International Journal of Foundations of Computer Science

STATE COMPLEXITY OF CONCATENATION AND COMPLEMENTATION

JOZEF JIRÁSEK
Institute of Computer Science, P.J. Šafárik University, Jesenná 5, 040 01 Košice, Slovakia

and

GALINA JIRÁSKOVÁ *
Mathematical Institute, Slovak Academy of Sciences, Grešákova 6, 040 01 Košice, Slovakia

and

ALEXANDER SZABARI
Institute of Computer Science, P.J. Šafárik University, Jesenná 5, 040 01 Košice, Slovakia

Received (received date)
Revised (revised date)
Communicated by Editor's name

ABSTRACT

We investigate the state complexity of concatenation and the nondeterministic state complexity of complementation of regular languages. We show that the upper bounds on the state complexity of concatenation are also tight in the case that the first automaton has more than one accepting state. In the case of nondeterministic state complexity of complementation, we show that the entire range of complexities, up to the known upper bound can be produced.

Keywords: Regular languages; State complexity; Nondeterministic state complexity.

1. Introduction

Finite automata are one of the simplest computational models. Despite their simplicity, some challenging problems concerning finite automata are still open. For instance, we recall the question of how many states are sufficient and necessary for two-way deterministic finite automata to simulate two-way nondeterministic finite automata. The importance of this problem is underlined by its relation to the well-known open question whether or not DLOGSPACE equals NLOGSPACE [23].

*Research supported by the VEGA grant No. 2/3164/23.

Recently, a renewed interest in regular languages and finite automata can be observed. For a discussion, the reader may refer to [13, 27]. Some aspects of this area are now intensively investigated. One such aspect is the state complexity of regular languages and their operations.

The state complexity of a regular language is the number of states of its minimal deterministic finite automaton (DFA). The nondeterministic state complexity of a regular language is the number of states of a minimum state nondeterministic finite automaton (NFA) accepting the language. The state complexity (the nondeterministic state complexity) of an operation on regular languages represented by DFAs (NFAs, respectively) is the number of states that are sufficient and necessary in the worst case for a DFA (an NFA, respectively) to accept the language resulting from the operation.

The state complexity of some operations on regular languages was investigated in [17, 1, 2]. Yu, Zhuang, and Salomaa [25] were the first to systematically study the complexity of regular language operations. Their paper was followed by several articles investigating the state complexity of finite language operations and unary language operations [3, 20, 21]. The nondeterministic state complexity of regular language operations was studied by Holzer and Kutrib in [9, 10, 11]. Further results on this topic are presented in [6, 4, 16] and state-of-the-art surveys for DFAs can be found in [29, 28].

In this paper, we investigate the state complexity of concatenation and the nondeterministic state complexity of complementation of regular languages. In the case of concatenation, we show that the upper bounds $m2^n - k2^{n-1}$ on the concatenation of an m-state DFA language and an n-state DFA language, where k is the number of accepting states in the m-state automaton, are tight for any integer k with $0 < k < m$. In the case of complementation, we show that for any positive integers n and m with $\log n \leq m \leq 2^n$, there is an n-state NFA language such that minimal NFAs for its complement have m states.

To prove the result on concatenation we show that a deterministic finite automaton is minimal. To obtain the result on complementation we use a fooling-set lower-bound technique known from communication complexity theory [12], cf. also [1, 2, 7].

The paper consists of five sections, including this introduction. The next section contains basic definitions and notations used throughout the paper. In Section 3 we present our result on concatenation. Section 4 deals with the problem of which kind of relations between the sizes of minimal NFAs for regular languages and minimal NFAs for their complements are possible. The last section contains concluding remarks and open problems.

2. Preliminaries

In this section, we recall some basic definitions and notations. For further details, the reader may refer to [24, 26].

Let Σ be an alphabet and Σ^* the set of all strings over the alphabet Σ including the empty string ε. The length of a string w is denoted by $|w|$ and the number of occurences of a symbol a in a string w by $\#_a(w)$. The power-set of a finite set A is denoted by 2^A and its maximum by $\max A$.

A *deterministic finite automaton* (DFA) is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is a finite input alphabet, $\delta : Q \times \Sigma \to Q$ is the transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of accepting states. In this paper, all DFAs are assumed to be complete, i.e., the next state $\delta(q, a)$ is defined for any state q in Q and any symbol a in Σ. The transition function δ is extended to a function from $Q \times \Sigma^*$ to Q in the natural way. A string w in Σ^* is accepted by the DFA M if the state $\delta(q_0, w)$ is an accepting state of the DFA M.

A *nondeterministic finite automaton* (NFA) is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$, where Q, Σ, q_0, and F are as above, and $\delta : Q \times \Sigma \to 2^Q$ is the transition function which can be naturally extended to the domain $Q \times \Sigma^*$. A string w in Σ^* is accepted by the NFA M if the set $\delta(q_0, w)$ contains an accepting state of the NFA M. We sometimes write $p \xrightarrow{w} q$ if $q \in \delta(p, w)$.

The *language accepted by* a finite automaton M, denoted $L(M)$, is the set of all strings accepted by the automaton M. Two automata are said to be *equivalent* if they accept the same language.

A DFA (an NFA) M is called *minimal* if all DFAs (all NFAs, respectively) that are equivalent to M have at least as many states as M. By a well-known result, each regular language has a unique minimal DFA, up to isomorphism. However, the same result does not hold for minimal NFAs.

