Reversal on Regular Languages and Descriptional Complexity

Juraj Šebej*

Institute of Computer Science, Faculty of Science, P.J. Šafárik University, Jesenná 5, 040 01 Košice, Slovakia
juraj.sebej@gmail.com

Abstract. We study the problem stated as follows: which values in the range from $\log n$ to 2^n may be obtained as the state complexity of the reverse of a regular language represented by a minimal deterministic automaton of n states? In the main result of this paper we use an alphabet of size $2n - 2$ to show that the entire range of complexities may be produced for any n. This considerably improves an analogous result from the literature that uses an alphabet of size 2^n. We also provide some partial results for the case of a binary alphabet.

1 Introduction

Reversal is an operation on formal languages defined by $L^R = \{w^R \mid w \in L\}$, where w^R is the mirror image of w, that is, the string w written backwards. The reverse of a regular language is again a regular language [12]. A nondeterministic finite automaton for the reverse of a regular language can be constructed from an automaton recognizing the given language by reversing all the transitions and swapping the role of initial and final states. This gives the upper bound 2^n on the number of states in the state complexity of reversal.

Mirkin [11] pointed out that Lupanov's ternary witness automaton [10] for determinization of nondeterministic automata proves the tightness of the upper bound 2^n for reversal in the case of a three-letter alphabet since the ternary nondeterministic automaton is the reverse of a deterministic automaton. Another ternary worst-case example for reversal was given in 1981 by Leiss [9], who also proved the tightness of the upper bound in the binary case. However, his binary automata have $n/2$ final states. In [8] we presented binary witness automata with a single final state. Moreover, the witness automata from [8] are so-called one-cycle-free-path automata which improved a result in [7].

In this paper we are interested not only in the worst-case complexity, but rather with all possible values that can be achieved as the state complexity of the reverse of a regular language represented by an n-state deterministic automaton.

Our motivation comes from the paper by Iwama, Kambayashi and Takaki [3], in which the authors stated the problem of whether there always exists a regular

* The author was supported by the Slovak Grant Agency for Science under contract VEGA 1/0479/12.

H. Jürgensen and R. Reis (Eds.): DCFS 2013, LNCS 8031, pp. 265–276, 2013.

language represented by a minimal n-state nondeterministic finite automaton such that the minimal deterministic automaton for the language has α states for any integers n and α with $n \leq \alpha \leq 2^n$. The values that cannot be obtained in such a way are called "magic" in [4]. The problem was solved positively in [6] by using a ternary alphabet. On the other hand, "magic" numbers exist in the case of a unary alphabet. The binary case is still open.

In the case of the operation of reversal, the possible complexities are in the range from $\log n$ to 2^n. Using an alphabet of size 2^n, Jiráskova [5] has shown that there are no gaps in the hierarchy of complexities for reversal for any n. Here we improve this result using an alphabet of size $2n - 2$. We prove that each number in the range from $\log n$ to 2^n can be obtained as the number of states in the minimal deterministic automaton for the reverse of a regular language represented by a minimal deterministic automaton of n states over an alphabet of size $2n-2$. Decreasing the input alphabet to a fixed size seems to be a challenging problem since nondeterministic automata obtained as the reverse of deterministic automata have some special properties, and so the constructions for NFA-to-DFA conversion [6] cannot be used.

In the second part of the paper, we consider the binary case. We get a continuous segment of a quadratic length of achievable complexities for $n \geq 8$. Using our Java program we did some computations. These computations show that each value from $\log n$ to 2^n may be a state complexity of a binary regular language represented by an n-state DFA, where $2 \leq n \leq 8$.

2 Preliminaries

We assume that the reader is familiar with the basic notions of automata theory, and for all unexplained notions we refer to [13,14].

All the *deterministic finite automata* (DFAs) in this paper are assumed to be complete, and our *nondeterministic finite automata* (NFAs) have multiple initial states and no ε-transitions. The *state complexity* of a regular language L, denoted by $\mathrm{sc}(L)$, is the number of states in the minimal DFA for L.

Every NFA $M = (Q, \Sigma, \delta, Q_0, F)$ can be converted to an equivalent DFA $M' = (2^Q, \Sigma, \delta', Q_0, F')$, where $\delta'(R, a) = \delta(R, a)$ for each subset R of Q and each a in Σ, and $F' = \{R \in 2^Q \mid R \cap F \neq \emptyset\}$ [12]. We call the DFA M' the *subset automaton* of the NFA M. The subset automaton M' need not be minimal since some of its states may be unreachable or equivalent.

The reverse w^R of a string w is defined as follows: $\varepsilon^R = \varepsilon$ and if $w = a_1 a_2 \cdots a_n$ with $a_i \in \Sigma$, then $w^R = a_n \cdots a_2 a_1$. 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 the DFA A by reversing all the transitions and by swapping the role of initial and final states, that is $A^R = \big(Q, \Sigma, \delta^R, F, \{s\}\big)$, where $\delta^R(q, a) = \{p \in Q : \ \delta(p, a) = q\}$. Let us recall the quite interesting result that no two distinct states in the subset automaton corresponding to the reverse of a minimal DFA are equivalent. This means that, throughout the paper, we need not prove distinguishability of states of the subset automaton.

