Theoretical Computer Science 449 (2012) 85–92

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Reversal of binary regular languages[☆]

Galina Jirásková[a,*], Juraj Šebej[b]

[a] Mathematical Institute, Slovak Academy of Sciences, Grešákova 6, 040 01 Košice, Slovakia
[b] Institute of Computer Science, P. J. Šafárik University, Jesenná 5, 041 54 Košice, Slovakia

ARTICLE INFO

Keywords:
Regular languages
Reversal
Descriptional complexity
Deterministic union-free languages

ABSTRACT

We present binary deterministic finite automata of n states that meet the upper bound 2^n on the state complexity of reversal. The automata have a single final state and are one-cycle-free-path automata; thus the witness languages are deterministic union-free languages. This result allows us to describe a binary language such that the nondeterministic state complexity of the language and of its complement is n and $n+1$, respectively, while the state complexity of the language is 2^n. Next, we show that if the nondeterministic state complexity of a language and of its complement is n, then the state complexity of the language cannot be 2^n. We also provide lower and upper bounds on the state complexity of such a language.

1. Introduction

Reversal is an operation on formal languages defined as $L^R = \{w^R \mid w \in L\}$, where w^R stands for the string w written backwards. The operation preserves regularity, as shown already by Rabin and Scott in 1959 [11]: A nondeterministic finite automaton for the reverse of a regular language can be obtained from an automaton recognizing the given language by swapping the role of initial and final states, and by reversing the transitions. This gives the upper bound 2^n on the state complexity of reversal, that is, on the number of states that are sufficient and necessary in the worst case for a deterministic finite automaton to accept the reversal of a language represented by a deterministic finite automaton of n states.

In 1966, Mirkin [10] pointed out that Lupanov's ternary witness automaton for nfa-to-dfa conversion presented in 1963 in [8] proves the tightness of the upper bound 2^n for reversal in the ternary case, since the ternary nondeterministic automaton is a reverse of a deterministic automaton.

Another ternary worst-case example for the reversal was given in 1981 by Leiss [7], who also proved the tightness of the upper bound in the binary case. However, his binary automata have $n/2$ final states.

In 2004, the authors of [12, Theorem 3] claimed a binary worst-case example with a single accepting state. Unfortunately, the result does not hold: when $n = 8$, with the initial and final state 1, the number of reachable states in the subset automaton for the reversal is 252 instead of 256. A similar problem arises whenever $n = 8 + 4k$ for a non-negative integer k. This result has been used in the literature several times, so our first aim is to present correct binary witness automata with a single final state.

We start with an observation that all the states in the subset automaton corresponding to the reverse of a minimal deterministic automaton are pairwise distinguishable [1,10]. This allows us to avoid the proof of distinguishability of states throughout the paper.

[☆] A preliminary version of this paper was presented at the 13th International Workshop on Descriptional Complexity of Formal Systems (DCFS 2011) held in Limburg, Germany, on 25–27 July 2011.

[*] Corresponding author. Tel.: +421 556228291.
E-mail addresses: jiraskov@saske.sk (G. Jirásková), juraj.sebej@gmail.com (J. Šebej).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2012.05.008

Then we present a ternary worst-case example with a very simple proof of reachability of all the subsets. In a more difficult way, we prove our main result that the upper bound 2^n can be met by a binary n-state deterministic finite automaton with a single final state. Our witness automata are uniformly defined for all integers n, and can be used in all the cases where the incorrect result from [12] has been used.

The binary witnesses allow us to prove the tightness of the upper bound also in the case of reversal of deterministic union-free languages, that is, languages represented by so called one-cycle-free-path deterministic automata, in which from each state there exists exactly one cycle-free accepting path [5]. This was our first motivation for finding binary worst-case examples with a single final state. We also need such examples to be able to describe a *binary* language such that the nondeterministic state complexity of the language and of its complement is n and $n + 1$, respectively, while the state complexity of the language (and of its complement) is 2^n. In both these cases, the existence of a single final state in the binary witness automaton for reversal is crucial. In the latter case, we also prove that, except when $n = 2$, there is no regular language such that both the language and its complement have nondeterministic state complexity n, while the state complexity of the language would be 2^n. We also provide lower and upper bounds on the state complexity of such a language, and conclude the paper with some open problems.

2. Preliminaries

This section gives some basic definitions, notation, and preliminary results used throughout the paper. For further details, we refer the reader to [13,15].

Let Σ be a finite alphabet, and let Σ^* be the set of all strings over the alphabet Σ including the empty string ε. A language is any subset of Σ^*. For a language L, we denote by L^c the complement of L, that is, the language $\Sigma^* \setminus L$. The cardinality of a finite set A is denoted by $|A|$, and its power-set by 2^A.

A *deterministic finite automaton* (dfa) is a quintuple $M = (Q, \Sigma, \delta, s, F)$, where Q is a finite set of states, Σ is a finite input alphabet, δ is the transition function that maps $Q \times \Sigma$ to Q, s is the initial (start) state, $s \in Q$, and F is the set of final states, $F \subseteq Q$. In this paper, all dfas are assumed to be complete. The transition function δ is extended to the domain $Q \times \Sigma^*$ in a natural way. The *language accepted by* dfa M is the set $L(M) = \{w \in \Sigma^* \mid \delta(s, w) \in F\}$. Two states of a dfa are distinguishable if there exists a string w which is accepted from one of the states and rejected from the other. Otherwise, the two states are equivalent.