Any nondeterministic finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ can be converted to an equivalent deterministic finite automaton $M' = (2^Q, \Sigma, \delta', q_0', F')$ using an algorithm known as the "subset construction" [22] in the following way. Every state of the DFA M' is a subset of the state set Q. The initial state of the DFA M' is $\{q_0\}$. A state R in 2^Q is an accepting state of the DFA M' if it contains an accepting state of the NFA M. The transition function δ' is defined by $\delta'(R, a) = \bigcup_{r \in R} \delta(r, a)$ for any state R in 2^Q and any symbol a in Σ. The DFA M' need not be minimal since some of its states may be unreachable or equivalent.

3. Concatenation

We start our investigation with concatenation operation. The state complexity of concatenation of regular languages represented by deterministic finite automata was studied by Yu *et al.* [25]. They showed that $m2^n - k2^{n-1}$ states are sufficient for a DFA to accept the concatenation of an $m-$state DFA language and an $n-$state DFA language, where k is the number of the accepting states in the $m-$state DFA. In the case of $n = 1$, upper bound m was shown to be tight, even for a unary alphabet. In the case of $m = 1$ and $n \geq 2$, the worst case $2^n - 2^{n-1}$ was given by the concatenation of two binary languages. Otherwise, the upper bound $m2^n - 2^{n-1}$ was shown to be tight for a binary alphabet in [16]. In the case of unary languages, the upper bound on concatenation is mn and it is known to be tight if m and n are relatively prime [25]. The unary case when m and n are not necessarily relatively prime was studied by Pighizzini and Shallit in [20, 21]. In this case, the tight bounds are given by the number of states in the noncyclic and in the cyclic parts of the

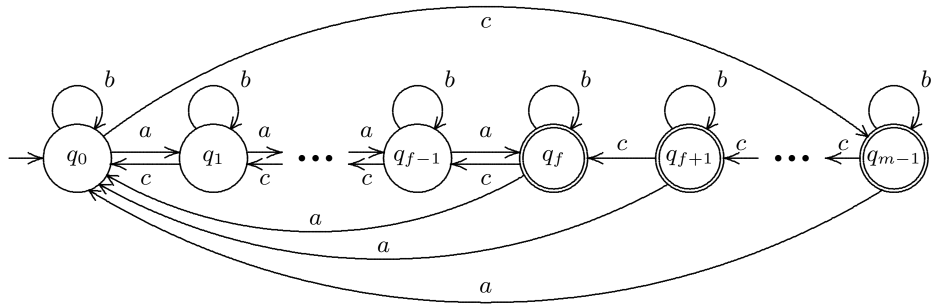

Fig. 1. text pod obrazkom 1

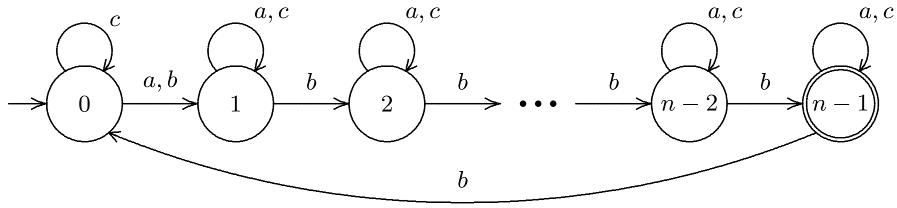

Fig. 2. text pod obrazkom 2

resulting automata. The next theorem shows that the upper bounds $m2^n - k2^{n-1}$ are also tight for any integer k with $0 < k < m$.

Theorem 1 *For any integers m, n, k such that $m \geq 2, n \geq 2$, and $0 < k < m$, there exist a DFA A of m states and k accepting states, and a DFA B of n states such that any DFA accepting the language $L(A)L(B)$ needs at least $m2^n - k2^{n-1}$ states.*

Proof. Let m, n, and k be arbitrary but fixed integers such that $m \geq 2, n \geq 2$, and $0 < k < m$. Let $\Sigma = \{a, b, c\}$.
Define an m-state DFA $A = (Q_A, \Sigma, \delta_A, q_0, F_A)$, where $Q_A = \{q_0, q_1, \ldots, q_{m-1}\}$, $F_A = \{q_{m-k}, q_{m-k+1}, \ldots, q_{m-1}\}$, and for any $i \in \{0, 1, \ldots, m-1\}$,

$$\delta_A(q_i, X) = \begin{cases} q_{i+1}, & \text{if } i < m-k \text{ and } X = a, \\ q_0, & \text{if } i \geq m-k \text{ and } X = a, \\ q_i, & \text{if } X = b, \\ q_{m-1}, & \text{if } i = 0 \text{ and } X = c, \\ q_{i-1}, & \text{if } i > 0 \text{ and } X = c. \end{cases}$$

Define an n-state DFA $B = (Q_B, \Sigma, \delta_B, 0, F_B)$, where $Q_B = \{0, 1, \ldots, n-1\}$, $F_B = \{n-1\}$, and for any $i \in \{0, 1, \ldots, n-1\}$,

$$\delta_B(i, X) = \begin{cases} 1, & \text{if } i = 0 \text{ and } X = a, \\ i, & \text{if } i > 0 \text{ and } X = a, \\ i+1, & \text{if } i < n-1 \text{ and } X = b, \\ 0, & \text{if } i = n-1 \text{ and } X = b, \\ i, & \text{if } X = c. \end{cases}$$

The DFA A and B are shown in Fig. 1 and Fig. 2, respectively.

