Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 330 (2005) 349–360
Theoretical Computer Science
www.elsevier.com/locate/tcs

Complementing unary nondeterministic automata☆

Filippo Mera, Giovanni Pighizzini*

Dipartimento di Informatica e Comunicazione, Università degli Studi di Milano, via Comelico 39, 20135 Milano, Italy

Received 14 November 2003; received in revised form 7 April 2004; accepted 23 April 2004

Abstract

We compare the nondeterministic state complexity of unary regular languages and that of their complements: if a unary language $\mathscr{L}$ has a succinct nondeterministic finite automaton, then nondeterminism is useless in order to recognize its complement, namely, the smallest nondeterministic automaton accepting the complement of $\mathscr{L}$ has as many states as the minimum deterministic automaton accepting it. The same property does not hold in the case of automata and languages defined over larger alphabets. We also show the existence of infinitely many unary regular languages for which nondeterminism is useless in their recognition and in the recognition of their complements.

Keywords: Formal languages; Finite state automata; Unary languages; Descriptional complexity; Complementation

1. Introduction

In the last few years, we observed a renewed interest for research in automata theory (for a discussion, we address the reader to [9,23]).

Some aspects of this field that were only partially considered in the early developments of the theory are now extensively and deeply investigated. Two such aspects are the descriptional complexity of automata and the analysis of unary languages.

☆ Partially supported by MIUR, under the projects "Complessità descrizionale di automi e strutture correlate" and "Linguaggi formali: metodi, modelli e applicazioni".

* Corresponding author.
E-mail addresses: mera@dico.unimi.it (F. Mera), pighizzi@dico.unimi.it (G. Pighizzini).

0304-3975/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2004.04.015

Descriptional complexity studies the costs of the description of objects (e.g., languages) by different formal systems (e.g., deterministic or nondeterministic automata, grammars, etc.). Furthermore, in this area formal systems with the same expressive power are compared with respect to their conciseness. Probably the first and most widely known result of this kind is the simulation of nondeterministic finite automata (nfa) by deterministic ones (dfa). While these two models share the same computational power, i.e., they characterize the class of regular languages [21], from the point of view of descriptional complexity they are very different. In fact, nfa's can be exponentially more succinct than dfa's, i.e., for any nonnegative integer n there is a language accepted by an n-state nfa that *requires* 2^n states to be accepted by a dfa [17,18]. For a recent survey on descriptional complexity, we address the reader to [4], while for descriptional complexity of regular languages some recent works are [24,3,7].

Languages and automata are called *unary* when they are defined over a one-letter alphabet. Some interesting properties of unary automata were studied by Chrobak [2], showing many important differences with respect to the general, i.e., "nonunary" case. These studies have been recently deepened, mainly in connection with descriptional complexity issues (see, e.g., [3,6,16,19,20]).

This paper continues this stream of research. In particular, we study some questions related to the complement operation. While it is trivial to show that any regular language and its complement can be recognized by dfa's with the same number of states, in the case of nfa's the situation can be totally different: the only known standard way to achieve complementation is to convert an nfa accepting a given language to a dfa, and then complement the set of final states. This may lead to an increment from n to 2^n in the number of states. This gap was originally proved to be optimal for a four-letter alphabet by Birget [1] and, recently, for a two-letter alphabet by Jirásková [11], who showed that for each integer n there exists a language $\mathcal{L}$ accepted by an nfa with n states such that *each nfa* accepting the complement of $\mathcal{L}$ has at least 2^n states. In this paper, we consider the same problem for unary languages. Our main result states that if a unary regular language $\mathcal{L}$ has a succinct nfa (i.e., an nfa with the minimum possible number of states with respect to the periodicity of the accepted language [10]), then nondeterminism is useless in order to recognize its complement, i.e., each nfa accepting $\mathcal{L}^c$ has at least the same number of states as the minimum dfa accepting it. Roughly speaking, succinct nfa's witness the optimality of the simulation of n-state unary nfa's by $e^{\Theta(\sqrt{n \ln n})}$-state dfa's proved in [2]. A similar result does not hold in the general case. In fact, we prove that, without the unary restriction, for each integer $n \geqslant 2$, there is a language $\mathcal{L}$ accepted by an nfa with n states such that the minimum dfa accepting it must have 2^n states (i.e., $\mathcal{L}$ is a witness of the state gap between nfa's and dfa's in the general case) and the complement of $\mathcal{L}$ is accepted by an nfa with $n+1$ states. In other words, in the general case there are languages with small nfa's accepting them and their complements.

We further deepen this investigation by proving the existence of unary languages for which nondeterminism is useless in their recognition and in the recognition of their complements. In particular, we show that for each integer $n \geqslant 2$, there exists a language $\mathcal{L}$ such that all nfa's accepting either $\mathcal{L}$ or its complement must have at least the same number of states n as the minimum dfa's accepting them.

We complete the paper by showing a class of languages for which the complementation of nfa's can be easily done, without increasing the number of states, and we discuss the possibility of extending our main result.

2. Preliminaries

In this section, we recall some basic notions, notations and facts used in the paper.

