]. We will also utilize the following lemma:

Lemma 2.1. *Let S be a finite simple semigroup and let $x, y, z \in S$ such that $y \mathrel{\mathcal{R}} z$. Then $xy = xz$ implies $y = z$.*

Proof. Suppose that $xy = xz$. Since S is simple, we have $y \mathrel{\mathcal{L}} xy$ and thus, there exists an element $p \in S^1$ such that $pxy = y$. Since $y \mathrel{\mathcal{R}} z$, there exists an element $q \in S^1$ with $yq = z$. It follows that $y = pxy = pxz = pxyq = yq = z$. ☐

Recognition by morphisms. Let $h\colon A^+ \to S$ be a morphism to a finite semigroup S. A pair (s, e) of elements of S is a *linked pair* if $se = s$ and $e^2 = e$. The set of all linked pairs of S is denoted by $F(S)$. For $s \in S$, we let $[s]_h = h^{-1}(s)$, and for a subset $P \subseteq S$, we use the notation $[P]_h$ to denote the set $\bigcup_{s \in P} [s]_h$. If h is understood from the context, we may skip the reference to the morphism in the subscript. A language $L \subseteq A^+$ is *recognized* by a morphism $h : A^+ \to S$ if L is a union of sets $[s_i]$ with $s_i \in S$. A language $L \subseteq A^\omega$ is *weakly recognized* by a morphism $h : A^+ \to S$ if it is a union of sets $[s_i][e_i]^\omega$ where (s_i, e_i) are linked pairs of S. A language $L \subseteq A^\omega$ is *strongly recognized* by a morphism $h : A^+ \to S$ if for all $s, t \in S$, $[s][t]^\omega \cap L \neq \emptyset$ implies $[s][t]^\omega \subseteq L$. It is easy to see that strong recognition implies weak recognition, see *e.g.* ([18], Thm. 2.2). Moreover, if a morphism strongly recognizes L, it also strongly recognizes its complement $A^\omega \setminus L$. By extension, we also say that a semigroup S recognizes (resp. weakly recognizes, strongly recognizes) a language L if there exists a morphism $h\colon A^+ \to L$ that recognizes (resp. weakly recognizes, strongly recognizes) L.

For a language $L \subseteq A^+ \cup A^\omega$, we have $u \equiv_L v$ if and only if

$$\begin{array}{l} (xuy)z^\omega \in L \Leftrightarrow (xvy)z^\omega \in L \text{ and} \\ z(xuy)^\omega \in L \Leftrightarrow z(xvy)^\omega \in L \end{array}$$

for all finite words $x, y, z \in A^*$. Keep in mind that $\varepsilon^\omega = \varepsilon$. The relation $\equiv_L$ is called the *syntactic congruence* of L. It is an extension of the classical syntactic congruence over finite words and was introduced by Arnold [1]. The congruence classes of $\equiv_L$ form the so-called *syntactic semigroup* $A^+/{\equiv_L}$ and the *syntactic morphism* $h_L\colon A^+ \to A^+/{\equiv_L}$ is the natural quotient map. If $L \subseteq A^*$ (resp. $L \subseteq A^\omega$) is regular (resp. ω-regular), the syntactic semigroup of L is finite and h_L recognizes (resp. strongly recognizes) the language L; see [1, 18].

Complexity classes. The class $\mathsf{ALOGTIME}$ consists of all languages recognizable by an alternating random access Turing machine with logarithmically bounded time. It is equivalent to the circuit complexity class NC^1; see [3, 5] for details. The class L (resp. NL) consists of all languages recognizable by a deterministic (resp. nondeterministic) Turing machine within logarithmically bounded space. The class coNL contains all languages whose complement is in NL. For $K, L \subseteq A^*$, a *log-time reduction* (resp. *log-space reduction*) from K to L is a log-time computable (resp. log-space computable) function $f\colon A^* \to A^*$ such that $w \in K \Leftrightarrow f(w) \in L$ for all

$w \in A^*$. We say that K is *log-time reducible* (resp. *log-space reducible*) to L if there exists a log-time (resp. log-space) reduction from K to L.

A language L is NC^1-*hard* if all languages from $\mathsf{NC}^1 = \mathsf{ALOGTIME}$ are log-time reducible to L. It is called NC^1-*complete* if, moreover, $L \in \mathsf{NC}^1 = \mathsf{ALOGTIME}$. A language L is NL-*hard* (resp. coNL-*hard*) if all languages in NL (resp. coNL) are log-space reducible to L. It is called NL-*complete* (resp. coNL-*complete*) if, moreover, $L \in \mathsf{NL}$ (resp. $L \in \mathsf{coNL}$). Since $\mathsf{NL} = \mathsf{coNL}$ by the Immerman-Szelepcsényi Theorem [10, 22], a language is coNL-complete if and only if it is NL-complete. One of the most prominent NL-complete problems is STCon which asks, for vertices s and t in a directed graph, whether there exists a path from s to t. There are many more NL-complete decision problems related to reachability in directed graphs. One of them is Cycle, formalized as follows.

Input:	A directed graph G
Question:	Does G contain a non-trivial directed cycle?

For proofs of NL-completeness of STCon and Cycle, we refer to [21]. Another similar problem is Strongly-Connected, defined as follows.

Input:	A directed graph G
Question:	Is G strongly connected?

Strongly-Connected is in NL since we can verify within logarithmic space that each vertex t is reachable from every other vertex s by running the STCon algorithm for each pair of vertices. NL-hardness follows from the fact that in a directed graph $G = (V, E)$, a node $t \in V$ is reachable from $s \in V$ if and only if the graph $G' = (V, E')$ with $E' = E \cup \{(v, s) \mid v \in V\} \cup \{(t, v) \mid v \in V\})$ is strongly connected. Therefore, STCon is log-space reducible to Strongly-Connected.

3. Lower bound techniques

3.1. Proving lower bounds for weakly recognizing morphisms

We first consider the general problem of proving lower bounds for the size of weakly recognizing semigroups for a given language L. In the case of recognizing morphisms over finite words and in the case of strongly recognizing morphisms, proving such bounds is easy since one only needs to compute the syntactic semigroup, which immediately yields a tight lower bound. In contrast, weakly recognizing morphisms do not admit minimal objects. However, it turns out that one can still use a relaxed version of Arnold's syntactic congruence.

We first prove a combinatorial lemma and then give the main result of this section.

Lemma 3.1. *Let $u, v \in A^+$ and let (s, e) be a linked pair. Then the word uv^ω is contained in $[s][e]^\omega$ if and only if there exists a factorization $v = v_1 v_2$ and powers $k \in \{1, 2, \ldots, |S|\}$ and $\ell \in \{1, 2, \ldots, 2|S| + 1\}$ such that ℓ is odd, $h(uv^k v_1) = s$ and $h(v_2 v^\ell v_1) = e$.*

Proof. Let $v = a_1 a_2 \cdots a_n$ with $n \geqslant 1$ and $a_i \in A$. If uv^ω is contained in $[s][e]^\omega$, there exists a factorization $uv^\omega = u' v'_1 v'_2 \cdots$ such that $h(u') = s$ and $h(v'_i) = e$ for all $i \geqslant 1$. Since u and v are finite words, there exist indices $j > i \geqslant 1$, powers $k, \ell \geqslant 1$ and a position $m \in \{1, \ldots, n\}$ such that $u' v'_1 v'_2 \cdots v'_{i-1} = uv^k a_1 a_2 \cdots a_m$ and $v'_i v'_{i+1} \cdots v'_j = a_{m+1} a_{m+2} \cdots a_n v^\ell a_1 a_2 \cdots a_m$. We set $v_1 = a_1 a_2 \cdots a_m$ and $v_2 = a_{m+1} a_{m+2} \cdots a_n$. Then $v_1 v_2 = v$,

$$h(uv^k v_1) = h(uv^k a_1 a_2 \cdots a_m) = h(u' v'_1 v'_2 \cdots v'_{i-1}) = se^{i-1} = s,$$
$$h(v_2 v^\ell v_1) = h(a_{m+1} a_{m+2} \cdots a_n v^\ell a_1 a_2 \cdots a_m) = h(v'_i v'_{i+1} \cdots v'_j) = e^{j-i+1} = e.$$