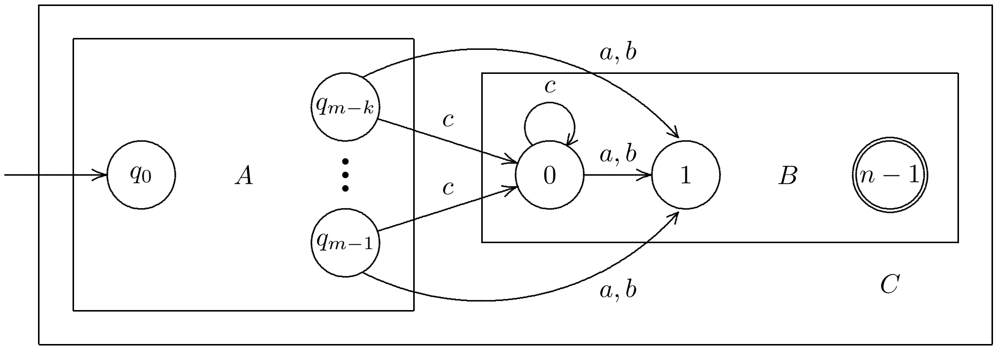

Fig. 3. text pod obrazkom 3

We first describe an NFA accepting the language $L(A)L(B)$, then we construct an equivalent DFA, and show that the DFA has at least $m2^n - k2^{n-1}$ reachable states no two of which are equivalent.

Consider the NFA $C = (Q, \Sigma, \delta, q_0, \{n-1\})$, where $Q = Q_A \cup Q_B$, and for any $q \in Q$ and any $X \in \Sigma$,

$$\delta(q, X) = \begin{cases} \{\delta_A(q, X)\}, & \text{if } q \in Q_A \setminus F_A, \\ \{\delta_A(q, X), \delta_B(0, X)\}, & \text{if } q \in F_A, \\ \{\delta_B(q, X)\}, & \text{if } q \in Q_B, \end{cases}$$

see Fig. 3.

Clearly, the NFA C accepts the language $L(A)L(B)$. Let $C' = (2^Q, \Sigma, \delta', \{q_0\}, F')$ be the DFA obtained from the NFA C by the subset construction. Let $\mathcal{R}$ be the following system of sets:

$$\mathcal{R} = \{\{q\} \cup S \mid q \in Q_A \setminus F_A \text{ and } S \subseteq Q_B\} \cup \{\{q\} \cup S \mid q \in F_A, S \subseteq Q_B \text{ and } 0 \in S\},$$

i.e., any set in $\mathcal{R}$ consists of exactly one state of Q_A and some states of Q_B, and if a set in $\mathcal{R}$ contains a state of F_A, then it also contains state 0. There are $m2^n - k2^{n-1}$ sets in $\mathcal{R}$. To prove the theorem it is sufficient to show that (I) any set in $\mathcal{R}$ is a reachable state of the DFA C' and (II) no two different states in $\mathcal{R}$ are equivalent.

We prove (I) by induction on the size of sets. The singletons $\{q_0\}, \{q_1\}, \ldots, \{q_{m-k-1}\}$ are reachable since $\{q_i\} = \delta'(\{q_0\}, a^i)$ for $i = 0, 1, \ldots, m-k-1$. Let $1 \le t \le n$ and assume that any set in $\mathcal{R}$ of size t is a reachable state of the DFA C'. Using this assumption we prove that any set $\{q_i, j_1, j_2, \ldots, j_t\}$, where
$0 \le j_1 < j_2 < \cdots < j_t < n$ if $0 \le i < m-k$, and
$0 = j_1 < j_2 < \cdots < j_t < n$ if $m-k \le i < m$,
is a reachable state of the DFA C'. There are two cases:

(i) $j_1 = 0$. Note that

- $\delta'(\{q_0\}, c^{m-i}) = \{q_i, 0\}$ for $i = 0, 1, \ldots, m-2$,
- $\delta'(\{q_0\}, c^{m+1}) = \{q_{m-1}, 0\}$, and

- $\delta'(\{j\}, c) = \{j\}$ for $j = 0, 1, \dots, n-1$.

Therefore we have $\{q_i, 0, j_2, j_3, \dots, j_t\} = \delta'(\{q_0, j_2, j_3, \dots, j_t\}, c^{m-i})$ for $i = 0, 1, \dots, m-2$, and $\{q_{m-1}, 0, j_2, j_3, \dots, j_t\} = \delta'(\{q_0, j_2, j_3, \dots, j_t\}, c^{m+1})$, where the set $\{q_0, j_2, j_3, \dots, j_t\}$ is reachable by induction.

(ii) $j_1 \geq 1$ and $0 \leq i < m - k$. Since $\delta'(\{j\}, a) = \{j\}$ for $j \geq 1$, we have $\{q_i, j_1, j_2, \dots, j_t\} = \delta'(\{q_0, 0, j_2 - j_1,\ j_3 - j_1, \dots, j_t - j_1\}, b^{j_1} a^i)$, where the latter set is considered in case (i).

To prove (II) let $\{q_i\} \cup S$ and $\{q_l\} \cup T$ be two different states in $\mathcal{R}$ with $0 \leq i \leq l \leq m - 1$. There are two cases:

(i) $i < l$. We will show that the string $c^i a^{m-k} b^{n-1}$ is accepted by the DFA C' starting in state $\{q_i\} \cup S$ but it is not accepted by the DFA C' starting in state $\{q_l\} \cup T$.