Given a set S, $\sharp S$ denotes its cardinality, and 2^S the family of all its subsets. As usual, we let $\mathbf{N}$ ($\mathbf{N}^+$) be the set of nonnegative (positive) integers. The absolute value of a number z is denoted by $|z|$. For any $z > 0$, $\ln z$ denotes the natural logarithm of z.

In the paper, we will use some notions and results from number theory. We refer the reader to books on this subject for proofs and details, e.g., [5]. The *greatest common divisor* of integers $a_1, \ldots, a_s$ is denoted by $\gcd(a_1, \ldots, a_s)$. Their *least common multiple* is denoted by $\mathrm{lcm}(a_1, \ldots, a_s)$.

Throughout the paper, when we write that an integer $z > 1$ *factorizes* as $z = p_1^{k_1} \cdot p_2^{k_2} \cdot \cdots \cdot p_s^{k_s}$, we will refer to the factorization in prime powers given by the Fundamental Theorem of Arithmetic, where, for $i = 1, \ldots, s$, p_i is a prime number and $k_i \in \mathbf{N}^+$. We recall that such a factorization is unique up to a permutation of the indices.

In the paper, we will make use of linear Diophantine equations. We are not going to recall general results on this subject. For our purposes, the following lemma is enough (a proof for a more general result can be found in [16]):

Lemma 2.1. *Given two positive integers a, b the set*

$$\{ax - by \mid x, y \in \mathbf{N}\}$$

coincides with the set of all integer multiples of $\gcd(a, b)$.

Finally, we recall the following result, related to modular arithmetic:

Theorem 2.1 (*Chinese Remainder Theorem*). *Let $m_1, m_2, \ldots, m_s$ denote s positive integers that are pairwise relatively prime, and let $a_1, a_2, \ldots, a_s$ denote any s integers. Then the congruences $x \equiv a_i \ (\mathtt{MOD}\ m_i)$, $i = 1, \ldots, s$, have common solutions. Any two solutions are congruent modulo $m_1 \cdot m_2 \cdot \cdots \cdot m_s$.*

Given an alphabet Σ, Σ^* denotes the set of strings on Σ, with the empty string ε. Given a string $x \in \Sigma^*$, $|x|$ denotes its length. A language $\mathcal{L}$ is said to be *unary* (or *tally*) whenever it can be built over a *single-letter* alphabet. In this case, we let $\mathcal{L} \subseteq a^*$.

The *reversal* of a string $x = a_1 a_2 \ldots a_{|x|}$, $a_i \in \Sigma$, $i = 1, \ldots, |x|$, is the string $a_{|x|} \ldots a_2 a_1$ and it is denoted as x^{R}. The *reversal* of a language $\mathcal{L}$, denoted as $\mathcal{L}^{\mathrm{R}}$, is the set of the reversals of the strings belonging to $\mathcal{L}$. As usual, the complement of a language $\mathcal{L} \subseteq \Sigma^*$, i.e., the set $\Sigma^* \setminus \mathcal{L}$, will be denoted as $\mathcal{L}^{\mathrm{c}}$.

Let us take a brief look at the computational model of finite automata. For a detailed exposition, we refer the reader to [8]. A *one-way nfa* over an input alphabet Σ is a 4-tuple $A = (Q, \delta, q_0, F)$, where Q is the finite set of states, $\delta : Q \times \Sigma \to 2^Q$ is the

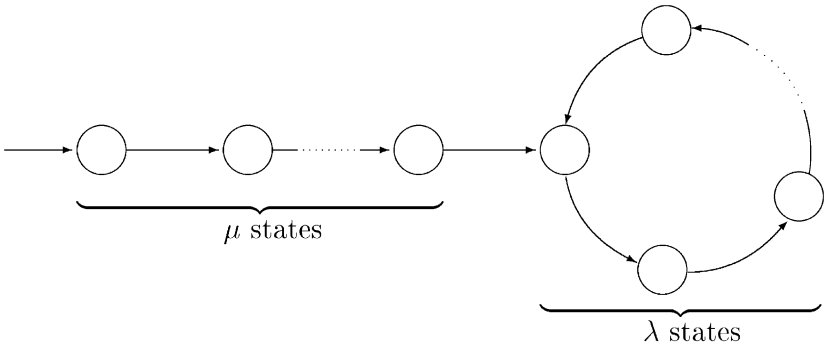

Fig. 1. A unary dfa.

transition function, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states. The transition function δ can be extended to strings in a standard way. The language accepted by A is the set $\mathcal{L}(A) = \{x \in \Sigma^* \mid \delta(q_0, x) \cap F \neq \emptyset\}$. The one-way automaton A is *dfa* if and only if $\sharp\delta(q, \sigma) = 1$ for any $q \in Q$ and $\sigma \in \Sigma$ (this implies that deterministic automata are assumed to be *complete*).

We recall that the *state complexity* of a regular language $\mathcal{L}$, written $\mathrm{sc}(\mathcal{L})$, is the number of states in the smallest deterministic finite automaton accepting $\mathcal{L}$ (see, e.g., [24]). The *nondeterministic state complexity* of $\mathcal{L}$, denoted by $\mathrm{nsc}(\mathcal{L})$, is the minimum number of states in nondeterministic automata accepting $\mathcal{L}$.

