# On the State and Computational Complexity of the Reverse of Acyclic Minimal DFAs

Galina Jirásková [1,⋆] and Tomáš Masopust [2,⋆⋆]

[1] Mathematical Institute, Slovak Academy of Sciences
Grešákova 6, 040 01 Košice, Slovak Republic
jiraskov@saske.sk
[2] Institute of Mathematics, Academy of Sciences of the Czech Republic
Žižkova 22, 616 62 Brno, Czech Republic
masopust@math.cas.cz

**Abstract.** We study the state complexity of the reverse of acyclic minimal deterministic finite automata, and the computational complexity of the following problem: Given an acyclic minimal DFA, is the minimal DFA for the reverse also acyclic? Note that we allow self-loops in acyclic automata. We show that there exists a language accepted by an acyclic minimal DFA such that the minimal DFA for its reverse is exponential with respect to the number of states, and we establish a tight bound on the state complexity of the reverse of acyclic DFAs. We also give a direct proof of the fact that the minimal DFA for the reverse is acyclic if and only if the original acyclic minimal DFA satisfies a certain structural property, which can be tested in quadratic time.

## 1 Introduction

The reverse of a machine or of a language is one of the classical operations in automata and formal language theory. However, in comparison with other operations, such as the boolean operations, the descriptional complexity of the reverse of regular languages is exponential in the worst case with respect to the number of states of minimal deterministic finite automata (DFAs). This paper demonstrates that this also holds true for a subclass of regular languages accepted by acyclic minimal DFAs. To prevent confusion with DFAs accepting only finite languages, it is important to explain here that we allow self-loops in acyclic automata. Thus, the notion of *acyclic* stands for automata without cycles of length two or more. This definition is adapted from the literature [7, 15, 16, 18].

The first part of this paper studies the state complexity of the reverse of acyclic minimal DFAs, and proves that the tight bound for this subclass is $2^{n-1}$, where $n$ is the number of states of the input acyclic DFA. This bound can be met by an acyclic DFA over a ternary alphabet with a dead state, or by an acyclic

⋆ Research supported by VEGA grant 2/0183/11 and by grant APVV-0035-10.
⋆⋆ Research supported by the GAČR grant P202/11/P028, and by RVO: 67985840.

DFA over a growing alphabet without the dead state. It remains open whether or not the upper bound can be met by an acyclic DFA over a binary alphabet independently on the presence of the dead state, as well as by an acyclic DFA over a fixed alphabet that has no dead state.

The exponential blow-up of states for this operation motivates the following computational complexity problem: Given an acyclic minimal DFA accepting a regular language, is the minimal DFA for the reverse of the language also acyclic? Surprisingly, the answer to this question depends only on a certain structural property of the input automaton which can be tested by a known algorithm with a quadratic-time complexity with respect to the size of the input automaton. This means that we do not need to compute the whole automaton for the reverse to answer the question. Although this result can be derived from other results concerning piecewise testable languages, as discussed in the conclusions, as far as the authors know it has never been proved directly in this context. Therefore, in the second part of this paper, we prefer to present a direct proof of the fact that the reverse is acyclic if and only if the original minimal acyclic automaton satisfies a structural property discussed below.

This problem can be generalized to many other operations and types of automata. It deserves attention especially in the case of operations that are of interest in practical applications and have exponential state complexity, such as projections or abstractions for DFAs [1, 6, 8, 9].

## 2 Preliminaries and Definitions

The cardinality of a set $\Sigma$ is denoted by $|\Sigma|$. An alphabet is a finite non-empty set. The free monoid generated by an alphabet $\Sigma$ is denoted by $\Sigma^*$. A string over $\Sigma$ is any element of $\Sigma^*$. The empty string (the identity of $\Sigma^*$) is denoted by $\varepsilon$. The length of a string $w$ is denoted by $|w|$. A language over $\Sigma$ is any subset of $\Sigma^*$.