The string $c^i a^{m-k} b^{n-1}$ is accepted by the NFA C starting in state q_i because

$$q_i \xrightarrow{c^i} q_0 \xrightarrow{a^{m-k}} q_{m-k} \xrightarrow{b} 1 \xrightarrow{b^{n-2}} n-1.$$

The string $c^i a^{m-k} b^{n-1}$ is not accepted by the NFA C starting in any other state because

- the only way how to accept the string b^{n-1} is to start in state 0 or in a state of F_A;
- the states 0 and q_i with $i > m - k$ cannot be reached after reading the string $c^i a^{m-k}$ since $m - k \geq 1$ and there is no move on reading a into these states;
- the only way how to reach state q_{m-k} after reading a^{m-k} is to start in state q_0;
- the only way how to reach state q_0 after reading c^i is to start in state q_i.

It follows that the string $c^i a^{m-k} b^{n-1}$ is accepted by the DFA C' starting in state $\{q_i\} \cup S$ but it is not accepted by the DFA C' starting in state $\{q_l\} \cup T$.

(ii) $i = l$. Without loss of generality, there is a state j in Q_B such that $j \in S$ and $j \notin T$ (note that $j \geq 1$ if $m - k \leq i \leq m - 1$). We will show that the string b^{n-1-j} is accepted by the DFA C' starting in state $\{q_i\} \cup S$ but it is not accepted by the DFA C' starting in state $\{q_l\} \cup T$.

The string b^{n-1-j} is accepted by the DFA C' starting in state $\{q_i\} \cup S$ because

$$j \xrightarrow{b^{n-1-j}} n-1.$$

On the other hand, the string b^{n-1-j} is not accepted by the DFA C' starting in state $\{q_l\} \cup T$ because

- the string b^{n-1-j} is not accepted by the NFA C starting in any state of the set $Q_B \setminus \{j\}$ or starting in any state q_i with $i \leq m-k-1$,
- if $m-k \leq i \leq m-1$, then $j \geq 1$ and so the string b^{n-1-j} is not accepted by the NFA C starting in q_i since to reach state $n-1$ from state q_i at least $n-1$ symbols must be read.

Thus our proof is complete. □

4. Complementation

We now turn our attention to complementation operation. For DFAs, it is an efficient operation since to accept the complement we can simply exchange accepting and rejecting states. On the other hand, the complementation of NFAs is an expensive task. The upper bound on the size of an NFA accepting the complement of an $n-$state NFA language is 2^n and it is known to be tight for a binary alphabet [16]. For complementation of unary NFA languages a crucial role is played by the function $F(n) = \max\{\text{lcm}\,(x_1, \ldots, x_k) \mid x_1 + \ldots + x_k = n\}$. It is known that $F(n) \in e^{\Theta(\sqrt{n \ln n})}$ and that $O(F(n))$ states suffice to simulate any unary $n-$state NFA by a DFA [5]. This means that $O(F(n))$ states are sufficient for an NFA to accept the complement of an $n-$state unary NFA language. The lower bound is known to be $F(n-1)+1$ in this case [16].

In this section, we deal with the question of which kind of relations between the nondeterministic complexity of a regular language and the nondeterministic complexity of its complement are possible. We provide a complete solution by showing that for any positive integers n and m with $\log n \leq m \leq 2^n$, there exists an $n-$state NFA language such that minimal NFAs for its complement have m states.

To obtain the above result we use a fooling-set lower-bound technique known from communication complexity theory [12]. Although lower bounds based on fooling sets may sometimes be exponentially smaller than the true bounds [14, 15], for some regular languages the lower bounds are tight [1, 2, 7]. In this section, the technique helps us to obtain tight lower bounds. After defining a fooling set, we give the lemma from [1] describing a fooling-set lower-bound technique. For the sake of completeness, we recall its proof here. Then, we give an example.

Definition 1 *A set of pairs of strings $\{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$ is said to be a fooling set for a regular language L if for any i and j in $\{1, 2, \ldots, n\}$,*

(1) $x_i y_i \in L$, and

(2) if $i \neq j$ then $x_i y_j \notin L$ or $x_j y_i \notin L$.

Lemma 1 (Birget [1]) *Let a set of pairs $\{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$ be a fooling set for a regular language L. Then any NFA for the language L needs at least n states.*

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be any NFA accepting the language L. Since $x_i y_i \in L$, there is a state p_i in Q such that $p_i \in \delta(q_0, x_i)$ and $\delta(p_i, y_i) \cap F \neq \emptyset$. Assume that a fixed choice of p_i has been made for any i in $\{1, 2, \ldots, n\}$. We prove that $p_i \neq p_j$ for $i \neq j$. Suppose by contradiction that $p_i = p_j$ for some $i \neq j$. Then the NFA M accepts both strings $x_i y_j$ and $x_j y_i$ which contradicts the assumption

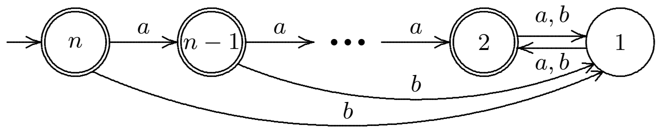

Fig. 4. text pod obrazkom 4

that the set $\{(x_i, y_i) \mid 1 \leq i \leq n\}$ is a fooling set for the language L. Hence the NFA M has at least n states. □

Example 1 Let $n \geq 1$, let $L_n = \{w \in \{a,b\}^* \mid \#_a(w) \equiv 0 \bmod n\}$, and let $\mathcal{A}_n = \{(a^i, a^{n-i}) \mid i = 1, 2, \dots, n\}$. Note that for any i and j in $\{1, 2, \dots, n\}$, (1) $a^i a^{n-i} \in L_n$, and (2) if $i \neq j$ then, w.l.o.g., $i < j$, so $0^i 0^{n-j} \notin L_n$. Hence the set $\mathcal{A}_n$ is a fooling set for the language L_n, and so any NFA for the language L_n needs at least n states.