In this paper, we will consider mainly *unary automata*, namely, automata defined over a one-letter input alphabet $\Sigma = \{a\}$. It is easy to observe that the transition graph of a unary dfa consists of a path which starts from the initial state and is followed by a cycle of one or more states (see Fig. 1).

As in [2], the *size* of a unary nfa A is the pair (λ, μ), where $\lambda \geqslant 1$ and $\mu \geqslant 0$ denote the number of states belonging to the cycles and those not belonging to any cycle, respectively.

Observing the shape of unary dfa's it is not difficult to conclude that unary regular languages correspond to *ultimately periodic sets* of integers:

Theorem 2.2. *Given a unary regular language $\mathcal{L}$ and two integers $\lambda \geqslant 1$, $\mu \geqslant 0$, the following statements are equivalent*:

(i) *$\mathcal{L}$ is accepted by a dfa of size (λ, μ)*;

(ii) *for any integer $m \geqslant \mu$, $a^m \in \mathcal{L}$ if and only if $a^{m+\lambda} \in \mathcal{L}$.*

A unary dfa is said to be *cyclic* when its graph is a cycle, i.e., when it is of size $(\lambda, 0)$ for some $\lambda \geqslant 1$. To emphasize the periodicity of the accepted language $\mathcal{L}$ we also say that $\mathcal{L}$ is *λ-cyclic*. The language $\mathcal{L}$ is said to be *properly* λ-cyclic if and only if it is λ-cyclic but not λ'-cyclic for any $\lambda' < \lambda$, namely, the minimum dfa accepting $\mathcal{L}$ consists of a cycle of λ states.

A language $\mathcal{L}$ accepted by a unary dfa of size (λ, μ) is said to be *ultimately λ-cyclic*. $\mathcal{L}$ is said to be *properly* ultimately λ-cyclic if and only if it is ultimately λ-cyclic but not ultimately λ'-cyclic for any $\lambda' < \lambda$. This is equivalent to say that the length of the cycle in

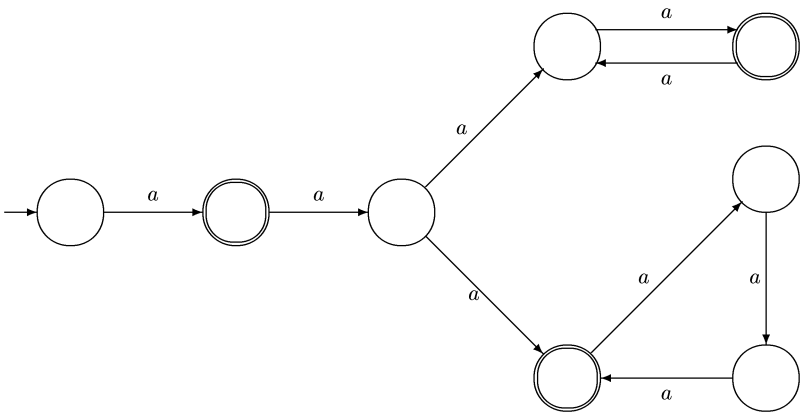

Fig. 2. An nfa in Chrobak normal form.

the minimum dfa accepting $\mathcal{L}$ is λ. For a properly ultimately λ-cyclic language the number λ is also called its *period*.

The following lower bound was proved for unary nfa's in [10] (the result was extended in the case of two-way automata in [15]):

Theorem 2.3. *Each nfa accepting a properly ultimately λ-cyclic language $\mathcal{L}$, where λ factorizes as $\lambda = p_1^{k_1} \cdot p_2^{k_2} \cdot \cdots \cdot p_s^{k_s}$, has at least $p_1^{k_1} + p_2^{k_2} + \cdots + p_s^{k_s}$ states in its cycles. Hence,* $\mathrm{nsc}(\mathcal{L}) \geqslant p_1^{k_1} + p_2^{k_2} + \cdots + p_s^{k_s}$.

An nfa is in *Chrobak normal form* if its graph consists of an initial path and a set of disjoint cycles. The last state of the path is connected via a leaving edge to one state in each cycle: this is the only nondeterministic choice in the automaton. In Fig. 2 an nfa in Chrobak normal form is depicted. The interest for this particular structure is related with the fact that, although making limited use of nondeterminism, each n-state unary nfa A can be transformed into an nfa A' in *Chrobak normal form* of size at most $(n, \mathrm{O}(n^2))$ [2]. Actually, a closer investigation of the proof for this result shows that the number of states in the cycles of A' can be bounded by that of the states in the cycles of A.

As shown in [2], given an automaton in Chrobak normal form an equivalent dfa can be obtained by just replacing the cycles with a unique cycle of length equal to the least common multiple of the lengths of the cycles in the given automaton. This implies that the period of the language accepted by an nfa in Chrobak normal form coincides with or divides the least common multiple of the lengths of the cycles.