Because S is a finite semigroup, we have $s^i \in \left\{s, s^2, \ldots, s^{|S|}\right\}$ for all $s \in S$ and for all $i \geqslant 1$. Therefore, we can assume without loss of generality that $1 \leqslant k, \ell \leqslant |S|$. If ℓ is even, we can replace ℓ by $2\ell + 1$ since $h(v_2 v^{2\ell+1} v_1) = h(v_2 v^\ell v_1 v_2 v^\ell v_1) = e^2 = e$. The converse implication is trivial. □

Theorem 3.2. *Let $L \subseteq A^\omega$ be a language weakly recognized by some morphism $h\colon A^+ \to S$ and let $u, v, z \in A^+$ and $x, y \in A^*$ be words such that one of the following two properties holds:*

1. $xuyz^\omega \in L$ *and* $xvyz^\omega \notin L$
2. $x(uy)^\omega \in L$ *and* $x(uyvy)^\omega \notin L$ *and* $x(vyuy)^\omega \notin L$.

Then $h(u) \neq h(v)$.

Proof. We consider finite words $u, v \in A^+$ such that $h(u) = h(v)$ and show that in this case, neither of the properties can hold.

If the first property holds, there exists a linked pair (s, e) such that $xuyz^\omega \in [s][e]^\omega \subseteq L$. Thus, by Lemma 3.1, we have $h(xuyz^k z_1) = s$ and $h(z_2 z^\ell z_1) = e$ for some factorization $z = z_1 z_2$ and powers $k, \ell \geqslant 1$. Now, since $h(xvyz^k z_1) = h(xuyz^k z_1) = s$, we obtain $xvyz^\omega \in [s][e]^\omega \subseteq L$, a contradiction.

If the second property holds, there exists a linked pair (s, e) of S such that $xw^\omega \in [s][e]^\omega \subseteq L$ where $w = uy$. Thus, by Lemma 3.1, we have $h(xw^k w_1) = s$ and $h(w_2 w^\ell w_1) = e$ for some factorization $w = w_1 w_2$, some power $k \geqslant 1$ and some odd power $\ell \geqslant 1$. Since ℓ is odd $(\ell - 1)/2$ is an integer and we have $h(w_2 (vyuy)^{(\ell-1)/2} vyw_1) = h(w_2 (uy)^\ell w_1) = e$. Now, if k is odd as well, we obtain $h(x(vyuy)^{(k-1)/2} vyw_1) = h(x(uy)^k w_1) = s$ and therefore, $x(vyuy)^\omega \in L$. Equivalently, if k is even, we have $h(x(uyvy)^{k/2} w_1) = h(x(uy)^k w_1) = s$ and hence, $x(uyvy)^\omega \in L$. Both cases contradict Property 2 above. ☐

The next proposition is another simple, yet useful, tool for proving lower bounds. It allows to transfer bounds from the setting of finite words to infinite words.

Proposition 3.3. *Let $\mathcal{A} = (Q, A, \delta, I, F)$ and let $a \in A$ be a letter such that for all $q \in Q$ and $q_f \in F$, we have $(q, a, q_f) \in \delta$ if and only if $q = q_f$. Let $K = L_{BA}(\mathcal{A})$ and let $L = L_{NFA}(\mathcal{A})$. Then each semigroup weakly recognizing K has at least $|A^+/{\equiv_L}|$ elements.*

Proof. Let $h\colon A^+ \to S$ be a morphism weakly recognizing K and consider two words $u, v \in A^+$ such that $u \not\equiv_L v$. Then, without loss of generality, there exist $x, y \in A^*$ such that $xuy \in L$ and $xvy \notin L$. This implies $xuya^\omega \in K$ since $(q_f, a, q_f) \in \delta$ for all $q_f \in F$. Equivalently, because of $(q, a, q_f) \notin \delta$ for all $q \in Q \setminus F$ and $q_f \in F$, we have $xvya^\omega \notin K$. By Theorem 3.2, this yields $h(u) \neq h(v)$. ☐

3.2. The full automata technique

The *full automata technique* is a useful tool for proving lower bounds for the conversion of automata to other objects. It was introduced by Yan [25] who attributes it to Sakoda and Sipser [20]. The technique works for both accepted and Büchi-accepted languages. However, we will prove the main result of this section only for the setting of finite words and use Proposition 3.3 to obtain analogous results for infinite words.

Let Q be a finite set and let I, F be subsets of Q. The *full automaton* $\mathcal{F}(Q, I, F)$ is the finite automaton (Q, B, Δ, I, F) defined by $B = 2^{Q^2}$ and by the transition relation $\Delta = \{(p, T, q) \in Q \times B \times Q \mid (p, q) \in T\}$.

Theorem 3.4. *Let $\mathcal{A} = (Q, A, \delta, I, F)$ be a finite automaton and let $\mathcal{F}(Q, I, F) = (Q, B, \Delta, I, F)$ be the corresponding full automaton. Then the syntactic semigroup of $L_{NFA}(\mathcal{A})$ divides the syntactic semigroup of $L_{NFA}(\mathcal{F}(Q, I, F))$.*

Proof. We first define a morphism $\pi\colon A^+ \to B^+$ by $\pi(a) = \{(p, q) \mid (p, a, q) \in \delta\}$. Let $K = L_{\mathrm{NFA}}(\mathcal{F}(Q, I, F))$ and let $L = L_{\mathrm{NFA}}(\mathcal{A})$. It suffices to show that $\pi(u) \equiv_K \pi(v)$ implies $u \equiv_L v$. Thus, consider $u, v \in A^+$ such that $\pi(u) \equiv_K \pi(v)$. In particular, for all $x, y \in A^*$, we have $\pi(xuy) \in K$ if and only if $\pi(xvy) \in K$. By the definition of π, we have $\pi(w) \in K$ if and only if $w \in L$ for all $w \in A^+$. Using the equivalence from above, this yields $xuy \in L$ if and only if $xvy \in L$ for all $x, y \in A^*$, thereby proving that $u \equiv_L v$. ☐

4. From automata to weakly recognizing morphisms

The standard construction for converting a finite automaton $\mathcal{A}$ to a recognizing morphism is the so-called *transition semigroup* of $\mathcal{A}$. For a given word $u \in A^+$, it encodes for each pair (p, q) of states whether there is a run of u on $\mathcal{A}$ starting in p and ending in q. Thus, for a finite automaton with n states the transition semigroup has 2^{n^2} elements. For details on the construction, we refer to [18, 19]. We show that this construction is optimal.

Theorem 4.1. *Let $\mathcal{A}$ be a finite automaton with n states. Then there exists a semigroup recognizing $L_{NFA}(\mathcal{A})$ (resp. weakly recognizing $L_{BA}(\mathcal{A})$) which has at most 2^{n^2} elements and this bound is tight.*

Proof. Each language that is accepted (resp. Büchi-accepted) by $\mathcal{A}$ is recognized (resp. weakly recognized) by the transition semigroup of $\mathcal{A}$ which has size 2^{n^2}.

To show that this is optimal, we consider the full automaton $\mathcal{F}(N, N, N) = (N, B, \Delta, N, N)$ where $N = \{1, \dots, n\}$ and let $L = L_{\mathrm{NFA}}(\mathcal{F}(N, N, N))$. For two different letters $X, Y \in B$ we may assume, without loss of generality, that there exist $p, q \in N$ such that $(p, q) \in X \setminus Y$. With $P = \{(p, p)\}$ and $Q = \{(q, q)\}$, we then have $PXQ \in L$ and $PYQ \notin L$. Thus, $X \not\equiv_L Y$. This shows that $B^+/{\equiv_L}$ has at least $|B| = 2^{n^2}$ elements.