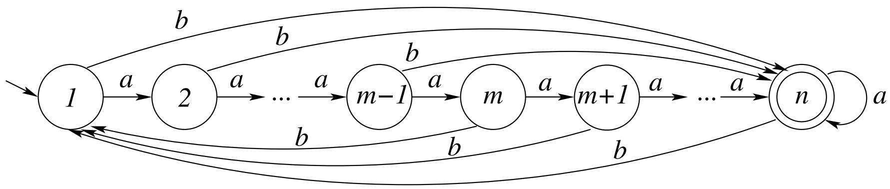

Fig. 1. The deterministic finite automaton for L; $\alpha = n + m, 1 \leqslant m \leqslant n$

Proposition 1 ([1,8,11]). *All the states of the subset automaton corresponding to the reverse of a minimal DFA are pairwise distinguishable.*

The following lemma from [5] shows that each number from n to $2n$ may be the state complexity of the reverse of a binary language represented by a minimal n-state DFA. We will use the lemma several times later in the paper.

Lemma 1 ([5]). *For all integers n and α with $2 \leqslant n \leqslant \alpha \leqslant 2n$, there exists a binary regular language L such that* $\mathrm{sc}(L) = n$ *and* $\mathrm{sc}(L^R) = \alpha$*.*

Proof (Sketch). For $\alpha = n + m$ $(0 \leq m \leq n)$, the DFA for L is shown in Fig. 1.

3 Linear Alphabet

It is known that there are no gaps in the hierarchy of complexities for reversal in the case of an alphabet of size 2^n [5]. The aim of this section is to show that a linear alphabet of size $2n - 2$ is enough to obtain each state complexity of reversal in the range from $\log n$ to 2^n.

We start with two examples. The first one shows how we can double the number of reachable states, respectively double and add one more state, in the subset automaton for reverse by adding one new state and two new letters. This illustrates our proof by mathematical induction given in this section.

The second example shows that we also are able to provide an explicit construction of an appropriate automaton for a given number of states in the original automaton and a given value of the state complexity of reversal.

Example 1. Consider the 3-state DFA B in Fig. 2 (top left) with the sole final state $f = 3$. Its reverse B^R is shown in Fig. 2 (bottom left), and the minimal DFA for the reverse has 5 states. Let us show how can we construct a 4-state DFA requiring $2 \cdot 5$ deterministic states for reverse, and a 4-state DFA requiring $2 \cdot 5 + 1$ deterministic states for reverse.

To get a 4-state DFA A whose reverse requires $2 \cdot 5$ deterministic states, add a new rejecting state N going to itself on a, b, and transitions on two new letters a_4, b_4 defined as follows: by a_4, state N goes to state f, and every other state of A goes to itself, and by b_4, every state of A goes to state N. The resulting 4-state DFA A is again minimal. Fig. 2 (top right) shows the reverse A^R of A.

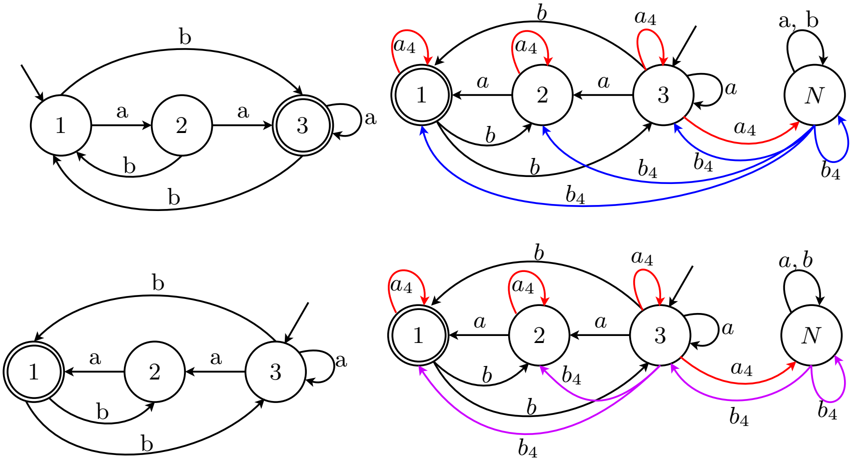

Fig. 2. The construction of 5-state DFAs requiring $2 \cdot 10$ and $2 \cdot 10 + 1$ states for reverse

In the subset automaton A' corresponding to NFA A^R, all the states that were reachable in the subset automaton B' from state $\{f\} = \{3\}$ will be reachable since $\{f\}$ is also the initial state of A' and we did not change transitions on a, b in states 1, 2, 3. Moreover, state $\{3\}$ goes to state $\{1, 3\}$ on a_4, and then all the states $X \cup \{N\}$, where X is reachable in B', will be reachable. No other set will be reachable in the subset automaton A', so A' has 10 states.