A *nondeterministic finite automaton* (nfa) is a quintuple $M = (Q, \Sigma, \delta, S, F)$, where Q, Σ, and F are defined the same way as in a dfa, S is the set of initial states, and δ is the nondeterministic transition function that maps $Q \times \Sigma$ to 2^Q. The transition function can be naturally extended to the domain $2^Q \times \Sigma^*$. The *language accepted by* nfa M is the set of strings $L(M) = \{w \in \Sigma^* \mid \delta(S, w) \cap F \neq \emptyset\}$.

Two automata are *equivalent* if they accept the same language. A dfa (an nfa) M is *minimal* if every dfa (respectively, every nfa) that is equivalent to M has at least as many states as M. It is well known that a dfa is minimal if and only if all its states are reachable from the initial state, and no two different states are equivalent.

The *state complexity* of a regular language L, $\mathrm{sc}(L)$, is the number of states in the minimal dfa accepting language L. The *nondeterministic state complexity* of a regular language L, $\mathrm{nsc}(L)$, is defined as the number of states in a minimal nfa with a single initial state for language L.

An nfa $(Q, \Sigma, \delta, S, F)$ can be converted to an equivalent dfa $(2^Q, \Sigma, \delta', S, F')$ by the subset construction [11]. The transition function δ' is defined by $\delta'(R, a) = \bigcup_{r \in R} \delta(r, a)$, and a state R in 2^Q is in F' if $R \cap F \neq \emptyset$. We call the resulting dfa the *subset automaton* corresponding to the given nfa. The subset automaton need not be minimal, since some states may be unreachable or equivalent.

The *reverse* w^R *of a string* w is defined as follows: $\varepsilon^R = \varepsilon$ and if $w = va$ for a string v in Σ^* and a symbol a in Σ, then $w^R = av^R$. The *reverse of a language* L is the language $L^R = \{w^R \mid w \in L\}$. The *reverse of a dfa* $A = (Q, \Sigma, \delta, s, F)$ is the nfa A^R obtained from A by reversing all the transitions and by swapping the roles of the initial and final states, that is, $A^R = (Q, \Sigma, \delta^R, F, \{s\})$, where $\delta^R(q, a) = \{p \in Q \mid \delta(p, a) = q\}$.

Proposition 1. *The reverse of a dfa A recognizes the language $L(A)^R$.*

Proposition 2. *Let L be a regular language with* $\mathrm{sc}(L) = n$. *Then* $\mathrm{sc}(L^R) \leqslant 2^n$.

Proof. The reverse A^R of the minimal dfa A for language L is an n-state nfa (possibly with multiple initial states) for language L^R. After applying the subset construction to nfa A^R, we get a dfa for language L^R of at most 2^n states. □

For the sake of completeness, we give a short proof of the fact that, in the subset automaton corresponding to the reverse of a minimal deterministic finite automaton, all states are pairwise distinguishable [1,4,10].

Proposition 3 (*[1,4,10]*)**.** *All the states of the subset automaton corresponding to the reverse of a minimal dfa are pairwise distinguishable.*

Proof. Let A^R be the reverse of a minimal dfa A, and let q be an arbitrary state of nfa A^R. Since state q is reachable in dfa A, there is a string w_q that is accepted in nfa A^R from state q. Moreover, string w_q is not accepted by nfa A^R from any other state, because otherwise there would be two distinct computations of dfa A on string w_q^R. It follows that, in the subset automaton corresponding to nfa A^R, all the states are pairwise distinguishable: two distinct subsets of the state set of nfa A^R must differ in a state q of the nfa, and therefore string w_q distinguishes the two subsets. □

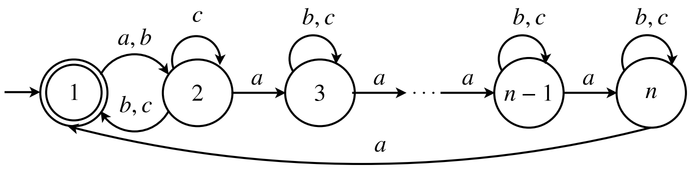

Fig. 1. Lupanov's ternary worst-case example for nfa-to-dfa conversion; the nfa is the reverse of a dfa.

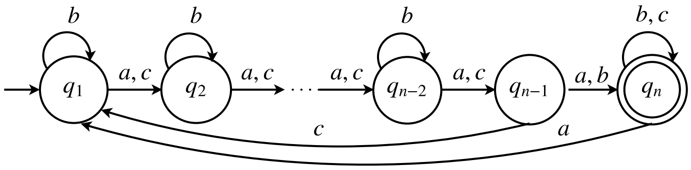

Fig. 2. The ternary n-state dfa meeting the upper bound 2^n for reversal.

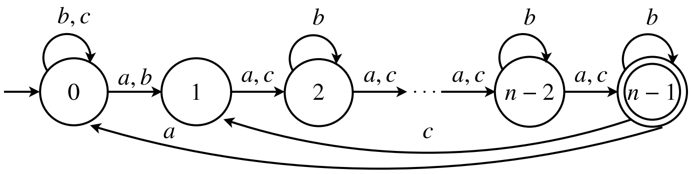

Fig. 3. The reverse of the dfa from Fig. 2; states renamed.