A *nondeterministic finite automaton* (NFA) is a 5-tuple $N = (Q, \Sigma, \delta, Q_0, F)$, where $Q$ is a finite non-empty set of states, $\Sigma$ is an input alphabet, $Q_0 \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final states, and $\delta : Q \times \Sigma \to 2^Q$ is a transition function which can be inductively extended to the domain $2^Q \times \Sigma^*$. The language *accepted* by $N$ is defined as the set $L(N) = \{w \in \Sigma^* \mid \delta(Q_0, w) \cap F \neq \emptyset\}$.

An NFA $N = (Q, \Sigma, \delta, Q_0, F)$ is a *complete deterministic finite automaton* (DFA) if $|Q_0| = 1$, and $|\delta(q, a)| = 1$ for each state $q$ in $Q$ and each input symbol $a$ in $\Sigma$. In that case, we identify singleton sets of states with their elements, that is, we write $q$ for a singleton set $\{q\}$. Moreover, we consider the transition function $\delta$ to be a total mapping from $Q \times \Sigma$ to $Q$ that can be extended to the domain $Q \times \Sigma^*$.

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 one. Otherwise, the two states are *equivalent*. A DFA is *minimal* if all its states are reachable from the initial state, and no two different states are equivalent. A DFA is *acyclic* if all strongly connected components [4] of the directed graph of the DFA are trivial, that is, they consist only of one element [7, 15, 16, 18]. Note that this definition allows self-loops.

The *subset automaton* corresponding to an NFA $N = (Q, \Sigma, \delta, Q_0, F)$ is the DFA $N' = (2^Q, \Sigma, \delta', Q_0, F')$, in which $F' = \{R \subseteq Q \mid R \cap F \neq \emptyset\}$ and $\delta'(R, a) = \delta(R, a)$ for each set $R$ in $2^Q$ and each symbol $a$ in $\Sigma$. The subset automaton $N'$ accepts the same language as the automaton $N$, but it need not be minimal since some of its states may be unreachable or equivalent.

The *reverse* $w^R$ *of a string* $w$ is inductively defined as follows: $\varepsilon^R = \varepsilon$ and $(va)^R = av^R$ for a string $v$ in $\Sigma^*$ and a symbol $a$ in $\Sigma$. The *reverse of a language* $L$ is the language $L^R = \{w^R \mid w \in L\}$. The *reverse of a DFA* $M = (Q, \Sigma, \delta, q_0, F)$ is the NFA $M^R$ obtained from $M$ by reversing all the transitions and by swapping the role of the initial and final states, that is, $M^R = (Q, \Sigma, \delta^R, F, \{q_0\})$, where $\delta^R(q, a) = \{p \in Q \mid \delta(p, a) = q\}$. It is known that the states of the subset automaton corresponding to the reverse of a minimal DFA are pairwise distinguishable [2, 3, 11]. For the sake of completeness, we give a short proof of this fact here.

**Lemma 1 ([2, 3, 11]).** *All distinct states of the subset automaton corresponding to the reverse of a minimal DFA are pairwise distinguishable.*

*Proof.* Let $M^R$ be the reverse of a minimal DFA $M$. Let $q$ be an arbitrary state of the NFA $M^R$. Since state $q$ is reachable in $M$, there exists a string $w_q$ accepted by $M^R$ from $q$. Furthermore, the string $w_q$ is not accepted from any other state of $M^R$; otherwise, there would be two distinct computations of the DFA $M$ on the string $w_q^R$. It follows that the states of the subset automaton corresponding to $M^R$ are pairwise distinguishable since two distinct subsets of the state set of $M^R$ must differ in a state $q$, and therefore the two subsets are distinguished by the string $w_q$. □

## 3 Main Results

This section presents the main results of this paper. First, we show that the worst-case state complexity of the reverse of a language represented by a minimal acyclic DFA is exponential in the number of states of the DFA. As a consequence of this result, we get that the direct construction of the minimal automaton for the reverse may be computationally unfeasible. This motivates the study of structural properties that would be helpful in deciding the question whether or not the minimal DFA for the reverse of a language is acyclic, if the language is represented by a minimal acyclic DFA. We prove that the acyclicity of the minimal DFA for the reverse is equivalent to a structural property testable in quadratic time.