Using this observation and the above mentioned fact that each n-state nfa A can be transformed into an nfa in Chrobak normal form of size bounded by $(n, \mathrm{O}(n^2))$, it turns out that A can be simulated by a dfa whose number of states is bounded by the following function (see [2] for the details):

$$F(n) = \max\{\mathrm{lcm}(x_1, \ldots, x_s) \mid x_1, \ldots, x_s \in \mathbf{N}^+ \text{ and } x_1 + \cdots + x_s = n\}.$$

Evaluating the growth rate of $F(n)$ is known as Landau's problem [12,13]. Several approximations for $F(n)$ are given in the literature, the best one known being contained in [22]. For our purposes it is enough to know that [1]

Lemma 2.2. $F(n) = e^{\Theta(\sqrt{n \ln n})}$.

3. Nondeterministic state complexity and language complements

It is well known that the simulation of nfa's by dfa's defined by the subset construction is optimal, i.e., for each integer n there is a language accepted by an n-state nfa such that the minimum dfa accepting it has 2^n states. As shown in [2] in the unary case the optimal cost of this simulation reduces to $F(n)$.

From another point of view, given an integer λ which factorizes as $\lambda = p_1^{k_1} \cdot p_2^{k_2} \cdot \cdots \cdot p_s^{k_s}$, each properly λ-cyclic language is accepted by a minimum dfa of λ states, while, as stated in Theorem 2.3, each nfa accepting it requires at least $p_1^{k_1} + p_2^{k_2} + \cdots + p_s^{k_s}$ states in the cycles. Given λ it is easy to exhibit a language matching this gap as, for instance, the language

$$\mathcal{L}_\lambda = \{a^m \mid \exists i, 1 \leqslant i \leqslant s,\ p_i^{k_i} | m\},$$

which is accepted by an nfa with s cycles of lengths $p_1^{k_1}, \ldots, p_s^{k_s}$.

On the other hand, it is possible to prove that the complement of this language requires λ states even to be accepted by an nfa (see, e.g., [6]). In other words, nondeterminism is useless in order to accept $\mathcal{L}_\lambda^c$.

Our main result generalizes this observation to *each* ultimately properly λ-cyclic language $\mathcal{L}$: if $\mathcal{L}$ has a "small" nfa then each nfa accepting $\mathcal{L}^c$ must have at least λ states.

Theorem 3.1. *Let $\mathcal{L}$ be a properly ultimately λ-cyclic unary language, where $\lambda > 1$ factorizes as $\lambda = p_1^{k_1} \cdot p_2^{k_2} \cdot \cdots \cdot p_s^{k_s}$. If $\mathcal{L}$ is accepted by an nfa with $p_1^{k_1} + p_2^{k_2} + \cdots + p_s^{k_s}$ states in the cycles, then each nfa accepting $\mathcal{L}^c$ contains a simple cycle of length a multiple of λ.*

Proof. Let A be the given nfa accepting $\mathcal{L}$ and A' be an nfa accepting $\mathcal{L}^c$.

We recall from Section 2 that any nfa can be turned into one in Chrobak normal form without increasing the number of its cyclic states. Furthermore, the period of the language accepted by an automaton in this form must divide the least common multiple of the cycle lengths. Since $\mathcal{L}$ is a properly ultimately λ-cyclic language, this allows us to suppose without loss of generality that the automaton A is in Chrobak normal form and it contains exactly s disjoint cycles, of lengths $p_1^{k_1}, p_2^{k_2}, \ldots, p_s^{k_s}$, respectively.

For $i = 1, \ldots, s$, let us consider the ith cycle of A, whose length is $p_i^{k_i}$. Let m_i be an integer such that a^{m_i} is accepted in a state belonging to this cycle and $a^{m_i + {p_i}^{k_i - 1}} \notin \mathcal{L}$. Such an m_i exists, since otherwise the cycle could be removed (if it does not contain any final

[1] The bound $F(n) = O(e^{\sqrt{n \ln n}})$ given in [2] was recently corrected by giving the estimate of Lemma 2.2.

state) or reduced to length ${p_i}^{k_i-1}$ (if for any a^z accepted in the cycle, the string $a^{z+{p_i}^{k_i-1}}$ belongs to $\mathcal{L}$) and $\mathcal{L}$ would not be properly ultimately λ-cyclic.

By Theorem 2.1, there exists an integer m such that, for $i = 1, \ldots, s$:

$$m \equiv m_i + {p_i}^{k_i-1} \ (\mathtt{MOD}\ p_i^{k_i}).$$

Furthermore, this integer m can be chosen so it is greater than the number of states of the automata A and A'. This last condition assures that any computation on a^m for A' must go through a cycle and that it must *end* in a cycle for A.

In particular, considering the automaton A, the states of the ith cycle reached on input a^m and on input $a^{m_i+{p_i}^{k_i-1}}$ coincide. Since $a^{m_i+{p_i}^{k_i-1}} \notin \mathcal{L}$, for $i = 1, \ldots, s$, this implies that $a^m \notin \mathcal{L}$. Hence, a^m is accepted by A'.