We first present a ternary regular language meeting the upper bound 2^n for reversal with a very easy proof of reachability of all the subsets. Let us recall that, in 1966, Mirkin [10] pointed out that the ternary Lupanov worst-case automaton for nfa-to-dfa conversion published in 1963 in [8] and shown in Fig. 1 is the reverse of a dfa. It follows that the bound 2^n for reversal is tight for alphabets of size at least three. The ternary worst-case example presented in 1994 by Yu et al. [16] with b and c interchanged is the same as Lupanov's example. A similar ternary witness language was given in 1981 by Leiss [7]. The proofs of reachability of all the subsets in the corresponding subset automata rely on the fact that all the permutations can be generated by the cyclic permutation and the transposition. The following example is a bit different.

Proposition 4 (*Ternary Worst-Case Example*)**.** *Let $n \geqslant 2$. There exists a ternary regular language L with $\mathrm{sc}(L) = n$ such that $\mathrm{sc}(L^R) = 2^n$.*

Proof. Consider the ternary language accepted by the dfa of Fig. 2. Construct an nfa for the reverse of the recognized language from the dfa by reversing all the transitions, and swapping the roles of the initial and final states. For the purpose of the proof, rename its states using numbers $0, 1, \ldots, n-1$, as shown in Fig. 3.

Let us show that the corresponding subset automaton has 2^n reachable states. The proof is by induction on the size of subsets of the state set $\{0, 1, \ldots, n-1\}$. All the singleton sets are reached from the initial state $\{0\}$ by strings in a^*, and the empty set is reached from state $\{1\}$ by b. Next, each set $\{i_1, i_2, \ldots, i_k\}$ of size k, where $2 \leqslant k \leqslant n$ and $0 \leqslant i_1 < i_2 < \cdots < i_k \leqslant n-1$, is reached from the set $\{0, i_3 - i_2 + 1, \ldots, i_k - i_2 + 1\}$ of size $k-1$ by string $bc^{i_2-i_1-1}a^{i_1}$. All the states are pairwise distinguishable by Proposition 3. □

3. Binary worst-case languages for reversal

The aim of this section is to present a binary n-state dfa with a single final state meeting the upper bound 2^n for reversal. Recall that our motivation for having just one final state comes from the research on self-verifying automata [6] and union-free languages [5]. A binary worst-case dfa with multiple final states was given by Leiss [7] in 1981; Fig. 4 shows the dfa when $n = 6$. Notice that here every other state is final, so it is rather inconvenient to display the figure in the general case.

The paper [12] presents the binary dfa scheme depicted in Fig. 5, and claims that every dfa A resulting from the scheme and satisfying $L(A) \neq \emptyset$ and $L(A) \neq \Sigma^*$ requires 2^n deterministic states for the reverse of language $L(A)$. Unfortunately, the example does not work: when $n = 8$, and with the initial and single accepting state 1, the resulting dfa has 252 reachable states instead of 256: notice that subsets $\{1, 4, 5, 8\}$, $\{8, 3, 4, 7\}$, $\{7, 2, 3, 6\}$, $\{6, 1, 2, 5\}$, are not reachable in the subset automaton corresponding to the reverse of the dfa, since each of them contains exactly one of the states 1 and 3, and therefore cannot be reached from any subset by b. Moreover, in the subset automaton, we have $\{6, 1, 2, 5\} \xrightarrow{a} \{7, 2, 3, 6\} \xrightarrow{a} \{8, 3, 4, 7\} \xrightarrow{a} \{1, 4, 5, 8\} \xrightarrow{a} \{6, 1, 2, 5\}$, and no other state goes to one of the four sets by a. It follows that none of these sets is reachable. A similar argument holds for every integer n with $n = 8 + 4k$, where the set $\{1, 4, 5, 8, 9, \ldots, n-4, n-3, n\}$ and all its shifts by a are not reachable.

The correct binary worst-case dfas with a single final state, uniformly defined for every n with $n \geqslant 2$, have been recently presented by Šebej in [14]. The next theorem gives an alternative proof for slightly modified Šebej automata. Notice that the

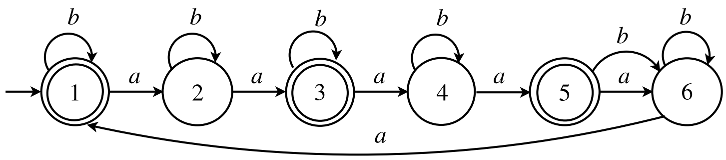

Fig. 4. Leiss's binary worst-case example for reversal with multiple final states; $n = 6$.

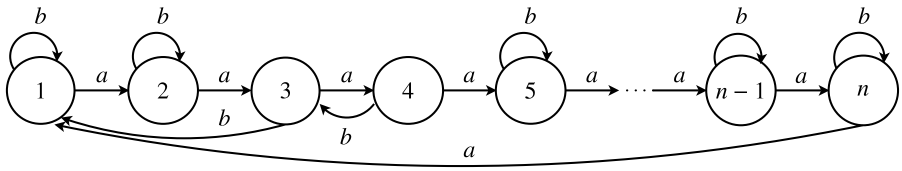

Fig. 5. The dfa scheme from [12, Theorem 3].

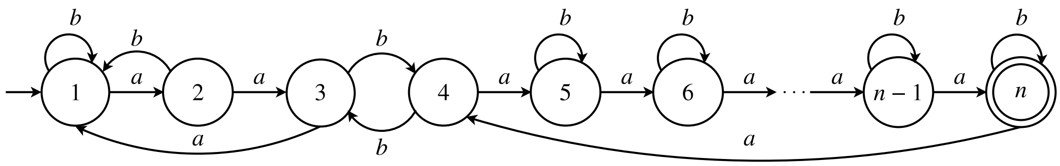

Fig. 6. The binary n-state dfa with a single final state meeting the upper bound 2^n for reversal.