Recall that in the general case, the worst-case state complexity of the reverse of a language represented by an $n$-state DFA is $2^n$ [5, 10–12, 19]. Our next result shows that for acyclic DFAs, the upper bound on the state complexity of the reverse is $2^{n-1}$.

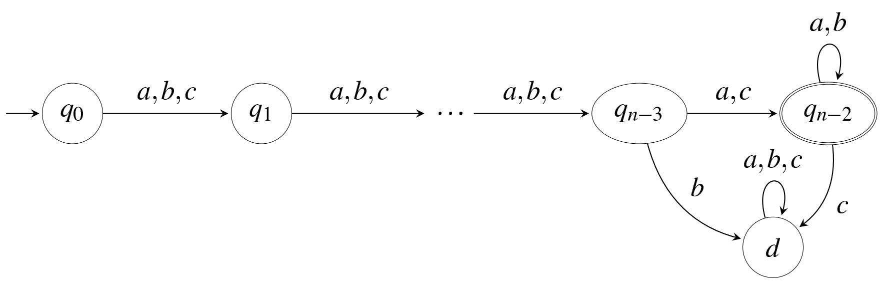


**Fig. 1.** The minimal acyclic DFA with the exponential reverse.

**Lemma 2.** *Let $M$ be an acyclic minimal DFA with $n$ states. Then the minimal DFA accepting the reverse of the language $L(M)$ has no more than $2^{n-1}$ states.*

*Proof.* Let $M = (Q, \Sigma, \delta, q_0, F)$ be an $n$-state acyclic minimal DFA, and construct the NFA $M^R$ for the reverse by swapping the role of the initial and final states, and by reversing all transitions. As $M$ is acyclic, we can topologically order its states from left to right so that no transition goes from right to left. Let $q$ be the rightmost state in this order. Since $M$ is complete, $q$ has self-loops under all symbols from $\Sigma$. If $q$ is not final, it is the dead state of $M$, and we can remove it before constructing $M^R$, that is, the subset automaton corresponding to $M^R$ has no more than $2^{n-1}$ states. On the other hand, if $q$ is final, it appears because of the self-loops in all reachable states of the subset automaton corresponding to $M^R$. This again gives the upper bound $2^{n-1}$ on the number of states. The proof is complete. □

The following results show that the upper bound is tight.

**Lemma 3.** *There exists an acyclic minimal DFA $M$ with $n$ states over the alphabet $\{a, b, c\}$ such that the minimal DFA accepting the reverse of the language $L(M)$ has $2^{n-1}$ states.*

*Proof.* Consider the DFA shown in Fig. 1. To construct its reverse, omit the dead state $d$, make state $q_{n-2}$ initial and state $q_0$ final, and reverse all the transitions. To simplify the proof, rename the states of the resulting NFA as shown in Fig. 2. We show that each subset of $\{0, 1, \ldots, n-2\}$ is reachable in the corresponding subset automaton.

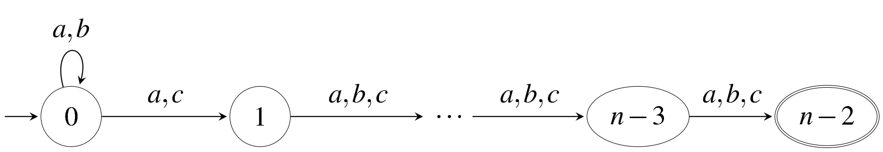


**Fig. 2.** The reverse of the DFA shown in Fig. 1; states renamed for the simplicity of the proof.

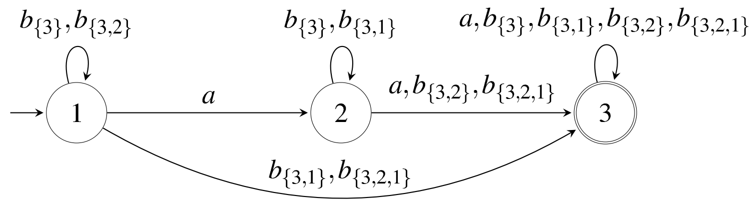


**Fig. 3.** The minimal acyclic DFA without the dead state with the exponential reverse.

The proof is by induction on the size of subsets. Each singleton set $\{i\}$ is reached from the initial state $\{0\}$ by $c^i$. Each subset $\{i_1, i_2, \ldots, i_k\}$ of size $k$, where $2 \le k \le n-1$ and $0 \le i_1 < i_2 < \cdots < i_k \le n-2$, is reached from the set $\{0, i_3 - i_2, i_4 - i_2, \ldots, i_k - i_2\}$ of size $k-1$ by the string $ab^{i_2 - i_1 - 1}c^{i_1}$ since