Let us now consider the length p of a simple cycle in A' that contains a state visited during an accepting computation on a^m. Then, for any $y \in \mathbf{N}$, it holds that $a^{m+py} \in \mathcal{L}^c$. We are going to prove that for all $i = 1, \ldots, s$, ${p_i}^{k_i}$ divides p. Since a^{m_i} is accepted by A in the ith cycle, we also have, for any $x \in \mathbf{N}$, $a^{m_i+{p_i}^{k_i}x} \in \mathcal{L}$. Now consider the equation

$$m + py = m_i + {p_i}^{k_i}x. \tag{1}$$

Since $m = m_i + h\,{p_i}^{k_i} + {p_i}^{k_i-1}$, with h an integer, Eq. (1) may be rewritten as

$$h\,{p_i}^{k_i} + {p_i}^{k_i-1} = {p_i}^{k_i}x - py.$$

By Lemma 2.1 ${p_i}^{k_i}x - py$ with x and y ranging over $\mathbf{N}$ yields exactly all multiples of $\gcd({p_i}^{k_i}, p)$. We thus have integer solutions for Eq. (1) if and only if $\gcd({p_i}^{k_i}, p)$ divides ${p_i}^{k_i-1}$, but having a solution, say m', for this equation implies both $a^{m'} \in \mathcal{L}$ and $a^{m'} \in \mathcal{L}^c$, which is a contradiction. Hence, in order to avoid that $\gcd({p_i}^{k_i}, p)$ divides ${p_i}^{k_i-1}$, we must have that ${p_i}^{k_i}$ divides p. This allows us to conclude that λ divides p, i.e., the nfa A' accepting the complement of $\mathcal{L}$ contains a simple cycle of length a multiple of λ. □

We observe that with $\lambda = 1$ the sum of the prime factors of λ is zero. If we have an nfa accepting $\mathcal{L}$ with a cycle of 0 states, then this means that $\mathcal{L}$ is finite and this obviously implies that an nfa accepting $\mathcal{L}^c$ must have a cycle of length at least λ. Thus, the result of Theorem 3.1 can be extended to this trivial case.

As a consequence of Theorem 3.1, if a properly ultimately λ-cyclic language has a small nondeterministic state complexity, then the nondeterministic state complexity of its complement coincides with the deterministic one, i.e., it is at least λ.

We now prove that the same result does not hold in the case of non-unary alphabets:

Theorem 3.2. *For each integer $n \geqslant 1$ there exists a language $\mathcal{L}$ such that*
- $\mathrm{nsc}(\mathcal{L}) = n$, *i.e., the smallest nfa accepting $\mathcal{L}$ has n states*;
- $\mathrm{sc}(\mathcal{L}) = 2^n$, *i.e., the minimum dfa accepting $\mathcal{L}$ has 2^n states*;
- $\mathrm{nsc}(\mathcal{L}^c) \leqslant n + 1$, *i.e., there is an $(n + 1)$-state nfa accepting the complement of $\mathcal{L}$.*

Proof. In [14] it is shown that given $n \geqslant 1$, there is a language X accepted by a dfa A with n states such that the reversal of X requires 2^n states. We point out that this automaton A has only one final state which coincides with the initial state. It is thus possible to get an

n-state nfa from A accepting X^{R}. By considering $\mathcal{L} = X^{\mathrm{R}}$, this implies that $\mathrm{nsc}(\mathcal{L}) = n$ and $\mathrm{sc}(\mathcal{L}) = 2^n$.

To complete the proof we want to find an $(n+1)$-state nfa accepting $\mathcal{L}^{\mathrm{c}}$. To this aim we observe that $\mathcal{L}^{\mathrm{c}} = \left(X^{\mathrm{R}}\right)^{\mathrm{c}} = (X^{\mathrm{c}})^{\mathrm{R}}$. By complementing the set of final states of A we can get an n-state dfa accepting X^{c}.

It is known than given an n-state (deterministic or nondeterministic) automaton A accepting a language $\mathcal{L}$, it is easy to build a nondeterministic automaton A^{R} accepting $\mathcal{L}^{\mathrm{R}}$. The states of A^{R} are the same as A plus one extra state (used as an initial state). Hence, A^{R} has $n+1$ states. However, if A contains only one final state, the extra state is superfluous: in this case A^{R} can be found with n states.

Thus, from the resulting automaton accepting X^{c} we can get an $(n+1)$-state nfa accepting the reversal, i.e., the language $\mathcal{L}^{\mathrm{c}} = (X^{\mathrm{c}})^{\mathrm{R}}$. This completes the proof. □

We remark that the automaton A presented in [14] used to prove Theorem 3.2 is defined over an alphabet of three symbols. The authors do not know whether or not Theorem 3.2 can be strengthened by considering witness languages defined over a two-letter alphabet. For all two-letter examples presented in the literature witnessing the optimality of the subset construction (see, e.g., [17,18]), the nondeterministic state complexity of the complement seems hard to compute. However, in the two-letter case we have an example which is close to the gap: the language $\{a,b\}^* a \{a,b\}^{n-1}$ is accepted by a $(n+1)$-state nfa, while the minimum dfa accepting it has 2^n states (see, e.g., [6]). Furthermore, it is not difficult to show that its complement is accepted by an nfa with $2n+1$ states.

Whereas Theorem 3.1 states that if nondeterminism allows to obtain a very small nfa in the recognition of a language, then it is useless in that of its complement, we now show that there are unary languages such that nondeterminism can help in neither case.