Noting that the transitions labeled by the letter $\{(q, q) \mid q \in N\}$ form self-loops at each state, the Büchi case immediately follows by Proposition 3.3. ☐

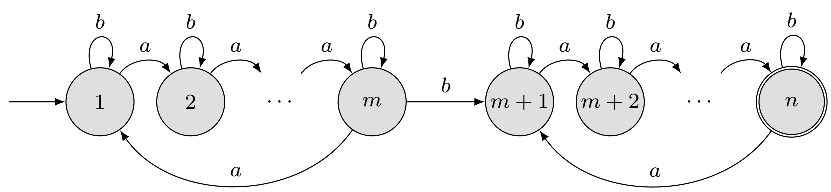
The proof of the optimality result requires a large alphabet that grows super-exponentially in the number of states of the automaton. A natural restriction is considering automata over fixed-size alphabets.

By a result of Chrobak [7], the size of the syntactic semigroup of an unary language accepted by a finite automaton of size n is in $2^{\mathcal{O}(\sqrt{n \log n})}$ (note that since unary languages are commutative, the syntactic monoid is isomorphic to the minimal deterministic automaton). Over infinite words, the unary case is uninteresting since the only language over the alphabet $A = \{a\}$ is $\{a^\omega\}$.

For binary alphabets, a lower bound can be obtained by combining the full automata technique with a result from the study of semigroups of binary relations [13, Proposition 6]. In order to keep the paper self-contained, we present a proof that is adapted to finite automata and does not require any knowledge of binary relations.

Theorem 4.2. *Let $A = \{a, b\}$ and let n be an odd natural number. There exists a language $L \subseteq A^+$ (resp. $L \subseteq A^\omega$) and a finite automaton with n states accepting (resp. Büchi-accepting) L, such that each semigroup recognizing (resp. weakly recognizing) L has at least $2^{(n-1)^2/4}$ elements.*

Proof. We first analyze the case of finite words. Let $m = (n-1)/2$ and let $M = \{1, \dots, m\}$. We consider the automaton $\mathcal{A}$ depicted below and let $L = L_{\mathrm{NFA}}(\mathcal{A})$.



For $1 \leqslant i, j \leqslant m$ we first define $p_{i,j} = (m + j - i)m - i$ and $q_{i,j} = (m + i - j + 2)m + i$. Furthermore, we set $u_{i,j} = a^{p_{i,j}} b a^{q_{i,j}}$. We claim that for each i, j there exists a path from state k to ℓ labeled by $u_{i,j}$ if and only if $(k, \ell) = (i, j + m)$ or $k = \ell$.

The two a-cycles have length m and $m + 1$, respectively. Since for each pair (i, j) we have $p_{i,j} + q_{i,j} = 2m(m + 1)$ and since one can always stay in the same state when reading the letter b, there clearly exists a path from each state to itself labeled by $u_{i,j}$. Now, fix some (i, j) and let $(k, \ell) = (i, j + m)$. We have $i + p_{i,j} = (m + j - i)m$ which means that, when starting in state i, one can reach state m by reading $a^{p_{i,j}}$. Being in state m, one of the b-transitions leads to state $m + 1$. From there on, we make a single step backwards whenever reading the factor a^m. Thus, by reading the word $a^{q_{i,j}}$, we perform $(m+i-j+2)-i = m-j+2$ backward steps in total, finally reaching state $n+1-(m-j+2) = 2m+2-(m-j+2) = m+j = \ell$. The converse direction of our claim follows immediately since the automaton is deterministic when restricted to a-transitions and since one can only reach states $\ell > m$ by using the transition $(m, b, m+1)$.

For $X \subseteq M \times M$, we now define u_X as the concatenation of all $u_{i,j}$ with $(i,j) \in X$, where the factors are ordered according to their indices (i,j). By the above argument, it is easy to see that there is a path from state i to $j+m$ labeled by u_X if and only if $(i,j) \in X$. Since there are $2^{m^2} = 2^{(n-1)^2/4}$ subsets of the Cartesian product $M \times M$, it remains to show that for different subsets $X, Y \subseteq M \times M$, we have $u_X \not\equiv_L v_Y$. To this end, assume without loss of generality that $(i,j) \in X \setminus Y$. Then $a^{i-1} u_X a^{n-j} \in L$ but $a^{i-1} u_Y a^{n-j} \notin L$, as desired.

For the Büchi case note that for all $i \in Q$, we have $(i, b, n) \in \delta$ if and only if $i = n$. Therefore, by Proposition 3.3 and the arguments above, the smallest semigroup weakly recognizing $L_{\mathrm{BA}}(\mathcal{A})$ has at least $2^{(n-1)^2/4}$ elements. ☐

The construction above does not reach the 2^{n^2} bound obtained when using a larger alphabet. However, this is not surprising, given the following result.

Proposition 4.3. *Let $m \in \mathbb{N}$ be a fixed integer and let A be an alphabet of size m. Then there exists an integer $n_m \geqslant 1$ such that for each finite automaton $\mathcal{A}$ over A with $n \geqslant n_m$ states, the language $L_{NFA}(\mathcal{A}) \subseteq A^*$ (resp. $L_{BA}(\mathcal{A}) \subseteq A^\omega$) is recognized (resp. weakly recognized) by a morphism onto a semigroup with less than 2^{n^2} elements.*

We do not give a full proof of the proposition here, but the claim essentially follows from a careful analysis of the subsemigroup of the transition semigroup generated by the transitions corresponding to the letters in A. Applying Devadze's Theorem [8, 14] to the matrix representation of this subsemigroup shows that it is proper, *i.e.*, smaller than the full transition semigroup itself.

5. From weakly recognizing morphisms to automata

The well-known construction to convert weakly recognizing morphisms to finite automata with a Büchi-acceptance condition has quadratic blow-up [18]. We show that this is optimal up to a constant factor.

Theorem 5.1. *Let $A = \{a, b\}$, let $n \geqslant 3$, and let $L = \bigcup_{i=1}^{n} (ba^i bA^*)^\omega$. Then there exists a semigroup with $4n+3$ elements that weakly recognizes L and every finite automaton Büchi-accepting L has at least $n(n+1)/2$ states.*

Proof. We first define a semigroup $S = \left\{a^i, a^i b, ba^i, ba^i b \mid 1 \leqslant i \leqslant n\right\} \cup \{b, bb, 0\}$ by the multiplication $0 \cdot s = s \cdot 0 = 0$ for all $s \in S$ and

$$b^\ell a^i b^r \cdot b^m a^j b^s = \begin{cases} bb & \text{if } i = j = 0 \\ b^\ell a^{i+j} b^s & \text{if } r = m = 0 \text{ and } 1 \leqslant i+j \leqslant n \\ 0 & \text{if } r = m = 0 \text{ and } i+j > n \\ b^\ell a^i b & \text{otherwise} \end{cases}$$

where $\ell, m, r, s \in \{0, 1\}$ and $i, j \in \{0, \ldots, n\}$. The morphism $h \colon A^+ \to S$ defined by $h(a) = a$ and $h(b) = b$ now weakly recognizes L since L is the union of all sets $[ba^i b][ba^i b]^\omega$ with $1 \leqslant i \leqslant n$.

Now assume that we are given a finite automaton $\mathcal{A} = (Q, A, \delta, I, F)$ such that $L_{\mathrm{BA}}(\mathcal{A}) = L$. For each $i \in \{1, \ldots, n\}$, we consider the word $\alpha_i = (ba^i b)^\omega$ and let r_i be an accepting run of α_i. We first show that for $i \neq j$, we have $\inf(r_i) \cap Q \cap \inf(r_j) = \emptyset$, and then prove that $|\inf(r_i) \cap Q| \geqslant i$ for $1 \leqslant i \leqslant n$. Together, this yields

$$|Q| \geqslant \sum_{i=1}^{n} |\inf(r_i) \cap Q| \geqslant \sum_{i=1}^{n} i = n(n+1)/2.$$

Let $i, j \in \{1, \ldots, n\}$ such that $i \neq j$. We assume for the sake of contradiction that there exists a state $q \in Q$ with $q \in \inf(r_i)$ and $q \in \inf(r_j)$. Let $u \in ba^i bA^*$ be a prefix of α_i such that r_i visits q after reading u. Let $v \in A^*$ be a factor of α_j such that there exists a finite run labeled by v, which starts and ends in q, visits at least one final state and such that $v^\omega = (ba^j b)^\omega$ or $v^\omega = a^k b(ba^j b)^\omega$ for some $k \in \{0, \ldots, j\}$. Obviously, we then have $uv^\omega \in L_{\mathrm{BA}}(\mathcal{A})$ but $uv^\omega \notin L$, a contradiction.