To get a 4-state DFA A requiring $2 \cdot 5 + 1 = 11$ states for the reverse, we again add a new rejecting state N going to itself on a, b. We also add transitions on a_4 as above. Next we use the following conditions that are satisfied for B and B':

(i) Automaton B with the set of states Q_B has exactly one final state.
(ii) There exists a set $S_B = \{1, 2\}$ of states of B which is not reachable in B'. The set S_B does not contain the final state of B.
(iii) The set $S_B^c = \{3\}$, which is the complement of S_B in B, is reachable in B'.
(iv) S_B goes by each symbol either to itself, or to a set that is reachable in B'.
(v) States $\emptyset$ and Q_B are reachable in B'.

Now we add transitions on symbol b_4 defined as follows: by b_4, each state in the set S_B goes to state f, and every other state of A goes to state N. Fig. 2 (bottom right) shows the reverse A^R of the DFA A. The 10 subsets are reachable in A' as above, and moreover, the set S_B is reachable from $\{f\}$ by b_4. However, no other set is reachable, and so 11 states are reachable. □

Using the above described procedure, we will be able to construct n-state DFAs requiring 2α and $2\alpha + 1$ states from an $(n-1)$-state DFA requiring α states. Assuming that we can reach every value from n to $2^{n-1} - 1$ by $(n-1)$-state DFAs, then we will be able to reach all the value from $2n$ to $2^n - 1$ by n-state DFAs.

Although the proof by induction will be an existential proof, the next example shows that given n and α we, in fact, can provide the construction of an n-state DFA requiring α states for the reverse.

Example 2. Let $n = 8$ and $\alpha = 185$. We want to construct an 8-state DFA, the reverse of which after determinisation has 185 states. We start to divide the current value of α, or $\alpha - 1$ if α is odd, by two, and decrease the value of n by one, until the result is smaller then the new value of n multiplied by two:

n	α						
8	185	$2 \cdot 8 > 185$	*no*	$(185-1)/2 = 92$	N_8	a_8, b_8	S_8
7	92	$2 \cdot 7 > 92$	*no*	$92/2 = 46$	N_7	a_7, b_7	S_7
6	46	$2 \cdot 6 > 46$	*no*	$46/2 = 23$	N_6	a_6, b_6	S_6
5	23	$2 \cdot 5 > 23$	*no*	$(23-1)/2 = 11$	N_5	a_5, b_5	S_5
4	11	$2 \cdot 4 > 11$	*no*	$(11-1)/2 = 5$	N_4	a_4, b_4	S_4
3	5	$2 \cdot 3 > 5$	*yes*	*initiate*			S_3

We cannot construct this automaton directly using Lemma 1 because $185 > 2 \cdot 8$. We have to start from the 7-state automaton whose reverse requires $(185-1)/2 = 92$ deterministic states. Since $92 > 2 \cdot 7$, we repeat the previous case but now the number 92 is even. We have to start from 6-state automaton whose reverse requires $92/2 = 46$ states. As $46 > 2 \cdot 6$, we repeat the previous case. We have to start from 5-state DFA whose reverse requires $46/2 = 23$ deterministic states. Again $23 > 2 \cdot 5$, and we have to start from 4-state automaton requiring $(23-1)/2 = 11$ deterministic states for reverse. Still $11 > 2 \cdot 4$, we have to start from 3-state DFA requiring $(11-1)/2 = 5$ deterministic states for reverse. Now we have $5 < 2 \cdot 3$, and finally we use the initial DFA given by Lemma 1 which is the same as the DFA in Example 1 shown in Fig. 2 (top left).

Now we construct our automaton backwards through the calculations. We add states N_i for $i = 4, \ldots, 8$ step by step and in each step we also add symbols a_i, b_i. In case $i \in \{6, 7\}$ we use the construction from the first part of Example 1, for $i \in \{4, 5, 8\}$ the construction follows the second part of example. For simplicity, we discuss only the changes of the states $S_i, i = 3, \ldots, 8$ and define all a_i, b_i at once. By a_i all states go to itself except for 3 which goes by a_i to N_i. If $i \in \{6, 7\}$, then all the states in $\{1, 2, 3, N_1, \ldots, N_i\}$ go by b_i to N_i and all the other states to itself. If $i \in \{4, 5, 8\}$, then the states from S_i go by b_i to state 3 which is final, the states from $\{1, 2, 3, N_1, \ldots, N_i\} \setminus S_i$ go to state N_i, and states $N_{i+1}, \ldots, N_8$ go to itself. As we showed in Example 1 when we use the first type of the construction, we do not change the set S_i, otherwise we add N_i to it: $S_3 = \{1, 2\}$ $S_4 = \{1, 2, N_4\}$ $S_5 = S_6 = S_7 = \{1, 2, N_4, N_5\}$ $S_8 = \{1, 2, N_4, N_5, N_8\}$. The reverse of the resulting automaton is shown in Fig. 3. □

Now we use the principles of the above examples to show that we can reach each complexity from $n = n + 1$ to $2^n - 1$ for reversal in the case of a linear alphabet.