We start our investigation with two propositions.

Proposition 1 *For any m in $\{1, 2\}$, there is a 1-state NFA D_m such that minimal NFAs for the complement of the language $L(D_m)$ have m states.*

Proof. Let $\Sigma = \{a, b\}$. Consider the following 1-state NFAs:

$D_1 = (\{s\}, \Sigma, \delta_1, s, \{s\})$ with $\delta_1(s, X) = \{s\}$ for any $X \in \Sigma$,

$D_2 = (\{s\}, \Sigma, \delta_2, s, \{s\})$ with $\delta_2(s, a) = \{s\}$ and $\delta_2(s, b) = \emptyset$.

The NFAs D_1 and D_2 do satisfy the proposition since the complement of the language $L(D_1)$ is the empty language, and the set of pairs of strings $\{(\varepsilon, b), (b, \varepsilon)\}$ is a fooling set for the complement of the language $L(D_2)$. □

Proposition 2 *For any integer $n \geq 2$, there is a minimal NFA N of n states such that minimal NFAs for the complement of the language $L(N)$ have n states.*

Proof. Let n be arbitrary but fixed integer with $n \geq 2$. Let $\Sigma = \{a, b\}$. Define an n-state NFA $N = (Q, \Sigma, \delta, n, F)$, see Fig. 4, where $Q = \{1, 2, \dots, n\}$, $F = \{2, 3, \dots, n\}$, and for any $i \in Q$,

$\delta(1, a) = \delta(1, b) = \{2\}$,

$\delta(i, a) = \{i - 1\}$ and $\delta(i, b) = \{1\}$ if $i > 1$.

We are going to show that

(a) the NFA N is a minimal NFA for the language $L(N)$;

(b) the language $L^c(N)$ is accepted by an n-state DFA;

(c) any NFA for the language $L^c(N)$ needs at least n states.

Then, the proposition follows.

Consider the set of pairs $\mathcal{A} = \{(a^{i-1}, a^{n-i}b) \mid i = 1, 2, \dots, n\}$. The set $\mathcal{A}$ is a fooling set for the language $L(N)$ because for any i and j in $\{1, 2, \dots, n\}$, (1) $a^{i-1}a^{n-i}b \in L(N)$ since the string $a^{n-1}b$ is accepted by the NFA N, and (2) if $i < j$, then $a^{i-1}a^{n-j}b \notin L(N)$ since any string $a^l b$ with $0 \leq l < n - 1$ is not accepted by the NFA N. By Lemma 1, any NFA for the language $L(N)$ needs at least n states which proves (a).

To prove (b) note that the NFA N is, in fact, deterministic, and so after exchanging the accepting and the rejecting states we obtain an n-state DFA for the language $L^c(N)$.

Finally, consider the set of pairs $\mathcal{B} = \{(a^{i-1}, a^{n-i}) \mid i = 1, 2, \ldots, n\}$. The set $\mathcal{B}$ is a fooling set for the language $L^c(N)$ because for any i and j in $\{1, 2, \ldots, n\}$,
(1) $a^{i-1}a^{n-i} \in L^c(N)$ since the string a^{n-1} is not accepted by the NFA N, and
(2) if $i < j$, then $a^{i-1}a^{n-j} \notin L^c(N)$ since any string a^l with $0 \leq l < n-1$ is accepted by the NFA N. By Lemma 1, any NFA for the language $L^c(N)$ needs at least n states and our proof is complete. □

The following theorem is proved in [16].

Theorem 2 ([16]) *For any positive integer n, there exists a binary NFA M of n states such that any NFA for the complement of the language $L(M)$ needs at least 2^n states.*

In the next theorem, we show that the nondeterministic state complexity of complements of $n-$state NFA languages may be arbitrary between $n+1$ and $2^n - 1$.

Theorem 3 *For any integers n and m with $3 \leq n+1 \leq m \leq 2^n - 1$, there exists a minimal NFA M of n states such that minimal NFAs for the complement of the language $L(M)$ have m states.*

Proof. Let n and m be arbitrary but fixed integers such that $3 \leq n+1 \leq m \leq 2^n - 1$. Then m can be expressed as $m = n+k$ for an integer k with $1 \leq k \leq 2^n - 1 - n$. Let

$$\Sigma = \{a, b\} \cup \{c_1, c_2, \ldots, c_k\} \cup \{d_1, d_2, \ldots, d_k\}$$

be a $(2k+2)$-letter alphabet. We are going to define a minimal $n-$state NFA M over the alphabet Σ such that minimal NFAs for the language $L^c(M)$ have $n+k$ states. To this aim let

$$S_1, S_2, \ldots, S_{2^n-1-n}$$

be a sequence of subsets of the set $\{1, 2, \ldots, n\}$ that contain at least two elements and are ordered in such a way that for any i and j in $\{1, 2, \ldots, 2^n - 1 - n\}$, the following two conditions hold:

(1) if $\max S_i < \max S_j$, then $i < j$;

(2) if $\max S_i = \max S_j$ and $1 \in S_i \setminus S_j$, then $i < j$,

i.e., the subsets are ordered according to their maxima, and if two sets have the same maximum, then all sets that contain the state 1 precede the sets that do not contain the state 1. Clearly, there are several such orderings, we choose one of them. Note that $S_1 = \{1, 2\}$. For example, the subsets of $\{1, 2, 3, 4\}$ that contain at least two elements could be ordered as follows: $S_1 = \{1, 2\}$, $S_2 = \{1, 3\}, S_3 = \{1, 2, 3\}$, $S_4 = \{2, 3\}$, $S_5 = \{1, 4\}$, $S_6 = \{1, 2, 4\}$, $S_7 = \{1, 3, 4\}, S_8 = \{1, 2, 3, 4\}$, $S_9 = \{2, 4\}$, $S_{10} = \{3, 4\}$, $S_{11} = \{2, 3, 4\}$.