For the second part of the proof, assume again for the sake of contradiction that $|\inf(r_i) \cap Q| < i$ for some accepting run r_i of α_i. Then inside each $ba^i b$-factor, a state is visited twice and we can apply the standard pumping argument to show that a word in $A^\omega \setminus L_{\mathrm{BA}}(\mathcal{A})$ has an accepting run as well. □

6. Complementation

To date, the best construction for complementing weakly recognizing morphisms is the so-called *strong expansion* [18]. Given a morphism $h\colon A^+ \to S$, the strong expansion of h is a morphism $g\colon A^+ \to T$ which strongly recognizes all languages weakly recognized by h. If S has n elements, the size of T is 2^{n^2}. The purpose of this section is to give a lower bound for complementation. At the same time, the established bound also serves as a lower bound for the conversion of weak recognition to strong recognition since each morphism strongly recognizing a language also strongly recognizes its complement.

Complementing weakly recognizing morphisms is easy in the case of $\mathcal{J}$-trivial semigroups since each language weakly recognized by a $\mathcal{J}$-trivial semigroup S is already strongly recognized by S, *i.e.*, there is no need to compute the strong expansion if the $\mathcal{J}$-classes of the input are trivial already. In order to establish a lower bound, we thus consider the class of simple semigroups, which is dual to $\mathcal{J}$-trivial semigroups in the sense that simple semigroups consist of a single $\mathcal{J}$-class only.

Proposition 6.1. *Let $n \geqslant 1$ be an arbitrary integer and let $A = \{a_1, a_2, \ldots, a_n\}$. The language $L = \bigcup_{i=1}^{n} (a_i A^*)^\omega$ is weakly recognized by a simple semigroup with n elements and every semigroup weakly recognizing $A^\omega \setminus L$ has at least $n 2^{n-1}$ elements.*

Proof. The alphabet A can be extended to a semigroup by defining an associative operation $a \circ b = a$ for all $a, b \in A$. Now, the morphism $h\colon A^+ \to (A, \circ)$ given by $h(a) = a$ for all $a \in A$ weakly recognizes L. The semigroup $(A, \circ)$ contains $|A| = n$ elements and it is simple because we have $a \mathrel{\mathcal{L}} b$ for all $a, b \in A$.

Now, let $h\colon A^+ \to S$ be a morphism weakly recognizing $A^\omega \setminus L$. For a letter $b \in A$ and a subset $B \subseteq A \setminus \{b\}$, let $u_{b,B}$ be the uniquely defined word $ba_{i_1} a_{i_2} \cdots a_{i_\ell}$ such that $i_1 < i_2 < \cdots < i_\ell$ and $\{a_{i_1}, a_{i_2}, \ldots, a_{i_\ell}\} = B$. Consider two letters $b, c \in A$ and subsets $B \subseteq A \setminus \{b\}$, $C \subseteq A \setminus \{c\}$. If $b \neq c$, we have $u_{b,B} c^\omega \notin L$ and $u_{c,C} c^\omega \in L$. If $B \neq C$ we may assume, without loss of generality, that there exists a letter $a \in B \setminus C$. In this case, we have $a u_{c,C} c^\omega \notin L$ but $a(u_{b,B} u_{c,C})^\omega \in L$ and $a(u_{c,C} u_{b,B})^\omega \in L$. By Theorem 3.2, this suffices to conclude that $h(u_{b,B}) \neq h(u_{c,C})$ whenever $b \neq c$ or $B \neq C$ and therefore, S contains at least $|A|\, 2^{|A|-1} = n 2^{n-1}$ elements. □

Rather surprisingly, the established lower bound turns out to be asymptotically tight in the case of simple semigroups. More generally, for simple semigroups, the construction of the strong expansion can be improved such that only $n 2^n$ elements are needed. This will be proved in the remainder of this section.

We start with a morphism $h\colon A^+ \to S$ onto a simple semigroup with $n = |S|$ elements. Since S is simple, there exists a surjective mapping $\gamma\colon S \to G$ onto a finite group G that becomes a bijection when restricted to a single $\mathcal{H}$-class. Therefore, the mapping $\pi\colon (S/\mathcal{R}) \times G \times (S/\mathcal{L}) \to S$ with $\pi^{-1}(s) = (R_s, \gamma(s), L_s)$ for all $s \in S$ is well-defined and bijective. Moreover, for $s, t \in S$, we write $R_t \cdot s$ to denote the element $\pi(R_t, \gamma(s), L_s)$.

Let $T = \{(s, X) \mid s \in S, X \subseteq S\}$ and let $g\colon A^+ \to T$ be defined by

$$g(u) = (h(u), \{R_{h(q)} \cdot h(p) \mid p, q \in A^+, pq = u\})$$

for all $u \in A^+$. The set T can be extended to a semigroup by defining an associative multiplication

$$(s, X) \cdot (t, Y) = (st, X \cup \{R_t \cdot s\} \cup \hat{Y})$$

where $\hat{Y}$ denotes the set $\{\pi(R_y, \gamma(s(R_t \cdot y)), L_y) \mid y \in Y\}$. Under this extension, the mapping g becomes a morphism.

The following three technical lemmas capture important properties of the construction and are needed for the main proof.

Lemma 6.2. *Let $s, t \in S$. Then $R_t \cdot s$ is the unique element x such that $x \mathcal{R} t$, $x \mathcal{L} s$ and $\gamma(x) = \gamma(s)$ or, equivalently, the unique element x such that $x \mathcal{H} ts$ and $\gamma(x) = \gamma(s)$.*

Proof. Let $x = R_t \cdot s$. We have $(R_x, \gamma(x), L_x) = \pi^{-1}(x) = \pi^{-1}(R_t \cdot s) = (R_t, \gamma(s), L_s)$. Together with the fact that π is bijective, this establishes the first claim. For the second claim, note that since S is simple, $x \mathcal{R} t$ is equivalent to $x \mathcal{R} ts$ and $x \mathcal{L} s$ is equivalent to $x \mathcal{L} ts$. ☐

Lemma 6.3. *Let $u \in A^+$ with $g(u) = (s, X)$ and let $x \in S$. Then $x \in X \cup \{s\}$ if and only if there exists a factorization $u = pq$ with $p \in A^+$ and $q \in A^*$ such that $x \mathcal{H} h(qp)$ and $\gamma(x) = \gamma(h(p))$.*

Proof. Obviously, we have $x = s$ if and only if there exists a factorization $u = pq$ with $p = u$ and $q = \varepsilon$ satisfying the properties described above. Thus, it suffices to consider factorizations where $p, q \in A^+$. By Lemma 6.2, such a factorization exists if and only if $x = R_{h(q)} \cdot h(p)$ which is, in turn, equivalent to $x \in X$ by the definition of g. ☐

Lemma 6.4. *Let (t, f) be a linked pair of S, let $\big((s, X), (e, Y)\big)$ be a linked pair of T and let $\alpha \in [(s, X)]_g [(e, Y)]_g^\omega$. Then $\alpha \in [t]_h [f]_h^\omega$ if and only if $tq = s$, $pq = e$, $qp = f$, $R_q \cdot t \in X$ and $R_q \cdot p \in Y$ for some $p, q \in S$.*

Proof. For the direction from left to right, let $\alpha = uv_1v'_1v_2v'_2 \cdots$ such that $g(u) = (s, X)$, $g(v_iv'_i) = (e, Y)$, $h(uv_1) = t$ and $h(v'_iv_{i+1}) = f$ for all $i \geqslant 1$. Furthermore, we assume without loss of generality that $v_i, v'_i \neq \varepsilon$ for all $i \geqslant 1$ and that $h(v_1) = h(v_2)$. We set $p = h(v_1) = h(v_2)$ and $q = h(v'_1)$. Now, $tq = h(uv_1v'_1) = se = s$, $pq = h(v_1v'_1) = e$ and $qp = h(v'_1v_2) = f$. Moreover, by the definition of g, we have $R_q \cdot t = R_{h(v'_1)} \cdot h(uv_1) \in X$ and $R_q \cdot p = R_{h(v'_1)} \cdot h(v_1) \in Y$.