$$\begin{aligned}
&\{0, i_3 - i_2, i_4 - i_2, \ldots, i_k - i_2\} \xrightarrow{a} \\
&\{0, 1, i_3 - i_2 + 1, i_4 - i_2 + 1, \ldots, i_k - i_2 + 1\} \xrightarrow{b^{i_2 - i_1 - 1}} \\
&\{0, i_2 - i_1, i_3 - i_1, i_4 - i_1, \ldots, i_k - i_1\} \xrightarrow{c^{i_1}} \{i_1, i_2, i_3, i_4, \ldots, i_k\}\,.
\end{aligned}$$

This gives $2^{n-1}$ reachable states of the subset automaton, which are all pairwise distinguishable by Lemma 1. □

Note that the bound $2^{n-1}$ in the previous lemma follows naturally from the presence of the dead state, which is ignored in the construction of the reversed automaton. The next lemma shows, however, that the bound $2^{n-1}$ can also be met by an acyclic DFA without the dead state, but in this case we need an alphabet of exponential cardinality in comparison with the number of states, and it is not known whether the cardinality can be fixed.

**Lemma 4.** *There exists an acyclic minimal $n$-state DFA $M$ without the dead state over a growing alphabet such that the minimal DFA accepting the reverse of the language $L(M)$ has $2^{n-1}$ states.*

*Proof.* Let $\Sigma_n = \{a\} \cup \{b_S \mid S \subseteq \{1, 2, \ldots, n\} \text{ and } n \in S\}$ be an alphabet consisting of a symbol $a$, and $2^{n-1}$ symbols $b_S$ – one for each subset $S$ of $\{1, 2, \ldots, n\}$ with $n \in S$.

Define an $n$-state acyclic DFA $M$ over $\Sigma_n$ with the state set $\{1, 2, \ldots, n\}$, where 1 is the initial state and $n$ is the sole final state. By symbol $a$, state $n$ goes to itself, and every other state $i$ goes to state $i+1$. By symbol $b_S$, every state in $S$ goes to state $n$, and every other state goes to itself. Fig. 3 demonstrates this construction for $n = 3$.

In the subset automaton corresponding to the reverse of the DFA $M$, each subset $S$ of $\{1, 2, \ldots, n\}$ containing state $n$ is reached from the initial state $\{n\}$ by the symbol $b_S$. By Lemma 1, all these states are pairwise distinguishable, and the lemma follows. □

As a consequence of the previous three lemmata we get the following result.

**Theorem 1.** *Let $L$ be a language accepted by an acyclic minimal DFA with $n$ states. Then the minimal DFA accepting the reverse of the language $L$ has at most $2^{n-1}$ states. The bound is met by a ternary acyclic DFA with the dead state, or by an acyclic DFA over a growing alphabet without the dead state.* □

Now we turn to the problem whether the minimal DFA for the reverse of an acyclic minimal DFA is also acyclic. Theorem 1 implies that it may be computationally unfeasible to directly construct the minimal DFA for the reverse. Therefore, we study structural properties of acyclic minimal DFAs to solve the problem. To this end, we need several definitions.

For two states $p$ and $q$ of a DFA $M = (Q, \Sigma, \delta, q_0, F)$, we write $p \prec q$ if $p \neq q$ and state $q$ is reachable from state $p$, that is, there exists a string $w$ in $\Sigma^*$ such that $q = \delta(p, w)$. A state $p$ is called *maximal* if there exists no state $q$ such that $p \prec q$. Denote by $\Sigma(q)$ the set of all symbols appearing on the self-loops of state $q$, that is, $\Sigma(q) = \{a \in \Sigma \mid \delta(q, a) = q\}$.