Define an $n-$state NFA $M = (Q, \Sigma, \delta, n, F)$, where $Q = \{1, 2, \ldots, n\}$, $F =$

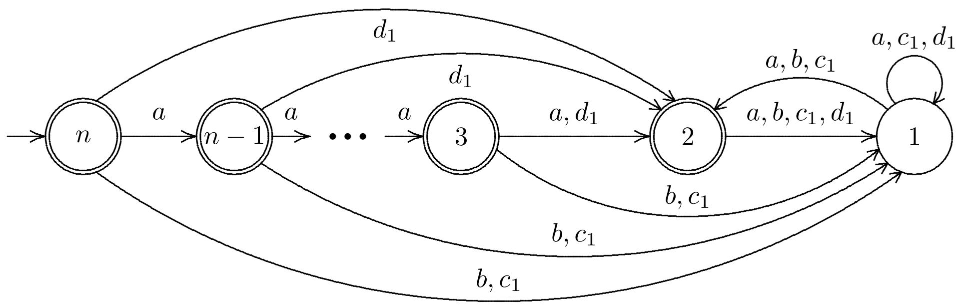

Fig. 5. text pod obrazkom 5

$\{2, 3, \ldots, n\}$, and for any $i \in Q$ and any $j \in \{1, 2, \ldots, k\}$,

$$\delta(i, X) = \begin{cases} \{1,2\}, & \text{if } i = 1 \quad \text{and } X = a, \\ \{i-1\}, & \text{if } i > 1 \quad \text{and } X = a, \\ \{2\}, & \text{if } i = 1 \quad \text{and } X = b, \\ \{1\}, & \text{if } i > 1 \quad \text{and } X = b, \\ S_j, & \text{if } i = 1 \quad \text{and } X = c_j, \\ \{1\}, & \text{if } i > 1 \quad \text{and } X = c_j, \\ \{1\}, & \text{if } i \in S_j \quad \text{and } X = d_j, \\ \{2\}, & \text{if } i \notin S_j \quad \text{and } X = d_j, \end{cases}$$

see Fig. 5.
We will show that
(a) the NFA M is a minimal NFA for the language $L(M)$;
(b) the language $L^c(M)$ can be accepted by an $(n+k)$-state DFA;
(c) any NFA for the language $L^c(M)$ needs at least $n+k$ states.
Then, the theorem follows immediately.

To prove (a) consider the following set of pairs of strings

$$\mathcal{A} = \{(a^{i-1}, a^{n-i}b) \mid i = 1, 2, \ldots, n\}.$$

The set $\mathcal{A}$ is a fooling set for $L(M)$ because for any i and j in $\{1, 2, \ldots, n\}$,

(1) $a^{i-1}a^{n-i}b \in L(M)$ since the string $a^{n-1}b$ is accepted by the NFA M, and

(2) if $i < j$, then $a^{i-1}a^{n-j}b \notin L(M)$ since for any l with $0 \leq l < n-1$, the string $a^l b$ is not accepted by the NFA M.

By Lemma 1, any NFA for $L(M)$ needs at least n states which proves (a).

To prove (b) let $M' = (2^Q, \Sigma, \delta', \{n\}, F')$ be the DFA obtained from the NFA M by the subset construction. Let $\mathcal{R}$ be the following system of sets

$$\mathcal{R} = \{\{1\}, \{2\}, \ldots, \{n\}, S_1, S_2, \ldots, S_k\}.$$

Note that the initial state $\{n\}$ of the DFA M' and the state $S_1 = \{1, 2\}$ belong to the system $\mathcal{R}$. We are going to prove that any set in $\mathcal{R}$ is a reachable state of the DFA M' and no other states are reachable in the DFA M'. Clearly, any set of the system $\mathcal{R}$ is reachable since we have $\{i\} = \delta'(\{n\}, a^{n-i})$ for $i = 1, 2, \ldots, n$, and $S_j = \delta'(\{1\}, c_j)$ for $j = 1, 2, \ldots, k$. To prove that no other subset of the set Q is a reachable state of the DFA M' it is sufficient to show that for any state R in $\mathcal{R}$ and any symbol X in Σ, the state $\delta'(R, X)$ is a member of the system $\mathcal{R}$. There are three cases:

(i) $R = \{1\}$. Then we have $(j = 1, 2, \ldots, k)$:

$$\delta'(\{1\}, X) = \begin{cases} \{1,2\}, & \text{if } X = a, \\ \{2\}, & \text{if } X = b, \\ S_j, & \text{if } X = c_j, \\ \{1\}, & \text{if } 1 \in S_j \text{ and } X = d_j, \\ \{2\}, & \text{if } 1 \notin S_j \text{ and } X = d_j. \end{cases}$$

Since all sets on the right are in the system $\mathcal{R}$, we are ready in this case.