For the converse implication, note that by Lemma 6.3, there exists a factorization $\alpha = uv_1v'_1v_2v'_2 \cdots$ such that $h(u) = s$, $h(v_iv'_i) = e$, $R_{h(v'_1)} \cdot h(uv_1) = R_q \cdot t$ and $R_{h(v'_i)} \cdot h(v_i) = R_q \cdot p$ for all $i \geqslant 1$. Since S is simple, $h(v_i) \mathcal{R} h(v_iv'_i) = e \mathcal{R} p$ and $h(v_i) \mathcal{L} (R_{h(v'_i)} \cdot h(v_i)) = (R_q \cdot p) \mathcal{L} p$ for all $i \geqslant 1$. Furthermore, $\gamma(h(v_i)) = \gamma(R_{h(v'_i)} \cdot h(v_i)) = \gamma(R_q \cdot p) = \gamma(p)$. Together, this yields $h(v_i) = p$ by Lemma 6.2. Similarly, we have $h(v'_i) \mathcal{R} (R_{h(v'_i)} \cdot h(v_i)) = (R_q \cdot p) \mathcal{R} q$ and thus, $ph(v'_i) = h(v_iv'_i) = pq$ implies $h(v'_i) = q$ for all $i \geqslant 1$ by Lemma 2.1. This shows that $h(uv_1) = sp = tqp = tf = t$ and $h(v'_iv_{i+1}) = qp = f$. We conclude that $\alpha \in [t][f]^\omega$. ☐

Theorem 6.5. *Let $h\colon A^+ \to S$ be a morphism onto a simple semigroup of size $n = |S|$ that weakly recognizes a language $L \subseteq A^\omega$. Then there exists a morphism $g\colon A^+ \to T$ to a semigroup of size $|T| = n2^n$ that strongly recognizes L.*

Proof. The construction we use is the one described following the proof of Proposition 6.1. Consider a linked pair $((s, X), (e, Y))$ of T as well as two infinite words $\alpha, \beta \in [(s, X)][(e, Y)]^\omega$. If $\alpha \in L$, there exists a linked pair (t, f) of S such that $\alpha \in [t][f]^\omega \subseteq L$. Lemma 6.4 immediately yields $\beta \in [t][f]^\omega \subseteq L$, thereby showing that g strongly recognizes L. ☐

7. Decision problems

The most common decision problems in formal language theory are *membership*, *inclusion*, *equivalence*, *emptiness* and *universality*. It is well-known that for every fixed regular language over finite words, the membership problem is contained in NC^1. Moreover, if the syntactic semigroup of the language contains a non-solvable group, the problem is NC^1-complete [2]. If the regular language is not fixed but given as a deterministic finite automaton as part of the input, the problem is L-complete, while for nondeterministic finite automata, the problem is NL-complete [11]. The emptiness problem is NL-complete for both deterministic and nondeterministic finite automata [11]. The inclusion, equivalence and universality problems are known to be NL-complete for deterministic finite automata and PSPACE-complete for nondeterministic finite automata [6, 15].

When the regular languages in the input are represented using recognizing morphisms instead of automata, the membership problem is decidable in deterministic log-space but the best lower bound is NC^1-hardness from the fixed language case. The inclusion, equivalence, emptiness and universality problems are NL-complete. With the exception of emptiness, the problems remain NL-complete even if the morphism in the input is required to be surjective. Most of these completeness results follow from NL-completeness of the *associative generation problem* (sometimes also referred to as *Cayley semigroup membership problem*) studied in [12] already. The situation over finite words closely resembles the situation for recognizing morphisms over infinite words, which is studied in this section.

Throughout this section, we assume that morphisms are encoded as a set of letters A, a set of elements S, a multiplication table and a set of tuples $(a, s) \in A \times S$, representing the images of the letters. The multiplication table is assumed to be given in some aligned form, such that a random access Turing machine can perform a table lookup to compute the product of two elements in $\mathcal{O}(\log |S|)$ time.

We define a family of semigroups which will be used to prove some hardness results later in this section. For a set V, the so-called *aperiodic Brandt semigroup* B_V is the set $V \times V \cup \{0\}$ together with the multiplication $(v, w) \cdot 0 = 0 \cdot (v, w) = 0 \cdot 0 = 0$ and

$$(v, w) \cdot (v', w') = \begin{cases} (v, w') & \text{if } w = v' \\ 0 & \text{otherwise} \end{cases}$$

for all $v, w, v', w' \in V$.

7.1. Fixed and general membership

Since arbitrary infinite words cannot be encoded using finitely many bits, we only consider ultimately periodic words. The *general membership problem* for weakly recognizing morphisms is defined as follows:

Input: A morphism $h\colon A^+ \to S$, a set $P \subseteq F(S)$ and words $u, v \in A^+$
Question: Is $uv^\omega \in [P]$?

In the *fixed membership problem*, both the morphism $h\colon A^+ \to S$ and the set $P \subseteq F(S)$ are fixed:

Input: Finite words $u, v \in A^+$
Question: Is $uv^\omega \in [P]$?

We first show that both problems are NC^1-hard. For fixed membership, NC^1-hardness in this context means that there exists some fixed semigroup for which the problem is NC^1-hard. More specifically, it can be shown that the hardness result holds whenever the fixed semigroup contains non-solvable groups.

Proposition 7.1. *The fixed membership problem for weakly recognizing morphisms is NC^1-hard if the fixed semigroup contains a non-solvable group. The general membership problem for weakly recognizing morphisms is NC^1-hard.*

Proof. The fixed membership problem for recognizing morphisms over finite words is NC^1-complete if the fixed semigroup contains a non-solvable group, such as the symmetric group of degree five, see [2]. The problem in the setting of finite words is easily reducible to the membership problem for weakly recognizing morphisms: suppose we are given a morphism $h\colon A^+ \to S$ and a set $P \subseteq S$. Without loss of generality, we may assume that S is a monoid; otherwise, we adjoin a neutral element 1. We extend the alphabet by adding a new letter c and we let $h(c) = 1$. Then, the input u is extended by adding $v = c$ as second word and the accepting set P is replaced by $\{(s, 1) \mid s \in P\}$. ☐

We now show that (under the hardness assumptions mentioned above) fixed membership is, in fact, NC^1-complete.

Theorem 7.2. *The fixed membership problem for weakly recognizing morphisms is contained in* ALOGTIME.

Proof. Let $h\colon A^+ \to S$ be a morphism, let $P \subseteq F(S)$ and let $u, v \in A^+$ be two words. We exploit the combinatorial properties shown in Lemma 3.1, allowing us to significantly reduce the number of factorizations that need to be tested.

To this end, we first guess a factorization $v = v_1 v_2$ (*i.e.*, a position $i \in \{1, \ldots, |v|\}$ representing this factorization), a linked pair $(s, e) \in P$ and integers $k \in \{1, \ldots, |S|\}$ and $\ell \in \{1, \ldots, 2|S| + 1\}$. We then verify that $s = h(uv^k v_1)$ and $e = h(v_2 v^\ell v_1)$. Note that the words $uv^k v_1$ and $v_2 v^\ell v_1$ have lengths linear in the size of the input. It therefore remains to show how to test $x_1 x_2 \cdots x_m = y$ for $x_1, x_2, \ldots, x_m, y \in S$ when m is linearly bounded by the input size.