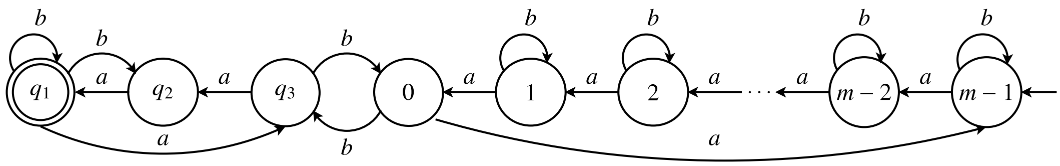

Fig. 7. The reverse of the dfa from Fig. 6; states renamed.

dfas in the theorem below are so-called one-cycle-free-path dfas, that is, from each state of the dfa, there exists exactly one cycle-free accepting path. Therefore, the resulting languages are deterministic union-free languages [5]. This shows that the upper bound 2^n on reversal is met by *binary deterministic union-free languages*, which improves a result from [5].

Theorem 5 (*Binary Worst-Case Example*)**.** *Let $n \geqslant 2$. There exists a binary (deterministic union-free) regular language L with $\mathrm{sc}(L) = n$ such that $\mathrm{sc}(L^R) = 2^n$.*

Proof. Consider the binary n-state dfa A of Fig. 6 with states $1, 2, \ldots, n$, of which 1 is the initial state, and state n is the single final state. By a, state 3 goes to state 1, state n goes to state 4, and every other state i goes to state $i + 1$. By b, state 2 goes to state 1, state 3 goes to state 4, state 4 goes to state 3, and every other state goes to itself.

If $n = 2$, then state 2 goes to 1 by a. If $n = 3$, then state 3 goes to itself by b. In these two cases, after applying subset construction to the reverse of dfa A, we get a four-state minimal dfa if $n = 2$, and an eight-state minimal dfa if $n = 3$.

Now, let $n \geqslant 4$. Construct an nfa for the reverse of language $L(A)$ from dfa A by swapping the roles of the initial and final states, and by reversing all the transitions. To simplify the proof, let us rename the states of the resulting nfa as shown in Fig. 7. The first three states are now denoted by q_1, q_2, and q_3, and the remaining states are numbered by $0, 1, \ldots, m - 1$, with $m = n - 3$. We are going to show that the corresponding subset automaton has 2^n reachable states.

Notice that, in the nfa, by bb state q_1 goes to $\{q_1, q_2\}$, state q_2 goes to the empty set, and every other state goes to itself.

In the corresponding subset automaton, all the singleton sets are reached from the initial state $\{m - 1\}$ by strings in a^*ba^*, and the empty set is reached from state $\{q_2\}$ by b.

Now, we show that every state $\{q_1, q_2, q_3\} \cup X$ with $X \subseteq \{0, 1, \ldots, m - 1\}$ is reachable. The proof is by induction on the size of X. The set $\{q_1, q_2, q_3\}$ is reached from state $\{q_1\}$ by $babb$. State $\{q_1, q_2, q_3\}$ goes by b to state $\{q_1, q_2\} \cup \{0\}$, and then by string abb to state $\{q_1, q_2, q_3\} \cup \{m - 1\}$, from which every state $\{q_1, q_2, q_3\} \cup \{j\}$ with $0 \leqslant j \leqslant m - 1$ can be reached by a string in a^*. Next, every state $\{q_1, q_2, q_3\} \cup \{i_1, i_2, \ldots, i_k\}$, where $2 \leqslant k \leqslant m$ and $0 \leqslant i_1 < i_2 < \cdots < i_k \leqslant m - 1$, is reached from state $\{q_1, q_2, q_3\} \cup \{i_2 - i_1, \ldots, i_k - i_1\}$ by $babba^{m-1-i_1}$. Since the latter state is reachable by the induction hypothesis, the former state is reachable as well. It follows that every state $\{q_1, q_2, q_3\} \cup X$ with $X \subseteq \{0, 1, \ldots, m - 1\}$ is reachable.

Next, we show that every state $\{q_3\} \cup X$ with $X \subseteq \{0, 1, \ldots, m - 1\}$ is reachable. For a subset X of $\{0, 1, \ldots, m - 1\}$ and a number i, denote by $X \oplus i$ the set that goes to X by a^i, that is, $X \oplus i = \{(x + i) \bmod m \mid x \in X\}$. We have $|X \oplus i| = |X|$. Consider the following five cases (here the arithmetic is modulo m).

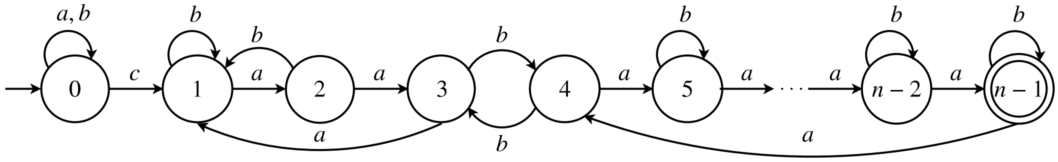

Fig. 8. The dfa for a left ideal meeting the bound $2^{n-1}+1$ for reversal; unspecified transitions on c go to state 1.