Let $\Sigma_i \subseteq \Sigma$ and $\delta_i$ be the restriction of the transition function $\delta$ of the DFA $M$ to the domain $Q \times \Sigma_i$. Denote by $\Gamma(\Sigma_i)$ the directed graph obtained from the deterministic automaton $(Q, \Sigma_i, \delta_i, q_0, F)$ by ignoring the labels of edges and eliminating the multi-edges. A *connected component* of the directed graph $\Gamma(\Sigma_i)$ with respect to a node $q$ is the set of all nodes which are connected with $q$ by a path disregarding the orientation of edges.

The following theorem characterizes the structural property which will be useful to derive the polynomial-time algorithm testing acyclicity of the reversed automaton. Although this result can be indirectly derived from other results concerning piecewise testable languages, as discussed in the conclusions, we prefer to give a direct proof of this fact here.

**Theorem 2.** *Let $M$ be an acyclic minimal DFA. The minimal DFA accepting the reverse of the language $L(M)$ is acyclic if and only if for each state $p$ of $M$, the connected component of the graph $\Gamma(\Sigma(p))$ containing state $p$ has a unique maximal state with respect to the relation $\prec$.*

*Proof.* Let $M = (Q, \Sigma, \delta, q_0, F)$ be an acyclic minimal DFA and assume that the minimal DFA for the reverse, denoted by

$$M' = (Q', \Sigma, \delta', F, \{R \subseteq Q \mid q_0 \in R\}),$$

where $Q' \subseteq 2^Q$, is acyclic. The DFA $M'$ is obtained from $M$ by setting $F$ to be the set of initial states, reversing all the transitions, converting the obtained NFA to a DFA, and minimizing the DFA. Each subset containing the initial state $q_0$ of $M$ is set to be a final state of $M'$.

Assume that $M'$ is acyclic. For the sake of contradiction, assume that there exists a state $p$ in $Q$ such that the connected component of the graph $\Gamma(\Sigma(p))$ containing state $p$ has two distinct maximal states. Since state $p$ is a maximal state of this component, there exists a state $q$ in that component that is maximal and different from $p$. The DFA $M$ is acyclic, thus either $p \not\prec q$ or $q \not\prec p$. Without loss of generality, we assume that $q \not\prec p$. Then, there exist a state $r$ in $Q$ and two strings $u, v$ in $\Sigma(p)^*$ such that $\delta(r,u) = p$ and $\delta(r,v) = q$. Since $M$ is minimal, states $p$ and $q$ are distinguished by a string $w$ in $\Sigma^*$. Let $w$ be accepted from $p$ and rejected from $q$ as depicted in Fig. 4; the other case is symmetric.

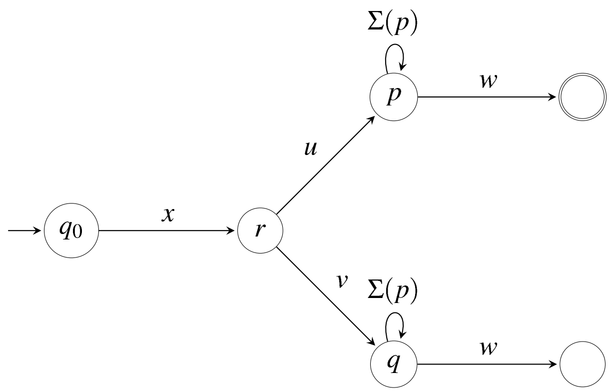


**Fig. 4.** Two maximal states $p$ and $q$ of the component $\Gamma(\Sigma(p))$ containing $p$.

Consider the computation of $M'$ on the string

$$w^R\, u^R\, v^R\, u^R\, v^R\, u^R\, v^R\, u^R\, v^R \ldots,$$

and let the computation be

$$F \xrightarrow{w^R} Z \xrightarrow{u^R} X_1 \xrightarrow{v^R} Y_1 \xrightarrow{u^R} X_2 \xrightarrow{v^R} Y_2 \cdots .$$