(ii) $R = \{i\}$ for an $i \neq 1$. Then for any X in Σ, the set $\delta'(\{i\}, X)$ is a singleton set and so is in $\mathcal{R}$.

(iii) $R = S_j$ for a j in $\{1, 2, \ldots, k\}$. Then the set $\delta'(S_j, a)$ is a subset of the set $\{1, 2, \ldots, \max S_k - 1\}$ or equals $\{1, 2\}$. Since the sets $S_1, S_2, \ldots, S_k$ are ordered according to their maxima, any subset of $\{1, 2, \ldots, \max S_k - 1\}$ is in the system $\mathcal{R}$. Next, the set $\delta'(S_j, b)$ is equal either to $\{1\}$ or to $\{1, 2\}$, and the set $\delta'(S_j, d_l)$, $l = 1, 2, \ldots, k$, is equal either to $\{1\}$, or to $\{2\}$, or to $\{1, 2\}$. Finally, the set $\delta'(S_j, c_l)$, $l = 1, 2, \ldots, k$, is equal either to $\{1\}$ or to $S_l \cup \{1\}$. Since the set $S_l \cup \{1\}$ precedes the set S_l or equals S_l, we are ready in this case.

Thus we have shown that the DFA M' obtained from the NFA M by the subset construction has exactly $n + k$ reachable states. After exchanging the accepting and the rejecting states of the DFA M' we obtain an $(n + k)-$state DFA for the language $L^c(M)$ which proves (b).

To prove (c) consider the following sets of pairs of strings

$$\mathcal{B} = \{(a^{i-1}, a^{n-i}) \mid i = 1, 2, \ldots, n\},$$

$$\mathcal{C} = \{(a^{n-1}c_j, d_j) \mid j = 1, 2, \ldots, k\}.$$

We will show that the set $\mathcal{B} \cup \mathcal{C}$ is a fooling set for the language $L^c(M)$.

(1) For any $i \in \{1, 2, \ldots, n\}$, the string $a^{i-1}a^{n-i}$ is in the language $L^c(M)$ since the string a^{n-1} is not accepted by the NFA M.
For any $j \in \{1, 2, \ldots, k\}$, the string $a^{n-1}c_j d_j$ is in the language $L^c(M)$ since

$$\delta(n, a^{n-1}) = \{1\}, \quad \delta(\{1\}, c_j) = S_j, \quad \delta(S_j, d_j) = \{1\}, \text{ and } 1 \notin F,$$

and so the string $a^{n-1}c_j d_j$ is not accepted by the NFA M.

(2) If $1 \le i < s \le n$, then the string $a^{i-1}a^{n-s}$ is not in the language $L^c(M)$ since the NFA M accepts any string a^l with $0 \le l < n-1$.
Next, if $1 \le j, t \le k$ and $j \ne t$, then, w.l.o.g., there is a state p in Q such that $p \in S_j$ and $p \notin S_t$. Thus,

$$p \in \delta(n, a^{n-1}c_j) \text{ and } 2 \in \delta(p, d_t),$$

and so the string $a^{n-1}c_j d_t$ is accepted by the NFA M, i.e., is not in the language $L^c(M)$.
Finally, if $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, k\}$, then the string $a^{n-1}c_j a^{n-i}$ is not in the language $L^c(M)$ since $\delta(n, a^{n-1}c_j) = S_j$, the size of the set S_j is at least two, and the string a^{n-i} is not accepted by the NFA M starting in state $n-i+1$ but it is accepted by M starting in any other state.

Thus the set $\mathcal{B} \cup \mathcal{C}$ is a fooling set for the language $L^c(M)$. By Lemma 1, any NFA for the language $L^c(M)$ needs at least $n+k$ states which completes our proof. □

Corollary 1 *For any positive integers r and s with $\log r \le s \le r$, there exists a minimal NFA E of r states such that minimal NFAs for the complement of the language $L(E)$ have s states.*

Proof. Let r and s be arbitrary but fixed positive integers with $\log r \le s \le r$. Then we have

$$s \le r \le 2^s,$$

and by the above results, there is a minimal s-state NFA S such that a minimal NFA, say R, for the language $L^c(S)$ has r states. Set $E = R$. Then the NFA E is a minimal r-state NFA for the language $L^c(S)$, and minimal NFAs for the complement of the language $L^c(S)$, i.e., for $L^c(E)$, have s states. □

Hence, we have shown the following result.

Theorem 4 *For any positive integers n and m with $\log n \le m \le 2^n$, there exists a minimal NFA M of n states such that minimal NFAs for the complement of the language $L(M)$ have m states.*

5. Conclusions

In this paper, we obtained several results concerning the state complexity of concatenation and the nondeterministic state complexity of complementation of regular languages.

In the case of concatenation, we showed that the upper bounds $m2^n - k2^{n-1}$ on the concatenation of an m-state DFA language and an n-state DFA language, where k is the number of the accepting states in the m-state automaton, are tight for any integer k with $0 < k < m$. To prove the result, we used a three-letter alphabet. In the case of $m = 3$, $k = 2$, and $n = 2$, the upper bound can be reached by the concatenation of two binary languages. The problem remains open for a binary alphabet and larger values of m, k, n.

In the case of complementation, we showed that for any positive integers n and m with $\log n \le m \le 2^n$, there exists a minimal NFA M of n states such that minimal

NFAs for the complement of the language $L(M)$ have m states. However, the input alphabet size grows exponentially with n. We conjecture that the input alphabet could be decreased at least to linear size.