(i) Let $m-2 \in X$. Take $X' = X \setminus \{m-2\}$. Then $0 \notin X' \oplus 2$. Therefore, state $\{q_1, q_2, q_3\} \cup (X' \oplus 2)$, which is reachable as shown above, goes to state $\{q_3\} \cup X$ by *baabb*, since we have

$$\{q_1, q_2, q_3\} \cup (X' \oplus 2) \xrightarrow{b} \{q_1, q_2\} \cup \{0\} \cup (X' \oplus 2) \xrightarrow{aa} \{q_2, q_3\} \cup \{m-2\} \cup X' \xrightarrow{bb} \{q_3\} \cup \{m-2\} \cup X' = \{q_3\} \cup X.$$

(ii) Let there exist an integer k such that $(m-2)-3k$ is in X. Then state $\{q_3\} \cup (X \oplus 3k)$ containing $m-2$ is reachable, as shown in case (i), and goes to state $\{q_3\} \cup X$ by a^{3k}.

(iii) Let there exist a k such that $(m-2)-3k-1$ is in X. Then state $\{q_3\} \cup (X \oplus 4)$ is reachable, as shown in case (ii), and goes to $\{q_3\} \cup X$ by *aabbaabb*, since we have

$$\{q_3\} \cup (X \oplus 4) \xrightarrow{aa} \{q_1\} \cup (X \oplus 2) \xrightarrow{bb} \{q_1, q_2\} \cup (X \oplus 2) \xrightarrow{aa} \{q_2, q_3\} \cup X \xrightarrow{bb} \{q_3\} \cup X. \tag{1}$$

(iv) Let there exist a k such that $(m-2)-3k-2$ is in X. Then $\{q_3\} \cup (X \oplus 4)$ is reachable, as in case (iii), and goes to $\{q_3\} \cup X$ by *aabbaabb*, as the chain of transitions in (1) shows.

(v) State $\{q_3\}$ is reachable since it is a singleton set.

Thus every state $\{q_3\} \cup X$ with $X \subseteq \{0, 1, \ldots, m-1\}$ is reachable. It follows that every state $\{q_1\} \cup X$ as well as every state $\{q_2\} \cup X$ is reachable, since each state $\{q_3\} \cup (X \oplus 1)$ goes to $\{q_2\} \cup X$ by a, and each state $\{q_3\} \cup (X \oplus 2)$ goes to $\{q_1\} \cup X$ by *aa*. Now, every state $\{q_1, q_2\} \cup X$ is reached from state $\{q_1\} \cup X$ by *bb*. States $\{q_1, q_3\} \cup X$ and $\{q_2, q_3\} \cup X$ with $X \subseteq \{0, 1, \ldots, m-1\}$ are reached from states $\{q_1, q_2\} \cup (X \oplus 1)$ and $\{q_1, q_2\} \cup (X \oplus 2)$ by a and *aa*, respectively. Finally, every state $\{q_2\} \cup X$ goes to state $\emptyset \cup X$ by *bb*.

Hence, all the subsets of the state set $\{q_1, q_2, q_3\} \cup \{0, 1, \ldots, m-1\}$ are reachable. By Proposition 3, the subsets are pairwise distinguishable, and our proof is complete. □

4. Binary witnesses for reversal in some other results

The incorrect result from [12] has been used for the reversals of left ideal languages and suffix-closed languages [2,3]. Here we restate these results, and present correct witnesses. Recall that a language L over an alphabet Σ is a left ideal if $L = \Sigma^* L$, and is suffix closed if, for every string w in L, every suffix of w is in L as well. Moreover, a language is suffix closed if and only if its complement is a left ideal: Language L is suffix closed iff $uv \in L$ implies that $v \in L$, which holds if and only if $w \in L^c$ implies $xw \in L^c$ for every x in Σ^*.

Theorem 6 ([2], Theorem 15, Reversal of Left Ideals)**.** *Let L be a left ideal with* $\mathrm{sc}(L) = n$, *where* $n \geqslant 3$. *Then* $\mathrm{sc}(L^R) \leqslant 2^{n-1}+1$, *and the bound is tight in the ternary case.*

Proof. The upper bound is from [2]. For tightness, consider the ternary language accepted by the dfa of Fig. 8, where unspecified transitions on c go to state 1. Notice that the language is a left ideal since, for every string w in L, the strings aw, bw, and cw are in L as well. Consider the subset automaton corresponding to the reverse of this dfa. By Theorem 5, all the subsets of $\{1, 2, \ldots, n-1\}$ are reachable from the initial state $\{n-1\}$. State $\{0, 1, \ldots, n-1\}$ is reached from state $\{1\}$ by c. All the states are pairwise distinguishable by Proposition 3. □

Theorem 7 ([3], Theorem 5, Reversal of Suffix-Closed Languages)**.** *Let L be a suffix-closed regular language with* $\mathrm{sc}(L) = n$, *where* $n \geqslant 3$. *Then* $\mathrm{sc}(L^R) \leqslant 2^{n-1}+1$, *and the bound is tight in the ternary case.*

Proof. The operation of reversal commutes with complementation, and the complement of a suffix-closed language is a left ideal. Thus the upper bound follows. For tightness, consider the complement of the languages accepted by the dfa of Fig. 8. The language is suffix closed, and it meets the upper bound, as shown in the proof above. □

Now, we use the binary witnesses for reversal described in the previous section to strengthen the following result by Mera and Pighizzini [9] showing that there exists a ternary witness language for nfa-to-dfa conversion with small nondeterministic complexity of its complement.

Theorem 8 ([9])**.** *For every positive integer n, there exists a regular language L over the alphabet* $\{a, b, c\}$ *such that*

- $\mathrm{nsc}(L) = n$,
- $\mathrm{nsc}(L^c) \leqslant n+1$,
- $\mathrm{sc}(L) = 2^n$.