Theorem 3.3. *Given an integer $\lambda > 1$ factorizing as $\lambda = p_1^{k_1} \cdot p_2^{k_2} \cdot \cdots \cdot p_s^{k_s}$, consider the language*

$$\mathcal{L} = \{a^m \mid \#\{i \mid p_i^{k_i} \; divides \; m\} \; is \; even\}.$$

Each nfa accepting $\mathcal{L}$ and each nfa accepting $\mathcal{L}^{\mathrm{c}}$ contains a simple cycle of length a multiple of λ.

Proof. Let A be an nfa accepting the language $\mathcal{L}$. We split the proof in two parts, depending on whether the number s is odd or even.

If s is even, then the word $a^{H\lambda}$ belongs to the language $\mathcal{L}$ for each $H \in \mathbf{N}$.

Let H be sufficiently large, in such a way that $a^{H\lambda}$ is accepted after going through a cycle, hence after visiting at least one state belonging to a simple cycle. Let ℓ be the length of this simple cycle. Then, for each $h \geqslant 0$, $a^{H\lambda+h\ell}$ is accepted by A, i.e., $a^{H\lambda+h\ell} \in \mathcal{L}$.

We prove that each prime power $p_j^{k_j}$ divides ℓ, for $j = 1, \ldots, s$. To this aim we consider the number

$$h_j = \frac{\lambda}{p_j^{k_j}} = p_1^{k_1} \cdot \cdots \cdot p_{j-1}^{k_{j-1}} \cdot p_{j+1}^{k_{j+1}} \cdot \cdots \cdot p_s^{k_s}. \tag{2}$$

We observe that for $i \neq j$, $p_i^{k_i}$ divides both λ and h_j and, hence, divides $H\lambda + h_j\ell$ too. If $p_j^{k_j}$ does not divide ℓ then it does not divide $H\lambda + h_j\ell$, namely, $H\lambda + h_j\ell$ has an odd number of

divisors among $p_1^{k_1}, \ldots, p_s^{k_s}$, which would imply the contradiction $a^{H\lambda+h_j\ell} \notin \mathcal{L}$. Hence, each $p_j^{k_j}$ divides ℓ, i.e., ℓ is a multiple of λ.

For s odd we consider the number

$$t = \frac{\lambda}{p_s} = p_1^{k_1} \cdot \cdots \cdot p_{s-1}^{k_{s-1}} \cdot p_s^{k_s-1}.$$

For any integer H such that $\gcd(H, \lambda) = 1$ the word a^{Ht} belongs to $\mathcal{L}$. Like in the previous case, let H and ℓ be such that a^{Ht} is accepted by crossing a simple cycle of length ℓ. Then each word $a^{Ht+h\ell}$ must belong to $\mathcal{L}$ for $h \geqslant 0$.

Also in this case we will prove that each prime power $p_j^{k_j}$ divides ℓ, for $j = 1, \ldots, s$. For $j < s$ we consider h_j as in (2). We observe that $p_s^{k_s}$ divides $h_j\ell$ but does not divide Ht. Hence $p_s^{k_s}$ does not divide $Ht + h_j\ell$. On the other hand, for each $i \neq j$, $p_i^{k_i}$ does divide $Ht + h_j\ell$. Since $a^{Ht+h_j\ell} \in \mathcal{L}$, $p_j^{k_j}$ must divide $Ht + h_j\ell$. Since p_j does not divide h_j, but $p_j^{k_j}$ divides t, this implies that $p_j^{k_j}$ divides ℓ, for $j = 1, \ldots, s-1$.

Now we consider $j = s$. Let k be the integer such that $\gcd(p_s^{k_s}, \ell) = p_s^k$. Hence, $k \leqslant k_s$ and p_s^k divides ℓ. We will show that $k = k_s$. Suppose, by contradiction, that $k < k_s$. Then, $\gcd(p_s^{k_s}, \ell)$ divides Ht and by Lemma 2.1 there are nonnegative integers $\hat{h}, \hat{k}$ such that the equality $Ht + \hat{h}\ell = \hat{k}p_s^{k_s}$ holds. $Ht + \hat{h}\ell$ is thus a multiple of $p_s^{k_s}$. Since it is also a multiple of all other prime powers $p_j^{k_j}$ this implies the contradiction $a^{Ht+\hat{h}\ell} \notin \mathcal{L}$. Hence ℓ is a multiple of $p_s^{k_s}$. This permits us to conclude that λ divides ℓ.

The proof for nfa's accepting $\mathcal{L}^c$ can be obtained just exchanging the case of s odd with the case of s even. □

Among unary regular languages with nondeterministic state complexity equal to the deterministic one, each language $\mathcal{L}$ considered in Theorem 3.3 seems to be harder in the sense that its complement has high order nondeterministic state complexity too. On the other hand, if we consider two-way deterministic or nondeterministic automata it is not difficult to see that the language $\mathcal{L}$ can be accepted by using $O(p_1^{k_1} + \cdots + p_s^{k_s})$ states.

4. Conclusions

It is quite natural to try to extend Theorem 3.1 in order to give a general result relating the nondeterministic state complexities of a unary regular language and its complement. This does not seem to be very easy.