Lemma 2. *For every* n, α *with* $3 \le n+1 \le \alpha \le 2^n - 1$*, there exists a language* L *over an alphabet* Σ*,* $|\Sigma| \le 2n - 2$*, such that* $\mathrm{sc}(L) = n$ *and* $\mathrm{sc}(L^R) = \alpha$*.*

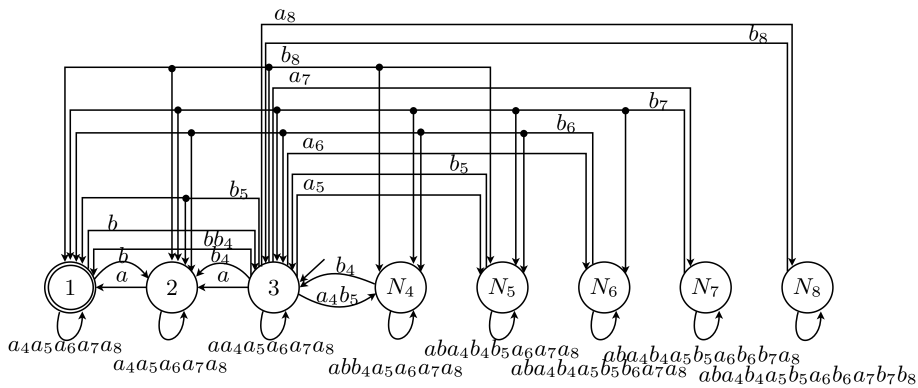

Fig. 3. The reverse of an 8-state automaton which has 185 reachable states after determinisation

Proof. For a DFA A, we denote by A' the subset automaton of the reverse of A.

The proof is by induction on the number of states n of the minimal DFA for L. We are going to show that for every α with $n+1 \leq \alpha \leq 2^n - 1$ there exists an n-state minimal DFA A over an alphabet Σ with $|\Sigma| \leq 2n-2$ such that the minimal DFA for $L(A)^R$ has α states, and moreover, the following five conditions for automata A, A' are satisfied:

(i) The DFA A with the state set Q_A has exactly one final state.
(ii) There exists a set S_A of states of A which is not reachable in the subset automaton A'. The set S_A does not contain the final state of A.
(iii) The set S_A^c, which is the complement of S_A in A, is reachable in A'.
(iv) S_A goes by each symbol either to itself, or to a set that is reachable in A'.
(v) The states $\emptyset$ and Q_A are reachable in A'.

The base case is $n=2$ and $\alpha=3$. Consider the binary DFA A from Lemma 1 for $n=2$ and $\alpha=3$. The DFA A satisfies the conditions (i)-(v) with $S_A=\{1\}$.

Let $n>2$, and assume that the theorem holds for $n-1$, that is, for every β with $n \leq \beta \leq 2^{n-1}-1$, there exists an $(n-1)$-state automaton B over an alphabet Σ_B, $|\Sigma_B| \leq 2(n-1)-2$, such that the minimal DFA for $L(B)^R$ has β states, and moreover, the following five conditions are satisfied:

(i) Automaton B with the state set Q_B has exactly one final state.
(ii) There exists a set S_B of states of B which is not reachable in B'. The set S_B does not contain the final state of B.
(iii) The set S_B^c, which is the complement of set S_B in B, is reachable in B'.
(iv) S_B goes by each symbol either to itself, or to a set that is reachable in B'.
(v) States $\emptyset$ and Q_B are reachable in DFA B'.

Now we prove that for every α with $n+1 \leq \alpha \leq 2^n-1$, there exists an n-state DFA A such that the minimal DFA for language $L(A)^R$ has α states, and moreover, the five conditions above are satisfied for automata A, A'.

We consider three cases depending on the value of α: (1) $n+1 \le \alpha \le 2n-1$; (2) $2n \le \alpha \le 2^n - 1$ and α is even; (3) $2n \le \alpha \le 2^n - 1$ and α is odd.

(1) Let $n+1 \le \alpha \le 2n-1$. Similarly as in the base case we use the automaton from Lemma 1; notice that this is possible for our values of n and α. The DFA A satisfies the conditions (i)-(v) with $S_A = \{1, 2, \dots, m\}$.

(2) Let $2n \le \alpha \le 2^n - 1$ and α is an even number. Now we use the $(n-1)$-state automaton B over an alphabet Σ_B with the state set Q_B, and the final state f for $\beta = \alpha/2$ from the induction hypothesis. We construct the n-state DFA A from DFA B by adding a new non-final state N, and transitions on two new letters a_n, b_n. We have to define the transitions on new letters in all states of A, and the transitions on all letters in state N to make A deterministic. Let us define transitions on a_n as follows: state N goes by a_n to the final state f, and every other state of A goes to itself on a_n. By b_n, each state of A goes to state N. State N goes by each old letter in Σ_B to itself.