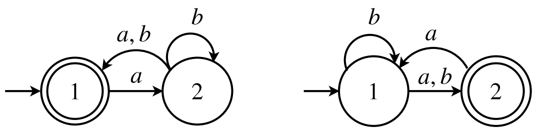

Fig. 9. The 2-state nfas for a language L and L^c, respectively, with $\mathrm{sc}(L) = 4$.

Our first question is if $\mathrm{nsc}(L^c)$ in the above theorem can be decreased to n. Fig. 9 shows that, when $n = 2$, this can indeed happen. The complement of the 2-state nfa language of Fig. 9 (left) is accepted by the 2-state nfa of Fig. 9 (right), while the state complexity of the language (and of its complement) is 4. However, the next theorem shows that this is only an exception.

Theorem 9. *Let $n \neq 2$. There is no regular language L such that*
- $\mathrm{nsc}(L) = n$,
- $\mathrm{nsc}(L^c) = n$,
- $\mathrm{sc}(L) = 2^n$.

Proof. If $n = 1$, then one of the language or its complement must be empty; thus the other one is Σ^*. Both are 1-state dfa languages. Assume that there is a language L with $\mathrm{sc}(L) = 2^n$ and $\mathrm{nsc}(L) = \mathrm{nsc}(L^c) = n$ and $n \geqslant 3$.

Let M be an n-state nfa for the language L with states $q_1, \ldots, q_n$, of which q_1 is the single initial state. Let N be an n-state nfa for the complement L^c with states $p_1, \ldots, p_n$, of which p_1 is a single initial state. Let M' and N' be the 2^n-state dfas obtained from nfas M and N, respectively, by the subset construction. Since both nfas M and N have at least one final state, both dfas M' and N' have at least 2^{n-1} final states. Since the sum of final states in dfas M' and N' is 2^n, it follows that both nfas M and N must have exactly one final state. Moreover, exactly one of the initial states of nfas M and N is accepting. Without loss of generality, state q_1 is accepting, and so state p_1 is rejecting. Since q_1 is the only accepting state in nfa M, we have

$$\text{if } u \in L \text{ and } v \in L, \quad \text{then also } uv \in L. \tag{2}$$

Let p_2 be the final state of nfa N. Since $n \geqslant 3$, and nfa N has exactly one final state, state p_3 is rejecting. Since rejecting state $\{p_1, p_3\}$ is reachable in the subset automaton N', there is a string u such that the initial state $\{p_1\}$ goes by u to the rejecting state $\{p_1, p_3\}$ in dfa N' for language L^c. This means that string u is in language L. Now we are going to show that, in dfa N', the rejecting states $\{p_1, p_3\}$ and $\{p_1\}$ must be equivalent.

If a string v is accepted by dfa N' from state $\{p_1\}$, then it is also accepted from state $\{p_1, p_3\}$. If a string v is rejected by dfa N' from state $\{p_1\}$, then v must be in language L. But then, by (2), we have that $uv \in L$, and so $uv \notin L^c$. This means that v must be rejected by dfa N' from state $\{p_1, p_3\}$; recall that $\{p_1\}$ goes to $\{p_1, p_3\}$ by u. Hence, states $\{p_1\}$ and $\{p_1, p_3\}$ are equivalent, which is a contradiction with $\mathrm{sc}(L) = \mathrm{sc}(L^c) = 2^n$. □

Now, using the fact that the binary witnesses for reversal from Fig. 6 have a single final state, we can prove the following theorem.

Theorem 10. *Let $n \neq 2$. There exists a* binary *regular language L such that*
- $\mathrm{nsc}(L) = n$,
- $\mathrm{nsc}(L^c) = n + 1$,
- $\mathrm{sc}(L) = 2^n$.

Proof. If $n = 1$, then take $L = \{\varepsilon\}$. Let $n \geqslant 3$. Let A be the binary n-state dfa from Theorem 5 meeting the upper bound 2^n on the state complexity of reversal and shown in Fig. 6 on page 5. Set $L = L(A)^R$. Then language L is accepted by n-state nfa A^R that has a single initial state, and Theorem 5 shows that $\mathrm{sc}(L) = \mathrm{sc}(L(A^R)) = 2^n$. Hence $\mathrm{nsc}(L) = n$.

Next, we have $L^c = (L(A)^R)^c = (L(A)^c)^R$. A dfa for language $L(A)^c$ is obtained from dfa A by interchanging the accepting and rejecting states. The reverse of this dfa is an nfa that has multiple initial states, and so we add a new initial state to get an nfa with a single initial state for language $(L(A)^c)^R$. Therefore, taking into account the above theorem, we have $\mathrm{nsc}(L^c) = n + 1$. □

Now, we can ask what the largest value of $\mathrm{sc}(L)$ is if the nondeterministic state complexity of both L and L^c is n. The next lemma shows that the state complexity of such a language may be $2^{n-1} + 1$.

Lemma 11. *Let $n \geqslant 3$. There exists a* binary *regular language L such that* $\mathrm{nsc}(L) = \mathrm{nsc}(L^c) = n$ *and* $\mathrm{sc}(L) = 2^{n-1} + 1$.

Proof. Consider the language L accepted by the n-state nfa of Fig. 10. The nfa is constructed from the reverse of the $(n-1)$-state Šebej binary dfa of Fig. 6 by making state 1 final, and by adding a new initial rejecting state 0 that goes to state 1 by a, and to itself by b. Then the initial state of the corresponding subset automaton is the rejecting state $\{0\}$, that goes to itself on b, and to $\{1\}$ by a. By Theorem 5, all the subsets of $\{1, 2, \ldots, n-1\}$ are reachable and pairwise distinguishable. The rejecting state $\{0\}$ is distinguished from any other rejecting state by bba.