Since $w$ is accepted by $M$ from $p$ but rejected from $q$, state $p$ is in $Z$ but $q$ is not. Moreover, since $p$ has a loop on each symbol in $\Sigma(p)$, it occurs in every $X_i$ and $Y_i$. Now, consider the state $r$. It occurs in every set $X_i$ since $p$ goes to $r$ by $u^R$ in $M'$. However, $r$ does not occur in any $Y_i$ because otherwise we would have

$$r \xrightarrow{v} q \xrightarrow{uvuvuv\cdots uvu} q \xrightarrow{w} f$$

in $M$ for a final state $f$ of $M$; thus, string $w$ would be accepted from state $q$, which is a contradiction. Now consider a sequence $X_1, Y_1, X_2, Y_2, \ldots$ of subsets of the states of $M$. Since we only have a finite number of such subsets, there exists a cycle in this sequence. Let $X$ and $X'$ be two consecutive subsets on this cycle. Then state $r$ is in exactly one of $X$ and $X'$. Without loss of generality, let $r \in X$. Since $M$ is minimal, state $r$ is reached in $M$ from the initial state $q_0$ by a string $x \in \Sigma^*$. It follows that $x^R$ is accepted from $X$ in $M'$. On the other hand, since $M$ is deterministic and $r \notin X'$, string $x^R$ is not accepted from $X'$ in $M'$. Thus $X$ and $X'$ are not equivalent, and therefore the cycle is not a self-loop. This contradicts our assumption that $M'$ is acyclic.

To prove the converse implication, assume that for each state $p$ of $M$, the connected component of the graph $\Gamma(\Sigma(p))$ containing $p$ has a unique maximal state with respect to the relation $\prec$. For the sake of contradiction, assume that there exists a cycle of length at least two in the DFA $M'$. Let $S$ and $T$ be two different sets on this cycle. Without loss of generality, we can assume that there exists a state $r$ in $M$ with $r \notin S$ and $r \in T$. Assume that $S$ goes to $T$ by a string $u$, and $T$ goes to $S$ by a string $v$ on the cycle in $M'$, see Fig. 5. For $i \geq 0$, let $p_i = \delta(r, u^R(v^R u^R)^i)$ be the states of $M$ reached from the state $r$ by strings $u^R(v^R u^R)^i$. Then all the states $p_i$ belongs to $S$. Since $M$ is acyclic, there exists $j$ such that $p_j$ goes to itself on each symbol occurring in $uv$, denoted by $\Sigma(uv)$. Since $p_j$ is in $S$ and goes to itself on each symbol from $\Sigma(uv)$, it is also in $T$. Denote $p = p_j$. Then $p$ is maximal with respect to $\Sigma(uv)$. Now the aim is to find another maximal state in the connected component of $\Gamma(\Sigma(uv))$ containing $p$.

To this aim, let $s_i = \delta(r, v^R(u^R v^R)^i)$ for $i \geq 0$. Since $M$ is acyclic, there exists an index $k$ such that $s_k$ goes to itself on each symbol from $\Sigma(uv)$. Set $q = s_k$. State $q$ is in the same connected component as $p$ since both $p$ and $q$ are reached from $r$ in $M$. We need to show that $q \neq p$. Assume to the contrary that $q = p$. Then state $r$ is reached in $M'$ from state $p$ by the string $(vu)^k v$. Since state $p$ is in $T$, state $r$ is in $S$, which is a contradiction. Hence states $p$ and $q$ are distinct maximal states in the same connected component of the graph $\Gamma(\Sigma(uv))$. Since $M$ is acyclic, either $p \not\prec q$ or $q \not\prec p$. Assume that $q \not\prec p$, and consider the graph $\Gamma(\Sigma(p))$. Then $\Gamma(\Sigma(uv)) \subseteq \Gamma(\Sigma(p))$, state $p$ is maximal with respect to $\Sigma(p)$, and states $p$ and $q$ are connected in the graph $\Gamma(\Sigma(p))$. State $q$ or a successor of $q$ is maximal in the same connected component of $\Gamma(\Sigma(p))$, but it is different from $p$ because $q \not\prec p$. □

Now we demonstrate this technique on the following example.