Since the DFA B is minimal, the states of B are reachable and pairwise distinguishable in the DFA A as well because we did not change the old transitions and the finality of old states. The state N is reached from the state f on b_n. We need to show that N is not equivalent to any other state of B. The final state f and the state N are not equivalent since N is not final. The state N is not equivalent to any other state of B since a_n is accepted in A only from N and f.

Now we prove that the subset automaton A' has $\alpha = 2\beta$ states. All the states that are reachable in the subset automaton B' are also reachable in the subset automaton A' since the initial state of A' is the same as the initial state of B', namely $\{f\}$, and we did not change the old transitions. Moreover, the state $\{f\}$ goes to the state $\{f, N\}$ by a_n, from which each state $X \cup \{N\}$, where X is reachable in B', can be reached by old letters; recall that the state N goes to itself on each old letter. To show that no other state is reachable in A' notice that every set X that is reachable in B' goes by a_n to itself if $f \notin X$, and to $X \cup \{N\}$ otherwise, and by b_n to the empty set that is reachable in B' by induction. Next, each state $X \cup \{N\}$ goes by a_n to itself if $f \in X \cup \{N\}$, and to X otherwise, by b_n to $Q_B \cup \{N\}$, and by each old letter in Σ_B it goes to a set $X' \cup \{N\}$ where X' is reachable in B'. This means that A' has exactly 2β reachable states and, by Lemma 1, pairwise distinguishable states. Finally, we need to verify the conditions (i)-(v) for automata A, A'.

(i) Automaton A has one final state because we defined N as a non-final state.
(ii) Let $S_A = S_B$. Then S_A is not reachable in A', and it does not contain the final state of A.
(iii) Since S_B^c was reachable in B' on some string w, it follows that the set $S_A^c = S_B^c \cup \{N\}$ is reachable in A' by $a_n w$.
(iv) If a is a letter in Σ_B, then S_A goes either to itself or to a set that is reachable in B' by the induction hypothesis. By a_n, the set S_A goes to itself since $f \notin S_A$. By b_n, the set S_A goes to the empty set, which is reachable in B', thus in A', by the induction hypothesis.
(v) The empty set is reachable in B' and therefore in A'. Since Q_B is reachable in B' by a string w, the set $Q_A = Q_B \cup \{N\}$ is reachable in A' by $a_n w$.

(3) Let $2n \le \alpha \le 2^n - 1$ and α is odd. This part of the proof is similar to part (2) with the following changes. By b_n, each state of the set S_B goes to state f, and every other state of A goes to state N. It follows that now also the state S_B is reachable in A', and so A' has exactly $2\beta + 1$ reachable states. The new set S_A is equal to $S_B \cup \{N\}$. The proof of the theorem is now complete. □

Using the results of Lemma 1, Lemma 2, and the results from [8,9] we can prove the main result of this paper which shows that there are no gaps in the hierarchy of state complexities for reversal in the case of a linear alphabet.

Theorem 1. *For every* n, α *with* $n \ge 3$ *and* $\log n \le \alpha \le 2^n$*, there exists a language* L *over an alphabet* Σ*,* $|\Sigma| \le 2n-2$*, such that* $\mathrm{sc}(L) = n$ *and* $\mathrm{sc}(L^R) = \alpha$*.*

Proof. The case of $n + 1 \le \alpha \le 2^n - 1$ is covered by Lemma 2. For $\alpha = n$, we can use Lemma 1, and for $\alpha = 2^n$ we can use the results from [8,9]. When we reverse the languages from Lemma 2 and the two above mentioned languages, we obtain all the possible state complexities between $\log n \le \alpha \le n$. □

4 Binary Alphabet

In this section we consider the reversal of binary regular languages. Lemma 1 shows that each complexity from n to $2n$ is achievable for reversal in the binary case. The upper bound 2^n also can be met by a binary language [9, Proposition 2], [8, Theorem 5].

The aim of this section is to find a non-linear number of achievable complexities for reversal in the binary case. We show that each value from $\sqrt{8n}$ to $n^2/8$ can be obtained as the state complexity of the reverse of a binary language represented by an n-state deterministic finite automaton, where $n \ge 8$.

By using our Java program we show that all complexities are achievable in the binary case if $n \le 8$, except for $n = 1$ where 2 cannot be achieved, and $n = 2$ where 1 cannot be achieved. Moreover we compute the frequency of the state complexities of the reverses for $n = 2, 3, 4, 5$. The results of our computations are given in the graphs in the end of this section Fig. 6.

Now we are going to define special automata that we will use later to get a quadratic number of values that can be obtained as the state complexity of the reverse of a binary n-state regular language.