In the random access model, the product of two elements can be computed in constant time. To check whether $x_1 x_2 \cdots x_m = y$, we split the product into two factors $x_1 x_2 \cdots x_{\lfloor m/2 \rfloor}$ and $x_{\lfloor m/2 \rfloor + 1} \cdots x_{m-1} x_m$ and guess the results y_1 and y_2 of each of the factors. We then branch universally, running three verification branches in parallel. The first branch verifies that $y_1 y_2 = y$, the second and third branch verify that $x_1 x_2 \cdots x_{\lfloor m/2 \rfloor} = y_1$ and $x_{\lfloor m/2+1 \rfloor} \cdots x_m = y_2$ by running the algorithm recursively. Since the length of the products to verify is halved in each step, this procedure clearly terminates after logarithmically many steps. ☐

Note that the algorithm for still works if the morphism and the set P are part of the input, as long as the semigroup S is fixed. For general membership, we are able to give a deterministic algorithm that requires only logarithmic space but the exact complexity remains open.

Theorem 7.3. *The general membership problem for weakly recognizing morphisms is contained in* L.

Proof. Let $h\colon A^+ \to S$ be a morphism, let $P \subseteq F(S)$ and let $u, v \in A^+$ be two words. As in the proof of the previous theorem, we exploit the combinatorial properties shown in Lemma 3.1.

In a loop, we iterate over all possible factorizations $v = v_1 v_2$ (*i.e.*, positions $i \in \{1, \ldots, |v|\}$ representing this factorization), over all linked pairs $(s, e) \in P$ and over all integers $k \in \{1, \ldots, |S|\}$ and $\ell \in \{1, \ldots, 2|S| + 1\}$. All loop variables require only logarithmic space. Within this loop, we check whether $s = h(uv^k v_1)$ and $e = h(v_2 v^\ell v_1)$. To this end, we compute the images $h(uv^k v_1)$ and $h(v_2 v^\ell v_1)$ letter by letter. Logarithmic space suffices to store single elements and to perform single multiplications. ☐

7.2. Emptiness and surjectivity

We now investigate the complexity of deciding whether a specific element (resp. all elements) are in the image of a morphism.

Proposition 7.4. *The following problems are* NL*-complete:*

1. *Given a morphism $h\colon A^+ \to S$ and $s \in S$, decide whether $[s] \neq \emptyset$.*
2. *Given a morphism $h\colon A^+ \to S$ and a linked pair $(s, e) \in F(S)$, decide whether $[s][e]^\omega \neq \emptyset$.*
3. *Checking whether a morphism $h\colon A^+ \to S$ is surjective.*

Proof. To see that the first problem is in NL, consider the following algorithm. We successively guess the letters $a_1 a_2 \cdots a_n$ of some word in $[s]$, starting with the first letter a_1. After guessing the first letter, we compute its image $h(a_1)$ and store the result in some global variable x. After guessing the i-th letter for $i > 1$, we update x by setting $x \leftarrow x \cdot h(a_i)$. Finally, we check whether $x = s$.

This also yields an NL-algorithm for the other two problems. For the second problem, we use the algorithm above to verify that both $[s] \neq \emptyset$ and $[e] \neq \emptyset$. To test for surjectivity, we simply iterate over all elements of S, running the algorithm above for each of them.

For NL-hardness, we describe a log-space reduction of STCon to the first problem. Let $G = (V, E)$ be a directed graph with $|V| \geqslant 2$ and let $s, t \in V$ be two vertices. We want to check whether t is reachable from s. Let S be the aperiodic Brandt semigroup B_V, let $A = E \cup \{(v, v) \mid v \in V\}$ and let $h\colon A^+ \to S$ be the morphism defined by letting $h(a) = a$ for all $a \in A$. Then, by construction, the language $[(s, t)]$ is non-empty if and only if t is reachable from s in G. Note that the same construction works for the second problem: the language $[(s,t)][(t,t)]^\omega$ is also non-empty if and only if t is reachable from s. Since, more generally, $[(v,w)]$ is non-empty if and only if w is reachable from v for any pair of vertices $(v,w) \in V \times V$, this also yields a log-space reduction of Strongly-Connected to the third problem. □

The proposition immediately implies that for weakly recognizing morphisms, the emptiness problem is NL-complete as well, because the language $[P]$ recognized by a morphism $h\colon A^+ \to S$ is non-empty whenever there exists some $(s,e) \in P$ such that $[s][e]^\omega \neq \emptyset$.

Since deciding surjectivity is NL-complete already, we now ask for the complexity of each of the decision problems when the given morphisms are known to be surjective. Clearly, the emptiness problem becomes trivial since the language $[P]$ recognized by a surjective morphism is non-empty whenever P is non-empty.

7.3. Inclusion, equivalence and universality

In the remainder of this section, we show that inclusion, equivalence and universality remain NL-complete, even if only surjective morphisms are allowed in the input.

Theorem 7.5. *The inclusion, equivalence and universality problems for weakly recognizing morphisms are* NL-*complete. This still holds if all morphisms in the input are restricted to be surjective.*

Before presenting the proof, we make some preliminary considerations. Given two weakly recognizing morphisms $h\colon A^+ \to S$ and $g\colon A^+ \to T$, the direct product $S \times T$ weakly recognizes all languages recognized by h or by g. It is therefore non-restrictive to demand that inputs to the inclusion problem and the equivalence problem are given by a single morphism. With this simplification and with the restriction to surjective morphisms in mind, we now formalize the inclusion problem for weakly recognizing morphisms as follows:

Input:	A surjective morphism $h\colon A^+ \to S$ and two sets $P, Q \subseteq F(S)$
Question:	Is $[P] \subseteq [Q]$?

The equivalence problem and the universality problem can be defined analogously.

Testing for universality is equivalent to testing for equivalence while specifying the set of all linked pairs as second accepting set. Testing two morphisms for equivalence boils down to testing for inclusion twice, interchanging the accepting sets. The following proposition is a slightly stronger result.

Proposition 7.6. *The universality problem for weakly recognizing morphisms is log-space reducible to the equivalence problem for weakly recognizing morphisms. The equivalence problem for weakly recognizing morphisms is log-space reducible to the inclusion problem for weakly recognizing morphisms.*

Proof. For the first part of the statement, suppose we are given some morphism $h\colon A^+ \to S$ and a set $P \subseteq F(S)$. To check whether $[P] = A^\omega$, we transform the input to an instance of the equivalence problem by adding a second accepting set $Q = F(S)$.

For the second part of the statement, suppose we are given some morphism $h\colon A^+ \to S$ and two sets $P, Q \subseteq F(S)$ for which we want to check whether $[P] = [Q]$. We describe a log-space reduction to the inclusion problem.

Let $\overline{A}$ be a disjoint copy of A and let $\overline{S}$ be a disjoint copy of S. For $a \in A$, we denote by $\overline{a}$ the corresponding letter in $\overline{A}$ and for $s \in S$, we denote by $\overline{s}$ the corresponding element in $\overline{S}$. For a set $X \subseteq S \times S$, we let $\overline{X} = \{(\overline{s}, \overline{t}) \mid s, t \in S\}$.

We define the semigroup T as $S \cup \overline{S} \cup \{0\}$ where 0 is a new zero element. The binary operation $\circ$ on T is given by $s \circ t = st$, $\overline{s} \circ \overline{t} = \overline{st}$ and $s \circ \overline{t} = \overline{s} \circ t = x \circ 0 = 0 \circ x = 0$ for all $s, t \in S$ and $x \in T$. We also define a new alphabet $B = A \cup \overline{A}$ and a morphism $g\colon B^+ \to T$ by $g(a) = h(a)$ and $g(\overline{a}) = \overline{h(a)}$ for all $a \in A$. By construction, we have $[P] = [Q]$ if and only if $[P \cup \overline{Q}] \subseteq [Q \cup \overline{P}]$. □

This shows that inclusion is harder than equivalence and equivalence is harder than universality. It therefore suffices to show that inclusion is in NL and that universality is NL-hard.