We briefly discuss the following examples:

$$\mathcal{L}' = \{a^m \mid \exists k \geqslant 0\ (m = 4k \vee m = 6k)\},$$
$$\mathcal{L}'' = \{a^m \mid \exists k \geqslant 0\ (m = 4k \vee m = 6k + 1)\}.$$

Both $\mathcal{L}'$ and $\mathcal{L}''$ have period 12. For both the languages it is possible to build an accepting nfa in Chrobak normal form consisting of an initial path containing only one state and two cycles of lengths 4 and 6, respectively.

The complement of $\mathcal{L}'$ is the language

$$\mathcal{L}'^{\mathrm{c}} = \{a^m \mid 6 \text{ and } 4 \text{ do not divide } m\}.$$

It is possible to verify that each nondeterministic automaton accepting it has a cycle of 12 states. Then the smallest nfa for $\mathcal{L}'^{\mathrm{c}}$ coincides with the minimum dfa for it, defined by a cycle of 12 states.

We now show that in the case of $\mathcal{L}''$ nondeterminism allows both the language and its complement to be accepted by nfa's smaller than the minimum dfa.

We observe that $\mathcal{L}''$ is the disjoint union of the following two cyclic languages:

$$\mathcal{L}_1 = \{a^m \mid \exists k \geqslant 0\ (m = 4k)\},$$
$$\mathcal{L}_2 = \{a^m \mid \exists k \geqslant 0\ (m = 6k + 1)\}.$$

In particular $\mathcal{L}_1 = \mathcal{L}'' \cap EVEN$ and $\mathcal{L}_2 = \mathcal{L}'' \cap ODD$, where $EVEN = \{aa\}^*$ and $ODD = a\{aa\}^*$. Using these equalities we can derive that $\mathcal{L}''^{\mathrm{c}}$ can be expressed by complementing $\mathcal{L}_1$ with respect to the set *EVEN* and by complementing $\mathcal{L}_2$ with respect to the set *ODD*:

$$\begin{aligned}\mathcal{L}''^{\mathrm{c}} &= \left(\mathcal{L}_1^{\mathrm{c}} \cap EVEN\right) \cup \left(\mathcal{L}_2^{\mathrm{c}} \cap ODD\right)\\ &= \{a^m \mid \exists k \geqslant 0\ (m = 4k + 2 \vee m = 6k + 3 \vee m = 6k + 5)\}.\end{aligned}$$

From these equalities it is easy to conclude that $\mathcal{L}''^{\mathrm{c}}$, like $\mathcal{L}''$, can be accepted by an nfa with two cycles of lengths 4 and 6. The only difference is, of course, in the set of final states. In particular, in each cycle, the set of final states is chosen by complementing that of the corresponding cycle in the nfa accepting $\mathcal{L}''$ with respect to the set *EVEN* or *ODD*.

We now generalize this example in order to consider languages definable with respect to suitable partitions.

Let $\mathcal{L} \subseteq \{a\}^*$ be a unary language and $\mathcal{P} = \{P_1, P_2, \ldots, P_s\}$ a partition of $\{a\}^*$, i.e., $\bigcup_{i=1}^{s} P_i = \{a\}^*$, and $P_i \cap P_j = \emptyset$ for $i, j = 1, \ldots, s$ with $i \neq j$. Let $\mathcal{L}_1, \ldots, \mathcal{L}_s$ be the languages obtained by splitting $\mathcal{L}$ according to the partition $\mathcal{P}$, i.e., $\mathcal{L}_i = \mathcal{L} \cap P_i$, for $i = 1, \ldots, s$. Hence we can write

$$\mathcal{L} = \bigcup_{i=1}^{s} \mathcal{L}_i = \bigcup_{i=1}^{s} \mathcal{L}_i \cap P_i. \tag{3}$$

From these equalities, using the fact that $\mathcal{P}$ is a partition of $\{a\}^*$, we can prove that the complement of $\mathcal{L}$ can be expressed as follows:

$$\mathcal{L}^{\mathrm{c}} = \bigcup_{i=1}^{s} \mathcal{L}_i^{\mathrm{c}} \cap P_i. \tag{4}$$

We are now going to use equalities (3) and (4) in order to show a class of unary languages for which nfa's can be easily complemented, without increasing the number of states.

In particular, we consider languages $\mathcal{L}$ that may be described as the union of properly cyclic languages $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_s$ of periods $\lambda_1, \lambda_2, \ldots, \lambda_s$ such that a partition

$\mathcal{P} = \{P_1, \ldots, P_s\}$ of $\{a\}^*$ can be found with the following properties:
- for $i = 1, \ldots, s$, P_i is a cyclic language;
- for $i = 1, \ldots, s$, $\mathcal{L}_i \subseteq P_i$, i.e., $\mathcal{L}_i = \mathcal{L} \cap P_i$;
- for $i = 1, \ldots, s$, the period of P_i divides λ_i.

If these conditions hold, then it is possible to build nfa's accepting $\mathcal{L}$ and $\mathcal{L}^c$ with the same underlying graph. In particular, this graph consists of s independent cycles of lengths $\lambda_1, \lambda_2, \ldots, \lambda_s$ and one extra state, the initial state, connected via a leaving edge to each cycle. In other words, this graph corresponds to the Chrobak normal form, where the initial path consists only of the initial state.