*Example 1.* Consider the minimal DFA depicted in Fig. 6 (left). We have $\Sigma(1) = \{a, b\}$. Fig. 6 (right) shows the graph $\Gamma(\Sigma(1))$. The only connected component of $\Gamma(\Sigma(1))$ has two maximal states, namely 1 and $d$. By Theorem 2, the minimal DFA accepting the reverse of the language accepted by the DFA in Fig. 6 (left) has a cycle, as shown in Fig. 7. □

Notice that this technique requires to consider complete minimal DFAs and it works neither for incomplete DFAs nor for complete DFAs that are not minimal. The previous example does not work if we ignore the dead state. In addi-

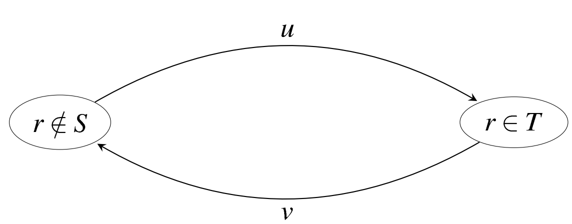


**Fig. 5.** A cycle in the minimal DFA $M'$ for the reverse.

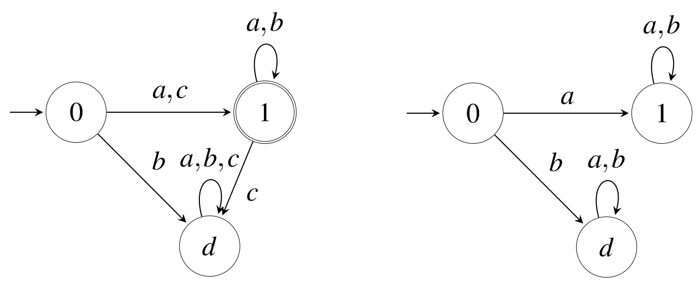


**Fig. 6.** An acyclic DFA and its graph $\Gamma(\Sigma(1))$.

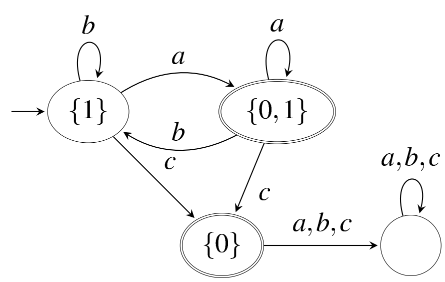


**Fig. 7.** The minimal DFA for the reverse of the language accepted by the DFA in Fig. 6 (left).

tion, in the case of non-minimal automata, we can have two different maximal accepting/non-accepting states that can be equivalent.

The condition whether for each state $p$ of $M$, the connected component of $\Gamma(\Sigma(p))$ containing state $p$ has a unique maximal state with respect to the relation $\prec$ can be tested using the algorithm presented by Trahtman [18]. The algorithm runs in time $O(n^2)$, where $n$ is the sum of the number of states and the number of transitions in $M$. As a consequence, we have the following theorem.

**Theorem 3.** *Let $M$ be an acyclic minimal deterministic finite automaton with $m$ states and $k$ transitions. Let $n = mk$. There exists an algorithm solving the problem of acyclicity of the minimal deterministic automaton for the reverse of the language $L(M)$ in time $O(n^2)$.* □

## 4 Conclusions

We discussed the state complexity of acyclic minimal DFAs, and the problem of deciding whether or not the minimal DFA for the reverse of a language is acyclic if the language is represented by an acyclic minimal DFA. We showed that the minimal DFA for the reverse is acyclic if and only if the minimal acyclic DFA for the original language possesses a special structural property. This property can be tested in quadratic time using the result of Trahtman [18], even though the construction of the minimal DFA for the reverse may be exponential.

We could also ask the opposite question: Is there a structural property ensuring that the minimal DFA for the reverse of a language is acyclic if the language is represented by a minimal DFA with a cycle? As far as the authors know, this question is open. Let us also mention that the work by Trahtman is motivated by the investigation of a proper subclass of the class of regular languages, the class of so-called piecewise testable languages introduced by Simon in [13].