We will present a nondeterministic log-space algorithm to decide the inclusion problem. The following variant of Lemma 3.1 serves as a basis for our construction.

Lemma 7.7. *Let $u, v \in A^+$ such that $(h(u), h(v))$ forms a linked pair and let (s, e) be a linked pair. Then uv^ω is contained in $[s][e]^\omega$ if and only if there exists a factorization $v = v_1 v_2$ such that $h(uv_1) = s$ and $h(v_2 v v_1) = e$.*

It is an immediate consequence of Lemma 3.1 and the definition of linked pairs. We are now able to describe the algorithm.

Theorem 7.8. *The inclusion problem for weakly recognizing morphisms is in* NL.

Proof. Because of $\mathsf{NL} = \mathsf{coNL}$ it suffices to show that the complement of the inclusion problem is in NL.

Let $h \colon A^+ \to S$ be a morphism onto a finite semigroup S and let P and Q be sets of linked pairs of S.

We start by guessing a triple $(s, 1, e) \in S \times S^1 \times S$ such that $(s, e) \in P \setminus Q$. Then, we successively take the current triple (s, x, y), guess some element $q \in S^1$ and some letter $a \in A$ such that $y = h(a)q$, and replace the current triple by $(s, xh(a), q)$. If for any of the intermediate triples (s, x, y), we have $(sx, yxyx) \in Q$, we discard the current computation branch without accepting. If we arrive at some triple of the form $(s, e, 1)$, we halt and accept.

Logarithmic space obviously suffices to perform the computation. It remains to show that there exists an accepting branch if and only if $[P] \not\subseteq [Q]$.

Suppose that the algorithm accepts and let $a_1, \ldots, a_k$ be the letters guessed on an accepting path. Let $u \in A^+$ be an arbitrary word such that $h(u) = s$ and let $\alpha = u(a_1 a_2 \cdots a_k)^\omega$. Then $\alpha \in [P] \setminus [Q]$ by Lemma 7.7.

For the converse direction, assume that there exists a word $\alpha \in [P] \setminus [Q]$. Since every non-empty ω-regular language contains an ultimately periodic word, we may assume without loss of generality that $\alpha = u(a_1 a_2 \cdots a_k)^\omega$ where $h(u) = s$ and $h(a_1 a_2 \cdots a_k) = e$ for a linked pair $(s, e) \in P$. Lemma 7.7 shows that, starting with the initial guess $(s, 1, e)$, the sequence $a_1, \ldots, a_k$ of guessed letters now yields an accepting computation path in the algorithm. ☐

Let us now show that universality is NL-hard, thereby providing a proof for the second part of the Theorem 7.5.

Theorem 7.9. *The universality problem for weakly recognizing morphisms is* NL*-hard.*

Proof. We describe a log-space reduction of CYCLE to the complement of the universality problem for weakly recognizing morphisms. This shows that the universality problem is coNL-hard and thus, by the Theorem of Immerman-Szelepcsényi, it is NL-hard as well.

Let $G = (V, E)$ be a directed graph with $n \geqslant 2$ vertices. Let $V' = V \cup \{\circledast\}$, where $\circledast$ is a new symbol that is not in V, and let $A = \{\circledast\} \times V \cup E \cup V \times \{\circledast\}$. Let S be the aperiodic Brandt semigroup $B_{V'}$, let $h \colon A^+ \to S$ be the morphism defined by $h(a) = a$ for all $a \in A$ and let $P = \{((v, \circledast), (\circledast, \circledast)) \mid v \in V\} \cup \{(0, 0)\}$.

For all $v, w \in V$, we have $h((v, \circledast)(\circledast, w)) = (v, w)$, $h((\circledast, v)(v, \circledast)) = (\circledast, \circledast)$ and $h((v, \circledast)(v, \circledast)) = 0$ so h is surjective. It remains to show that $[P] \neq A^\omega$ if and only if G contains a directed cycle.

Suppose that G contains a directed cycle $v_1 v_2 \cdots v_k v_1$ and consider the infinite word $\alpha = ((v_1, v_2)(v_2, v_3) \cdots (v_{k-1}, v_k)(v_k, v_1))^\omega$. For each linked pair (s, e) of S with $\alpha \in [s][e]^\omega$, there exists some $i \in \{1, \ldots, k\}$ such that $s = (v_1, v_i)$ and $e = (v_i, v_i)$, and thus $(s, e) \notin P$. This shows that $\alpha \in A^\omega \setminus [P]$.

Conversely, suppose that there exists an infinite word $\alpha \in A^\omega \setminus [P]$. Then there exists a factorization $\alpha = uw_1 w_2 \cdots$ with $h(u) = (v, w)$ and $h(w_i) = (w, w)$ for some $v, w \in V$ such that the letter $\circledast$ does not occur in any of the words w_i. The sequence of vertices appearing in the finite word w_1 now induces a directed cycle in G.

The size of the constructed semigroup is $(n + 1)^2 + 1 \in \mathcal{O}(n^2)$ and multiplications in S can be performed in space $2 \log((n + 1)^2 + 1) \in \mathcal{O}(\log n)$. ☐

7.4. Binary alphabets

One may observe that the log-space reduction described in the previous section does not work for a fixed alphabet: the size of the alphabet depends on the input. Indeed, the universality problem is trivial for unary alphabets as $\{a^\omega\}$ is the only non-empty ω-regular language over $A = \{a\}$. We will now show that the three decision problems covered before are NL-complete for binary alphabets already. In the binary case, the input essentially consists of a semigroup, two generators and a set of linked pairs. In view of Theorem 7.8 and the proof of Theorem 7.5, it suffices to show that the universality problem remains NL-hard.

For a surjective morphism $h\colon A^+ \to S$ and a set of linked pairs P, we now aim at constructing a surjective morphism $g\colon \mathbb{B}^+ \to T$ over the binary alphabet $\mathbb{B} = \{a, b\}$ and a set $Q \subseteq F(T)$ such that $[Q]_g = \mathbb{B}^\omega$ if and only if $[P]_h = A^\omega$. Furthermore, we want to be able to perform multiplications in T in logarithmic space. To avoid confusion, we use $\circ$ to denote the operation of S in the remainder of this section, and for $s, t \in S$ we denote by st the two-letter word over the free semigroup S^+.

The elements of A are denoted by $a_1, \ldots, a_m$. We set $\Gamma = \mathbb{B} \cup S \cup \{0\}$ and define a semigroup $T = \Gamma^+/R$ by

$$R = \left\{ba^ib = h(a_i), st = s \circ t \mid 1 \leqslant i \leqslant m \text{ and } s, t \in S\right\} \cup \\ \left\{a^{m+1} = aba = b^3 = 0x = x0 = 0 \mid x \in \Gamma\right\}.$$

The morphism $g\colon \mathbb{B} \to T$ is given by $g(a) = a$ and $g(b) = b$. It is convenient to identify each element of T with a representative of minimal length. The set of minimal representatives for the non-zero classes consists of the non-empty elements a^i and $(a^ib)^\ell s(ba^j)^r$ with $i, j \in \{0, \ldots, m\}$ and $\ell, r \in \{0, 1\}$ and $s \in S \cup \{\varepsilon\}$. Using this notation, we say a linked pair (u, e) of T is *valid* if $u \in S \cup Sba^*$. An infinite word $\alpha \in \mathbb{B}^\omega$ is valid if $\alpha \in [u][e]^\omega$ for some valid linked pair (u, e) of T. Let

$$Q = \{(u, e) \in F(T) \mid (u, e) \in P \text{ or } (u, e) \text{ is not valid}\}.$$

Lemma 7.10. *An infinite word $\beta \in \mathbb{B}^\omega$ is valid if and only if there exists a sequence of natural numbers $(i_1, i_2, i_3, \ldots)$ such that $\beta = ba^{i_1}bba^{i_2}bb\cdots$. Furthermore, if such a sequence exists, then $\beta \in [Q]_g$ if and only if $a_{i_1}a_{i_2}\cdots \in [P]_h$.*

Proof. The first part of the claim follows by a simple case analysis: first note that $(0, 0)$ is the only linked pair with the second component being 0 and that $\beta \in [0][0]^\omega$ implies $\beta \notin [u][e]^\omega$ for all linked pairs $(u, e) \neq (0, 0)$. By the definition of T, we have $\beta \in [0][0]^\omega$ if and only if β contains any of the factors a^{m+1}, aba or b^3. The definition of valid linked pairs also ensures that each word β in the preimage of such a linked pair is contained in $ba\mathbb{B}^\omega$.