To this aim let $2 \le p < m \le n - 2$ and let $A = (\{1, 2, \dots, n\}, \{a, b\}, \delta, 1, \{n\})$ be a DFA, in which the transitions are as follows. Each state i goes by a to state $i+1$, except for n which goes to itself. By b, each state i with $i \le p-1$ goes to state $i + 1$, each state i with $p \le i \le m - 1$ goes to itself, state m goes to n, and each state i with $i \ge m+1$ goes to $m+1$. Since two distinct states can be distinguished by a string in a^*, the automaton A is minimal. Fig. 4 shows the reverse of A, and the next lemma deals with the complexity of the reverse of $L(A)$.

Lemma 3. *Let* $n \ge 5$ *and* $2 \le p < m \le n-2$ *and let* L *be the language accepted by the DFA in Fig. 4. Then* $\mathrm{sc}(L^R) = n + m + 1 + p\,(p - 1)/2$*.*

Proof. Let $n \geq 5$ and $2 \leq p < m \leq n-2$.

Consider the NFA A^R for $L(A^R)$ shown in Fig. 4. We will show that the minimal DFA for $L(A^R)$ has $n+m+1+p\,(p-1)/2$ states. To prove this, by Lemma 1, it is enough to show that the subset automaton corresponding to the NFA A^R shown in Fig. 5 has exactly $n+m+1+p\,(p-1)/2$ reachable states.

Denote for $i = 1, 2, \ldots, n$,

$$S_i = \{n, n-1, ..., i+1, i\},$$

and for $i = 1, 2, \ldots, p-1$

$$T_i = \{p, p-1, \ldots, p-i\}.$$

For an integer j with $0 \leq j \leq p-i-1$, denote

$$T_i \ominus j = \{p-j, p-1-j, \ldots, p-i-j\}.$$

Let
$\mathcal{R}_1 = \{S_i \mid 1 \leq i \leq n\}$,
$\mathcal{R}_2 = \{\{i\} \mid 1 \leq i \leq m\} \cup \{\emptyset\}$,
$\mathcal{R}_3 = \{T_i \ominus j \mid 1 \leq i \leq p-1 \text{ and } 0 \leq j \leq p-i-1\}$,
$\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$.

The family $\mathcal{R}$ consists of $n+m+1+p(p-1)/2$ sets, and we will prove that (1) each set in $\mathcal{R}$ is reachable in the subset automaton and (2) no other set is reachable. Every S_i in R_1 is reachable from the initial state $\{n\}$ by a^{i-1}. Every $\{i\}$ in R_2 is reachable from $\{n\}$ by ba^{m-i}, and $\emptyset$ is reachable from $\{1\}$ by a. Every $T_i \ominus j$ in R_3 is reachable from $\{n\}$ by $ba^{m-p}b^i a^j$. This proves (1).

Since the initial state $\{n\}$ is in $\mathcal{R}$, to show (2) it is enough to prove that each set in $\mathcal{R}$ goes to a set in $\mathcal{R}$ by both a and b. By a, each S_i in R_1 goes to S_{i-1}, except for S_1 which goes to itself, each i in R_2 goes to $i-1$, except for 1 which goes to the empty set, each $T_i \ominus j$ in R_3 goes to $T_i \ominus (j+1)$, except for $T_i \ominus (i-1)$ which goes to $T_{i-1} \ominus (i-2)$ and $T_1 \ominus (p-2)$ which goes to $\{1\}$. By b, each S_i goes to m if $i \geq m+2$, to S_m if $i = m+1$, to itself if $m \leq i \leq p+1$, and if $i \leq p$ then S_i goes to the same state as it goes on a. Next, the set $\{i\}$ goes to the