Further investigations may concern the deterministic concatenation and the nondeterministic complementation of finite languages.

Acknowledgements

We would like to thank Professor Geffert for proposing the problem on the nondeterministic complementation and for our helpful discussions on the topic.

References

1. J. C. Birget, Intersection and union of regular languages and state complexity, *Inform. Process. Lett.* **43** (1992) 185–190.
2. J. C. Birget, Partial orders on words, minimal elements of regular languages, and state complexity, *Theoret. Comput. Sci.* **119** (1993) 267–291.
3. C. Câmpeanu, K. Culik II, K. Salomaa, and S. Yu, State complexity of basic operations on finite languages, in: O. Boldt, H. Jürgensen (Eds.), *Proc. 4th International Workshop on Implementing Automata (WIA'99),* LNCS 2214, Springer-Verlag, Heidelberg, 2001, pp. 60–70.
4. C. Câmpeanu, K. Salomaa, and S. Yu, Tight lower bound for the state complexity of shuffle of regular languages, *J. Autom. Lang. Comb.* **7** (2002) 303–310.
5. M. Chrobak, Finite automata and unary languages, *Theoret. Comput. Sci.* **47** (1986) 149–158.
6. M. Domaratzki, State complexity and proportional removals, *J. Autom. Lang. Comb.* **7** (2002) 455–468.
7. I. Glaister, J. Shallit, A lower bound technique for the size of nondeterministic finite automata, *Inform. Process. Lett.* **59** (1996) 75–77.
8. M. Holzer, K. Salomaa, and S. Yu, On the state complexity of k-entry deterministic finite automata, *J. Autom. Lang. Comb.* **6** (2001) 453–466.
9. M. Holzer, M. Kutrib, State complexity of basic operations on nondeterministic finite automata, in: J.M. Champarnaud, D. Maurel (Eds.), *Implementation and Application of Automata (CIAA 2002),* LNCS 2608, Springer-Verlag, Heidelberg, 2003, pp. 148–157.
10. M. Holzer, M. Kutrib, Unary language operations and their nondeterministic state complexity, in: M. Ito, M. Toyama (Eds.), *Developments in Language Theory (DLT 2002),* LNCS 2450, Springer-Verlag, Heidelberg, 2003, pp. 162–172.
11. M. Holzer, M. Kutrib, Nondeterministic descriptional complexity of regular languages, *Internat. J. Found. Comput. Sci.* **14** (2003) 1087-1102.
12. J. Hromkovič, *Communication Complexity and Parallel Computing,* Springer-Verlag, Berlin, Heidelberg, 1997.
13. J. Hromkovič, Descriptional complexity of finite automata: concepts and open problems, *J. Autom. Lang. Comb.* **7** (2002) 519–531.
14. J. Hromkovič, S. Seibert, J. Karhumäki, H. Klauck, and G. Schnitger, Communication complexity method for measuring nondeterminism in finite automata, *Inform. and Comput.* **172** (2002) 202–217.
15. G. Jirásková, Note on minimal automata and uniform communication protocols, in:

C. Martin-Vide, V. Mitrana (Eds.), *Grammars and Automata for String Processing: From Mathematics and Computer Science to Biology, and Back,* Taylor and Francis, London, 2003, pp. 163–170.

16. G. Jirásková, State complexity of some operations on regular languages, in: E. Csuhaj-Varjú, C. Kintala, D. Wotschke, Gy. Vaszil (Eds.), *Proc. 5th Workshop Descriptional Complexity of Formal Systems,* MTA SZTAKI, Budapest, 2003, pp. 114–125.
17. E. Leiss, Succinct representation of regular languages by boolean automata, *Theoret. Comput. Sci.* **13** (1981) 323–330.
18. O. B. Lupanov, A comparison of two types of finite automata, *Problemy Kibernetiki,* **9** (1963) 321-326 (in Russian).
19. F. R. Moore, On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata, *IEEE Trans. Comput.* **20** (1971) 1211–1214.
20. G. Pighizzini, Unary language concatenation and its state complexity, in: S. Yu, A. Pun (Eds.), *Implementation and Application of Automata: 5th International Conference, CIAA 2000,* LNCS 2088, Springer-Verlag, 2001, pp. 252–262.
21. G. Pighizzini, J. Shallit, Unary language operations, state complexity and Jacobsthal's function, *Internat. J. Found. Comput. Sci.* **13** (2002) 145–159.
22. M. Rabin, D. Scott, Finite automata and their decision problems, *IBM Res. Develop.* **3** (1959) 114–129.
23. W. J. Sakoda, M. Sipser, Nondeterminism and the size of two-way finite automata, in: *Proc. 10th Annual ACM Symp. on Theory of Computing,* 1978, pp. 275–286.
24. M. Sipser, *Introduction to the theory of computation,* PWS, Boston, 1997.
25. S. Yu, Q. Zhuang, and K. Salomaa, The state complexity of some basic operations on regular languages, *Theoret. Comput. Sci.* **125** (1994) 315–328.
26. S. Yu, Chapter 2: Regular languages, in: G. Rozenberg, A. Salomaa, (Eds.), *Handbook of Formal Languages - Vol. I,* Springer-Verlag, Berlin, New York, pp. 41–110.
27. S. Yu, A renaissance of automata theory? *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS* **72** (2000) 270–272.
28. S. Yu, State complexity of finite and infinite regular languages, *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS* **76** (2000) 270–272.
29. S. Yu, State complexity of regular languages, *J. Autom. Lang. Comb.* **6** (2001) 221–234.