For the second part of the claim, since we already know that β is valid, it suffices to show that $\beta \in [Q]_g$ if and only if $a_{i_1}a_{i_2}\cdots \in [P]_h$. This is immediate after checking that factorizations of $a_{i_1}a_{i_1}\cdots$ correspond to factorizations of β where each factor starts and ends with the letter b. ☐

We can now prove the correctness of the construction.

Theorem 7.11. *Let $g\colon \mathbb{B}^+ \to T$ be a morphism onto a finite semigroup T and let $Q \subseteq F(T)$. The problem of deciding whether $[Q]_g = \mathbb{B}^\omega$ is NL-hard.*

Proof. We give a log-space reduction of the general universality problem for weakly recognizing morphisms, with the alphabet given as part of the input, to the binary case. By Theorem 7.9, NL-hardness follows.

To this end, let $h\colon A^+ \to S$ be an arbitrary morphism, let $P \subseteq F(S)$ and let g, Q, T be the structures obtained from the construction described above. If $a_{i_1}a_{i_2}\cdots \notin [P]_h$, then $ba^{i_1}bba^{i_2}bb\cdots \notin [Q]_g$ by Lemma 7.10. Conversely, if $\beta \notin [Q]_g$, then β must be valid and therefore $\beta = ba_{i_1}bba_{i_2}bb\cdots$ for some $i_j \geqslant 1$. This implies $a_{i_1}a_{i_2}\cdots \notin [P]_h$. Thus, $[P]_h = A^\omega$ if and only if $[Q]_g = \mathbb{B}^\omega$.

Note that $|T| = (m^2 + 4m + 4)\,|S| + m^2 + 5m + 3$ which is in $\mathcal{O}(m^2\,|S|)$. Therefore, multiplications in T can be performed in logarithmic space. ☐

8. Discussion and open problems

We presented lower bound techniques and gave tight bounds for the conversion between finite automata and weakly recognizing morphisms. We also investigated the computation complexity of language-theoretic decision problems for weakly recognizing morphisms.

One can use techniques similar to those described in Section 4 to obtain a 3^{n^2} lower bound for the conversion of finite automata with transition-based Büchi acceptance to strongly recognizing morphisms. However, with the usual state-based Büchi acceptance criterion, the analysis becomes much more involved and it is not clear whether the 3^{n^2} upper bound can be reached. Analogously, there is no straightforward adaptation of the conversion of weakly recognizing morphisms into BA in Section 5 to strongly recognizing morphisms. It would be interesting to see whether the quadratic lower bound also holds in this setting.

It would also be nice to close the remaining gaps between the upper and the lower bounds, in particular for complementation and the conversion of weakly recognizing morphisms to strong recognition. We showed that there is an exponential lower bound and gave an asymptotically optimal construction for simple semigroups which was a first candidate for semigroups that are hard to complement. It is easy to adapt this construction to families of semigroups where the size of each $\mathcal{J}$-class is bounded by a constant. However, for the general case, the gap between $n2^{n-1}$ and 2^{n^2} remains.

In our analysis of lower and upper bounds, we always only considered the size of automata and the size of semigroups as parameters. Can the bounds be improved by considering the size of the accepting set, *i.e.*, the number of linked pairs used to describe a language?

When it comes to the computational complexity of weakly recognizing morphisms, the general membership problem seems to be an interesting candidate for further research. Even in the case of finite words, it is not clear whether there exists an algorithm for general membership better than the naïve log-space algorithm or whether the problem is complete for deterministic logarithmic space.

References

[1] A. Arnold, A syntactic congruence for rational ω-languages. *Theor. Comput. Sci.* **39** (1985) 333–335.

[2] D.A.M. Barrington, Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1, in *Proc. of the 18th Annual ACM Symposium on Theory of Computing, May 28–30, 1986, Berkeley, California, USA*, edited by J. Hartmanis. ACM, NY (1986) 1–5.

[3] D.A.M. Barrington, N. Immerman and H. Straubing, On uniformity within NC^1. *J. Comput. Syst. Sci.* **41** (1990) 274–306.

[4] J.R. Büchi, Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* **6** (1960) 66–92.

[5] S.R. Buss, The boolean formula value problem is in ALOGTIME, in *Proc. of the 19th Annual ACM Symposium on Theory of Computing*, New York, USA (1987) 123–131.

[6] S. Cho and D.T. Huynh, The parallel complexity of finite-state automata problems. *Inform. Comput.* **97** (1992) 1–22.

[7] M. Chrobak, Finite automata and unary languages. *Theor. Comput. Sci.* **47** (1986) 149–158.

[8] H.M. Devadze, Generating sets of the semigroup of all binary relations in a finite set. *Doklady Akademii Nauk BSSR* **12** (1968) 765–768.

[9] M. Holzer and B. König, On deterministic finite automata and syntactic monoid size. *Theor. Comput. Sci.* **327** (2004) 319–347.

[10] N. Immerman, Nondeterministic space is closed under complementation. *SIAM J. Comput.* **17** (1988) 935–938.

[11] N.D. Jones, Space-bounded reducibility among combinatorial problems. *J. Comput. Syst. Sci.* **11** (1975) 68–85.

[12] N.D. Jones, Y.E. Lien and W.T. Laaser, New problems complete for nondeterministic log space. *Math. Syst. Theory* **10** (1976) 1–17.

[13] K.H. Kim and F.W. Roush, Two-generator semigroups of binary relations. *J. Math. Psychol.* **17** (1978) 236–246.

[14] J. Konieczny, A proof of Devadze's theorem on generators of the semigroup of boolean matrices. *Semigroup Forum* **83** (2011) 281–288.

[15] A.R. Meyer and L.J. Stockmeyer, The equivalence problem for regular expressions with squaring requires exponential space, in *13th Annual Symposium on Switching and Automata Theory*, IEEE Computer Society, Washington, DC, USA (1972) 125–129.

[16] Ch.H. Papadimitriou, Computational Complexity. Addison Wesley, MA, USA (1994).

[17] J. Pécuchet, Variétés de semis groupes et mots infinis, in *STACS 1986, Proceedings* (1986) 180–191.

[18] D. Perrin and J.-É. Pin, Infinite words. Vol. 141 of *Pure and Applied Mathematics*. Elsevier, Berlin, Heidelberg, Germany, (2004).

[19] J.-É. Pin, Varieties of Formal Languages. North Oxford Academic, London and Plenum, New-York (1986).

[20] W.J. Sakoda and M. Sipser, Nondeterminism and the size of two way finite automata, in *STOC 1978, Proceedings*. ACM Press, San Diego, California, USA (1978) 275–286.
[21] M. Sipser, Introduction to the Theory of Computation. 1st edition, International Thomson Publishing, Boston, MA, USA (1996).
[22] R. Szelepcsényi, The method of forced enumeration for nondeterministic automata. *Acta Inform.* **26** (1988) 279–284.
[23] W. Thomas, Automata on infinite objects, in Handbook of Theoretical Computer Science, chapter 4. Elsevier (1990) 133–191.
[24] H. Vollmer, Introduction to Circuit Complexity. Springer, Berlin, Germany (1999).
[25] Q. Yan, Lower bounds for complementation of omega-automata via the full automata technique. *Log. Meth. Comput. Sci.* **4** (2008) lmcs:992.