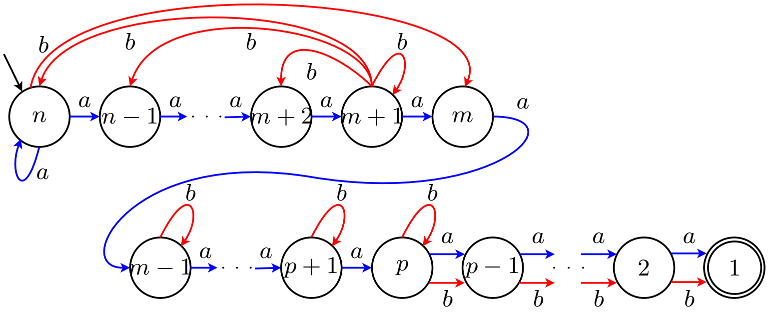

Fig. 4. Automaton which equivalent DFA has exactly $n+m+1+p.(p-1)/2$ states

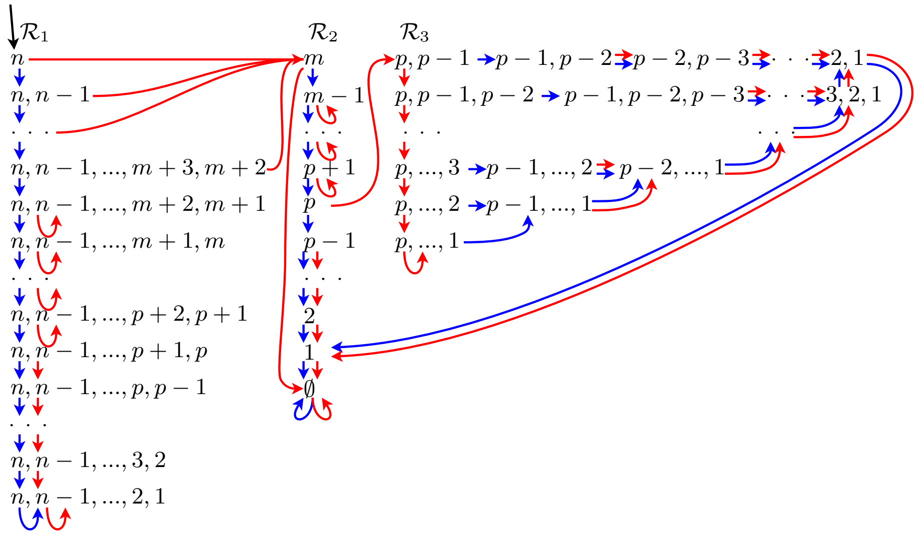

Fig. 5. Sketch of subset construction of the automaton from Fig. 4

empty set of states by b when i equals m or 1, to itself if $m-1 \geq i \geq p+1$, to $T_1 \ominus 0$ when $i = p$, otherwise it goes to $\{i-1\}$. The set T_i goes to T_{i+1} whenever $i \neq p-1$, to $T_{p-2} \ominus 1$ when $i = p-1$, otherwise it goes to the same state as it goes on a. The empty set goes to itself by both a and b. Since all the resulting sets are in $\mathcal{R}$, the proof of (2) is complete.

Hence, the subset automaton has exactly $n + m + 1 + p\,(p-1)/2$ reachable and, by Lemma 1, pairwise distinguishable states. This proves the theorem. $\square$

The next lemma shows that each value from $n+5$ to $(n^2+10n-8)/8$ can be obtained as the state complexity of the reverse of an n-state binary language.

Lemma 4. *For every n and α with $n \geq 5$ and $n+5 \leq \alpha \leq (n^2+10n-8)/8$, there exists a binary regular language L such that $\mathrm{sc}(L) = n$ and $\mathrm{sc}(L^R) = \alpha$.*

Proof. Let $n+5 \leq \alpha \leq (n^2+10n-8)/8$. Then

$$n+5 \leq \alpha \leq n+2+(1+2+\cdots+\lfloor (n-3)/2 \rfloor + 1) + \lfloor (n-3)/2 \rfloor - 1.$$

This means that there exists an integer p such that $2 \leq p \leq \lfloor (n-3)/2 \rfloor + 1$ and

$$(n+2)+(1+2+\cdots+p) \leq \alpha < (n+2)+(1+2+\cdots+p+p+1).$$

Then

$$\alpha = (n+2)+(1+2+\cdots+p)+i$$

for some integer i such that $0 \leq i \leq p$, respectively if $p = \lfloor (n-3)/2 \rfloor + 1$ then $0 \leq i \leq p-2$.

Set $m = p+i+1$. Then $p < m \leq n-2$. Let A be the DFA A from Lemma 3 defined for integers $n, m = p+i+1, p$. By Lemma 3, the minimal DFA for $L(A)^R$ has $n+(p+i+1)+1+p(p-1)/2 = (n+2)+(1+2+\cdots+p)+i = \alpha$ states. $\square$

Now we are able to get a continuous segment of a quadratic length of state complexities of reversal in the binary case.

Theorem 2. *For every n and α with $n \geq 8$ and $\sqrt{8n} \leq \alpha \leq n^2/8$, there exists a binary regular language L such that $\mathrm{sc}(L) = n$ and $\mathrm{sc}(L^R) = \alpha$.*

Proof. Let $n \geq 8$. If $n \leq \alpha < n+5 \leq 2n$, then the language is given by Lemma 1. The case of $n+5 \leq \alpha \leq n^2/8$ is covered by Lemma 4. Since $(L^R)^R = L$, when we reverse the languages mentioned in the two lemmas, we obtain all the possible state complexities of reversal between $\sqrt{8n} \leq \alpha \leq n$. □

Notice that using automata from Lemma 3 we are able to get additional

$$1 + 2 + \cdots + (n-2) - (\lfloor (n-3)/2 \rfloor + 2) \geq n^2/9$$

complexities outside the continuous segment in the previous lemma.