Denote the nfa of Fig. 10 restricted to states $1, 2, \ldots, n-1$ by N, and the language accepted by nfa N by $\hat{L}$. Since the reverse of nfa N is a dfa, we can complement this dfa and then again reverse it; this results in an nfa for the complement of $\hat{L}$ with the same transitions as in nfa N but with initial states $2, 3, \ldots, n-1$.

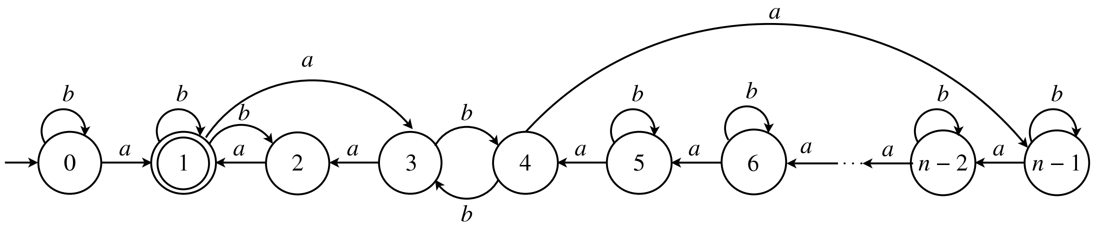

Fig. 10. An n-state nfa for a language L with $\mathrm{sc}(L) = 2^{n-1} + 1$.

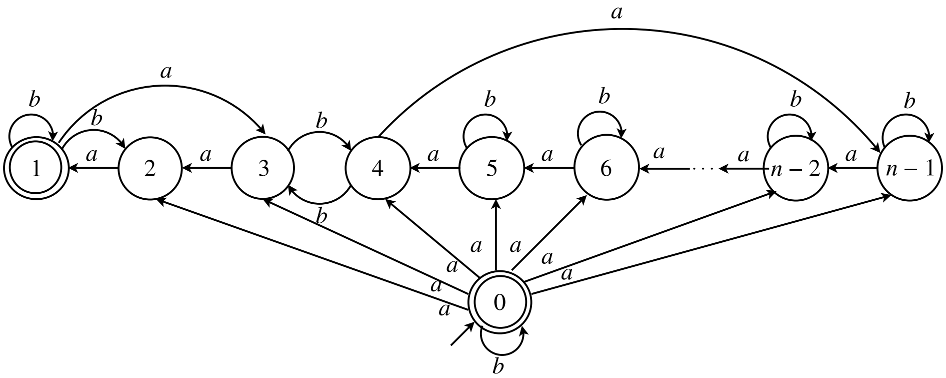

Fig. 11. An n-state nfa for the complement of the language from Fig. 10.

Now consider the n-state nfa shown in Fig. 11. It is constructed in a similar way as the nfa above, but, this time, the new initial state is accepting, and goes to itself by b, and to the set $\{2, 3, \ldots, n-1\}$ by a. Therefore, in the corresponding subset automaton, the initial state $\{0\}$ is accepting, goes to itself on b, and to $\{2, 3, \ldots, n-1\}$ by a, that is, by a it goes to the initial state of the dfa for the complement of the language $\hat{L}$. Since the state complexity of $\hat{L}^c$ is 2^{n-1}, all the subsets of $\{1, 2, \ldots, n-1\}$ must be reachable and pairwise distinguishable in the subset automaton. The accepting initial state $\{0\}$ is distinguished from any other accepting state by ba. Moreover, the subset automaton corresponding to the nfa of Fig. 10 can be obtained from the subset automaton for the nfa of Fig. 11 by interchanging the accepting and rejecting states (and renaming the states).

Since the subset automata have more than 2^{n-1} reachable and pairwise distinguishable states, both nfas are minimal. Thus we have $\mathrm{nsc}(L) = \mathrm{nsc}(L^c) = n$ and $\mathrm{sc}(L) = 2^{n-1} + 1$. Let us remark that the nfa of Fig. 10 accepts the language $b^*a\hat{L}$, while the nfa of Fig. 11 accepts its complement $b^* \cup b^*a\hat{L}^c$. □

The next question is whether or not the state complexity of a language L with $\mathrm{nsc}(L) = \mathrm{nsc}(L^c) = n$ may be larger than $2^{n-1} + 1$. Notice that in such a case at least one of the minimal nfas for the language or its complement must have just one accepting state. If every minimal n-state nfa for a language has more than one accepting state, then, in the corresponding subset automaton, at most 2^{n-2} states are rejecting. This means that, in the minimal dfa for the complement of this language, at most 2^{n-2} are accepting. Thus, if we want to have more than $2^{n-1}+1$ states in the dfa for the complement, at least $2^{n-2}+1$ of them must be rejecting. This can only happen if a minimal nfa for the complement has just one accepting state. The upper bound on the state complexity in this case is thus $2^{n-1} + 2^{n-2}$.

When the minimal nfas for both language and its complement have just one accepting state, the arguments of the proof of Theorem 9 show that states $\{p_1\}$ and $\{p_1\} \cup S$ with $S \subseteq \{3, 4, \ldots, n\}$, and $\{p_1, p_2\}$ and $\{p_1, p_2\} \cup S$ with $S \subseteq \{3, 4, \ldots, n\}$, are equivalent. This gives the upper bound $2^{n-1} + 2$ on the state complexity in such a case. Summarizing, we have the following result.