A *piecewise testable language* over an alphabet $A$ is a finite boolean combination of languages of the form $A^*a_1A^*a_2A^* \dots A^*a_kA^*$, where $k \geq 0$ and $a_i \in A$. Simon [14] characterized piecewise testable languages as the class of languages with $\mathcal{J}$-trivial syntactic monoids, see also Stern [15]. Stern suggested a polynomial-time algorithm of order $O(n^5)$ deciding whether or not a regular language is piecewise testable in [16]. Trahtman [18] improved this result by presenting an algorithm running in time quadratic in the size of the input, and provided a package TESTAS implementing the algorithm in [17].

Recently, Polák and Klíma [7] have mentioned another method for the verification of piecewise testability of a regular language. However, this method is based on the construction of a so-called *biautomaton*, which requires both the minimal DFA for a language and the minimal DFA for its reverse. According to Theorem 3, this construction may be unfeasible because of the complexity reasons.

**Acknowledgements.** The authors gratefully acknowledge useful suggestions and comments of the anonymous referees.

## References

1. Boutin, O., Komenda, J., Masopust, T., Schmidt, K., van Schuppen, J.H.: Hierarchical control with partial observations: Sufficient conditions. In: Proc. of IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 2011). pp. 1817–1822. Orlando, Florida, USA (2011)
2. Brzozowski, J.A.: Canonical regular expressions and minimal state graphs for definite events. In: Proc. of the Symposium on Mathematical Theory of Automata. MRI Symposia Series, vol. 12, pp. 529–561. Polytechnic Institute of Brooklyn, New York (1963)
3. Champarnaud, J.M., Khorsi, A., Paranthoën, T.: Split and join for minimizing: Brzozowski's algorithm, [Online]. http://jmc.feydakins.org/ps/c09psc02.ps
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. The MIT Press, 3rd edn. (2009)
5. Jirásková, G., Šebej J.: Note on reversal of binary regular languages. In: Proc. of DCFS 2011. Lecture Notes in Computer Science, vol. 6808, pp. 212–221. Springer (2011)
6. Jirásková, G., Masopust, T.: On a structural property in the state complexity of projected regular languages. Theoretical Computer Science (2012), in press, doi:10.1016/j.tcs.2012.04.009
7. Klíma, O., Polák, L.: On biautomata. In: Proc. of NCMA 2011. books@ocg.at, vol. 282, pp. 153–164. Austrian Computer Society (2011)

8. Komenda, J., Masopust, T., van Schuppen, J.H.: Synthesis of controllable and normal sublanguages for discrete-event systems using a coordinator. Systems & Control Letters 60(7), 492–502 (2011)
9. Komenda, J., Masopust, T., van Schuppen, J.H.: Supervisory control synthesis of discrete-event systems using a coordination scheme. Automatica 48(2), 247–254 (2012)
10. Leiss, E.: Succinct representation of regular languages by boolean automata. Theoretical Computer Science 13, 323–330 (1981)
11. Mirkin, B.G.: On dual automata. Kibernetika 2, 7–10 (1966), in Russian. English translation: Cybernetics 2, 6–9 (1966)
12. Salomaa, A., Wood, D., Yu, S.: On the state complexity of reversals of regular languages. Theoretical Computer Science 320, 315–329 (2004)
13. Simon, I.: Hierarchies of Events with Dot-Depth One. Ph.D. thesis, Dep. of Applied Analysis and Computer Science, University of Waterloo, Canada (1972)
14. Simon, I.: Piecewise testable events. In: Proc. of GI Conference on Automata Theory and Formal Languages. pp. 214–222. Springer (1975)
15. Stern, J.: Characterizations of some classes of regular events. Theoretical Computer Science 35, 17–42 (1985)
16. Stern, J.: Complexity of some problems from the theory of automata. Information and Control 66(3), 163–176 (1985)
17. Trahtman, A.: A package TESTAS for checking some kinds of testability. In: Proc. of CIAA 2003, Lecture Notes in Computer Science, vol. 2608, pp. 265–272. Springer (2003)
18. Trahtman, A.N.: Piecewise and local threshold testability of DFA. In: Proc. of FCT 2001. Lecture Notes in Computer Science, vol. 2138, pp. 347–358. Springer (2001)
19. Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theoretical Computer Science 125(2), 315–328 (1994)