In the last part of this section we discuss the small values of n. For $n = 2, 3, 4, 5$ we used the lists of pairwise non-isomorphic DFAs, and compute the state complexities of their reverses. The graphs in Fig. 6 show the number of automata with the corresponding complexities of reversal. It follows from the graphs that all the values of α from $\log n$ to 2^n can be reach for $n = 2, 3, 4, 5$ with the exception of $n = 2$ and $\alpha = 1$.

For $n = 6, 7, 8$ we changed the strategy of searching of appropriate automata. The first strategy was to define a so that state i goes to $i + 1$ and the last state

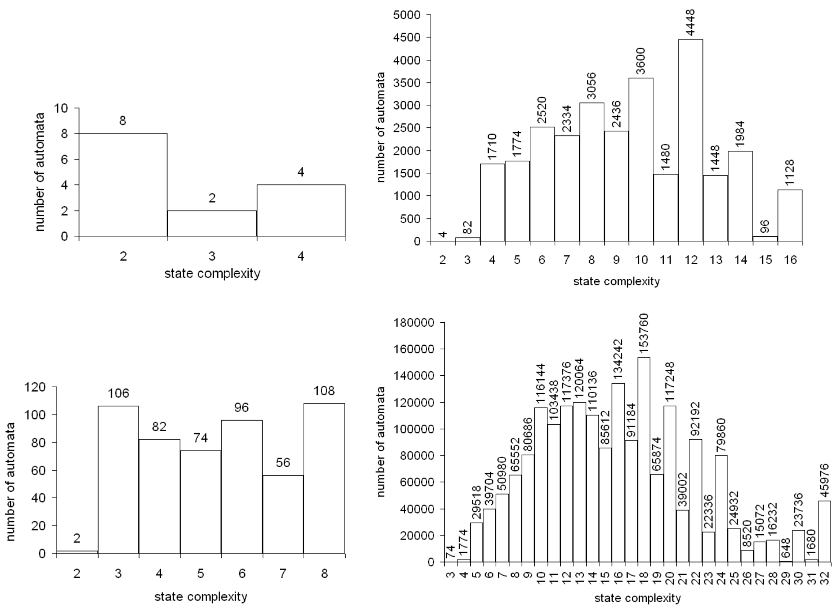

Fig. 6. The frequencies of state complexities for reversal: $n = 2$ top left, $n = 3$ bottom left, $n = 4$ top right, $n = 5$ bottom right

goes to state 0 because we do not have to control minimality in such a case. The other strategy was to generate all the transitions randomly but we used it only for the upper part of the range because here the minimality is guaranteed. We obtained all the complexities of reversal in the range from $\log n$ to 2^n for $n = 6, 7, 8$.

References

1. Brzozowski, J.A.: Canonical regular expressions and minimal state graphs for definite events. In: Fox, J. (ed.) Proceedings of the Symposium on Mathematical Theory of Automata, New York, NY, April 24-26. MRI Symposia Series, vol. 12, pp. 529–561. Polytechnic Press of the Polytechnic Institute of Brooklyn, Brooklyn (1963)
2. Geffert, V.: Magic numbers in the state hierarchy of finite automata. Inform. Comput. 205, 1652–1670 (2007)
3. Iwama, K., Kambayashi, Y., Takaki, K.: Tight bounds on the number of states of DFAs that are equivalent to n-state NFAs. Theoret. Comput. Sci. 237, 485–494 (2000)
4. Iwama, K., Matsuura, A., Paterson, M.: A family of NFAs which need $2^n - \alpha$ deterministic states. Theoret. Comput. Sci. 301, 451–462 (2003)
5. Jirásková, G.: On the state complexity of complements, stars, and reversals of regular languages. In: Ito, M., Toyama, M. (eds.) DLT 2008. LNCS, vol. 5257, pp. 431–442. Springer, Heidelberg (2008)
6. Jirásková, G.: Magic numbers and ternary alphabet. Inter. J. Found. Comput. Sci. 22, 331–344 (2011)
7. Jirásková, G., Masopust, T.: Complexity in union-free regular languages. Internat. J. Found. Comput. Sci. 22, 1639–1653 (2011)
8. Jirásková, G., Šebej, J.: Reversal of binary regular languages. Theoret. Comput. Sci. 449, 85–92 (2012)
9. Leiss, E.: Succinct representation of regular languages by Boolean automata. Theoret. Comput. Sci. 13, 323–330 (1981)
10. Lupanov, O.B.: A comparison of two types of finite automata. Problemy Kibernetiki 9, 321–326 (1963) (in Russian)
11. Mirkin, B.G.: On dual automata. Kibernetika (Kiev) 2, 7–10 (1966) (in Russian); English translation: Cybernetics 2, 6–9 (1966)
12. Rabin, M., Scott, D.: Finite automata and their decision problems. IBM Res. Develop. 3, 114–129 (1959)
13. Sipser, M.: Introduction to the theory of computation. PWS Publishing Company, Boston (1997)
14. Yu, S.: Regular languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. I, ch. 2, pp. 41–110. Springer, Heidelberg (1997)