Theorem 12. *Let L be a regular language such that $\mathrm{nsc}(L) = \mathrm{nsc}(L^c) = n$. Then $\mathrm{sc}(L) \leqslant 2^{n-1} + 2^{n-2}$. When both minimal nondeterministic automata for languages L and L^c have exactly one final state, we have $\mathrm{sc}(L) \leqslant 2^{n-1} + 2$. The lower bound $2^{n-1} + 1$ on the state complexity of such a language is met in the binary case.*

Proof. The upper bounds follow directly from the considerations above. Lemma 11 shows that the binary language accepted by the nfa of Fig. 10 meets the lower bound $2^{n-1} + 1$. □

Whether or not the upper bounds are tight remains open. Quite a lot of attempts to meet at least the bound $2^{n-1} + 2$ were unsuccessful, and even using larger alphabets was of no help.

5. Conclusions

We have examined the state complexity of reversal of regular languages. First, we presented a simple proof that the upper bound 2^n is tight in the ternary case. Then we described binary languages represented by n-state deterministic finite automata with a single final state meeting the upper bound on the state complexity of reversal. Moreover, these automata are one-cycle-free-path automata. Therefore, the binary witness languages are deterministic union-free languages, which improves a result from [5].

Using presented witnesses, we described a binary language such that the nondeterministic state complexity of the language and of its complement is n and $n + 1$, respectively, while the state complexity of the language (and of its complement) is 2^n. This decreases the size of the alphabet used in [9] for describing languages satisfying these requirements.

We also proved that there is no regular language with state complexity 2^n such that both the language and its complement would have nondeterministic state complexity n. If the nondeterministic state complexity of both a language and its complement is n, the upper bound on the state complexity of the language is $2^{n-1} + 2^{n-2}$. Moreover, if both minimal n-state nfas for a language and its complement have just one final state, then the upper bound is $2^{n-1} + 2$. The lower bound $2^{n-1} + 1$ is met by a binary language. Whether or not the upper bounds are tight remains open.

Acknowledgments

We would like to thank Jozef Štefan Jirásek for his help with the computational verification of our conjectures, and Monika Krausová for her help with drawing the figures. We are very grateful to an anonymous referee for his/her careful reading of the manuscript and useful remarks that have helped us to improve the presentation of this paper. The first author's research supported by VEGA grant 2/0183/11.

References

[1] J.A. Brzozowski, Canonical regular expressions and minimal state graphs for definite events, in: J. Fox (Ed.), Proceedings of the Symposium on Mathematical Theory of Automata, New York, NY, April 24-26, 1962, in: MRI Symposia Series, vol. 12, Polytechnic Press of the Polytechnic Institute of Brooklyn, Brooklyn, NY, 1963, pp. 529–561.
[2] J. Brzozowski, G. Jirásková, B. Li, Quotient complexity of ideal languages, in: A. López-Ortiz (Ed.), LATIN 2010, in: LNCS, vol. 6034, Springer, Heidelberg, 2010, pp. 208–211.
[3] J. Brzozowski, G. Jirásková, Ch. Zou, Quotient complexity of closed languages, in: F. Ablayev, E.W. Mayr (Eds.), CSR 2010, in: LNCS, vol. 6072, Springer, Heidelberg, 2010, pp. 84–95.
[4] J.-M. Champarnaud, A. Khorsi, T. Paranthoën, Split and join for minimizing: Brzozowski's algorithm. Available at: http://jmc.feydakins.org/ps/c09psc02.ps.
[5] G. Jirásková, T. Masopust, Complexity in union-free regular languages, in: Y. Gao, H. Lu, S. Seki, S. Yu (Eds.), DLT 2010, in: LNCS, vol. 6224, Springer, Heidelberg, 2010, pp. 255–266.
[6] G. Jirásková, G. Pighizzini, Optimal simulation of self-verifying automata by deterministic automata, Inform. and Comput. 209 (2011) 528–535.
[7] E. Leiss, Succinct representation of regular languages by Boolean automata, Theoret. Comput. Sci. 13 (1981) 323–330.
[8] U.I. Lupanov, A comparison of two types of finite automata, Problemy Kibern. 9 (1963) 321–326 (in Russian).
[9] F. Mera, G. Pighizzini, Complementing unary nondeterministic automata, Theoret. Comput. Sci. 330 (2005) 349–360.
[10] B.G. Mirkin, On dual automata, Kibern. 2 (1966) 7–10 (in Russian); Cybernet. 2 (1966) 6–9 (English translation).
[11] M. Rabin, D. Scott, Finite automata and their decision problems, IBM Res. Develop. 3 (1959) 114–129.
[12] A. Salomaa, D. Wood, S. Yu, On the state complexity of reversals of regular languages, Theoret. Comput. Sci. 320 (2004) 315–329.
[13] M. Sipser, Introduction to the Theory of Computation, PWS Publishing Company, Boston, 1997.
[14] J. Šebej, Reversal of regular languages and state complexity, in: D. Pardubská (Ed.), Proc. 10th ITAT, Šafárik University, Košice, 2010, pp. 47–54.
[15] S. Yu, Chapter 2: Regular languages, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages - Vol. I, Springer, Heidelberg, 1997, pp. 41–110.
[16] S. Yu, Q. Zhuang, K. Salomaa, The state complexity of some basic operations on regular languages, Theoret. Comput. Sci. 125 (1994) 315–328.