In the automaton A for $\mathcal{L}$, the ith cycle is used to accept the language $\mathcal{L}_i$; so the final states are chosen according to this language. In the automaton A_c for $\mathcal{L}^c$ the final states on the ith cycle are obtained by just complementing this choice with respect to the set P_i. More precisely, a state q on the ith cycle is final in A if and only if it is reachable by reading a word $a^m \in \mathcal{L}_i$ (this implies also that $a^m \in P_i$). In the automaton A_c the state q is final if and only if it is reachable by reading a word $a^m \in P_i$ such that $a^m \notin \mathcal{L}_i$.

The two examples presented at the beginning of this section display two opposite situations. Actually, it is possible to present many very different situations. Hence it seems to be difficult to get a precise relationship between the sizes of a minimal nfa accepting a language and its complement, generalizing that of Theorem 3.1.

Acknowledgements

The authors wish to thank the anonymous referees for their many, accurate suggestions.

References

[1] J.C. Birget, Partial orders on words, minimal elements of regular languages, and state complexity, Theoret. Comput. Sci. 119 (1993) 267–291 Erratum available at http://clam.rutgers.edu/~birget/papers.html.
[2] M. Chrobak, Finite automata and unary languages, Theoret. Comput. Sci. 47 (1986) 149–158 Erratum Theoret. Comput. Sci. 302 (2003) 497–498.
[3] K. Ellul, Descriptional complexity measures of regular languages, M. Math. Thesis, University of Waterloo, 2002.
[4] J. Goldstine, M. Kappes, C. Kintala, H. Leung, A. Malcher, D. Wotschke, Descriptional complexity of machines with limited resources, J. Univ. Comput. Sci. 8 (2) (2002) 193–234.
[5] G.H. Hardy, E.M. Wright, An Introduction to the Theory of Numbers, fifth ed., Oxford Science Publications, Oxford, 1979.
[6] M. Holzer, M. Kutrib, Unary language operations and their nondeterministic state complexity, Developments in Language Theory, DLT 2002, Lecture Notes in Computer Science, vol. 2459, 2003, pp. 162–172.
[7] M. Holzer, M. Kutrib, Nondeterministic descriptional complexity of regular languages, Internat. J. Found. Comput. Sci. 14 (2003) 1087–1102.
[8] J.E. Hopcroft, R. Motwani, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, second ed., Addison-Wesley, Reading, MA, 2001.
[9] J. Hromkovič, Descriptional complexity of regular languages (concepts and open problems), Proc. Descriptional Complexity of Automata, Grammars and Related Structures, Vienna, 2001, Report 16-2001, Otto-Von-Guericke-Universität Magdeburg, pp. 11–13.
[10] T. Jiang, E. McDowell, B. Ravikumar, The structure and complexity of minimal nfa's over a unary alphabet, Internat. J. Found. Comput. Sci. 2 (1991) 163–182.

[11] G. Jirásková, State complexity of some operations on regular languages, Proc. Descriptional Complexity of Formal Systems 2003, MTA SZTAKI, Budapest, pp. 114–125.
[12] E. Landau, Über die maximalordnung der permutationen gegebenen grades, Arch. Math. Phys. 3 (1903) 92–103.
[13] E. Landau, Handbuch der Lehre von der Verteilung der Primzahlen I, Teubner, Leipzig, Berlin, 1909.
[14] E. Leiss, Succinct representation of regular languages by boolean automata, Theoret. Comput. Sci. 13 (1981) 323–330.
[15] C. Mereghetti, G. Pighizzini, Two-way automata simulations and unary languages, J. Automat. Languages Combin. 5 (2000) 287–300.
[16] C. Mereghetti, G. Pighizzini, Optimal simulations between unary automata, SIAM J. Comput. 30 (2001) 1976–1992.
[17] A. Meyer, M. Fischer, Economy of description by automata, grammars, and formal systems, in: Proc. 12th Annual IEEE Symp. Switching and Automata Theory, 1971, pp. 188–191.
[18] F. Moore, On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic and two-way finite automata by deterministic automata, IEEE Trans. Comput. C-20 (1971) 1211–1219.
[19] C. Nicaud, Average state complexity of operations on unary automata, Proc. MFCS'99, Lecture Notes in Computer Science, vol. 1672, 1999, pp. 231–240.
[20] G. Pighizzini, J.O. Shallit, Unary languages operations, state complexity and Jacobsthal's function, Internat. J. Found. Comput. Sci. 13 (2002) 145–159.
[21] M.O. Rabin, D. Scott, Finite automata and their decision problems, IBM J. Res. Develop. 3 (1959) 198–200 also in: E.F. Moore (Ed.), Sequential Machines: Selected Papers, Addison-Wesley, Reading, MA, 1964, pp. 63–91.
[22] M. Szalay, On the maximal order in S_n and S_n^*, Acta Arith. 37 (1980) 321–331.
[23] S. Yu, A renaissance of automata theory?, EATCS Bull. 72 (2000) 270–272.
[24] S. Yu, State complexity of regular languages, J. Automat. Languages Combin. 6 (2001) 221–234.