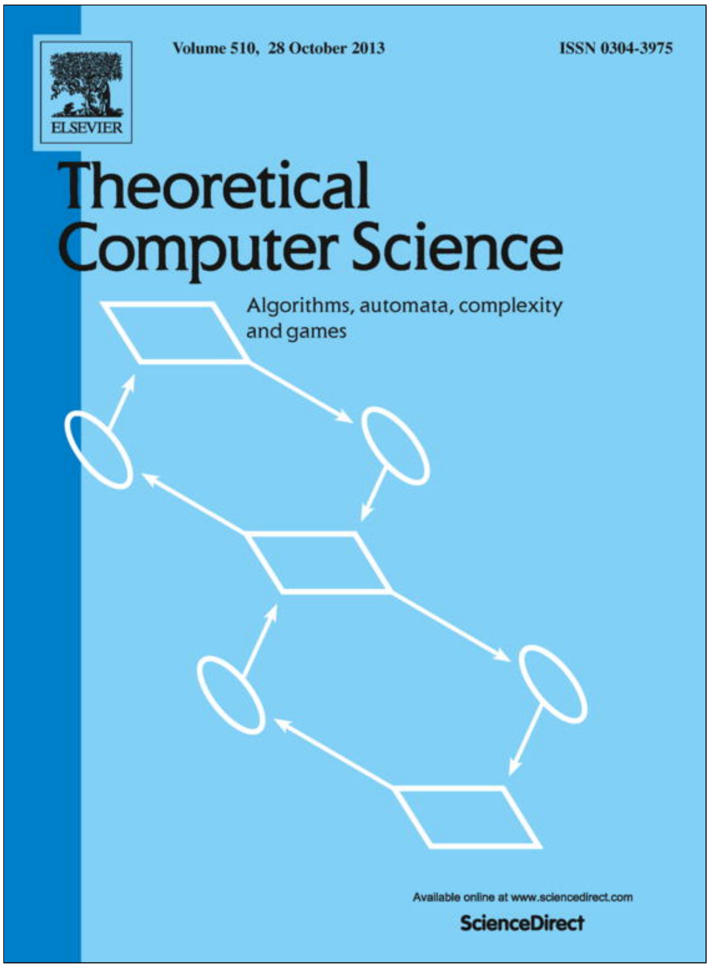
ELSEVIER
Volume 510, 28 October 2013
ISSN 0304-3975
Theoretical Computer Science
Algorithms, automata, complexity and games
Available online at www.sciencedirect.com
ScienceDirect

Theoretical Computer Science 510 (2013) 87–93

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

State complexity of combined operations for suffix-free regular languages

Hae-Sung Eom, Yo-Sub Han *

Department of Computer Science, Yonsei University, 50, Yonsei-Ro, Seodaemun-Gu, Seoul 120-749, Republic of Korea

ARTICLE INFO

Article history:
Received 25 August 2011
Received in revised form 1 July 2013
Accepted 15 September 2013
Communicated by M. Ito

Keywords:
State complexity
Suffix-free regular languages
Combined operations

ABSTRACT

We investigate the state complexity of combined operations for suffix-free regular languages. Suffix-free deterministic finite-state automata have a unique structural property that is crucial for obtaining the precise state complexity of basic operations. Based on the same property, we establish the state complexity of four combined operations: star-of-union, star-of-intersection, star-of-reversal and star-of-catenation. In the case of star-of-intersection, we only have an upper bound and the lower bound is open.

1. Introduction

Given a regular language L, researchers often use the number of states in the minimal DFA (deterministic finite-state automaton) for L to represent the complexity of L. Based on this notation, we define the state complexity of an operation for regular languages to be the number of states that are necessary and sufficient in the worst case for the minimal DFA that accepts the language obtained from the operation. Maslov [15] obtained the state complexity of catenation and later Yu et al. [23] investigated the state complexity further. The state complexity of an operation is calculated based on the structural properties of input regular languages and a given operation. Recently, due to large amount of memory, fast CPUs and massive data size, many applications using regular languages require finite-state automata (FAs) of very large size. This makes the estimated upper bound of the state complexity useful in practice since it helps to manage resources efficiently. Moreover, it is a challenging quest to verify whether or not an estimated upper bound can be reached.

Yu [22] gave a comprehensive survey of the state complexity of regular languages. Salomaa et al. [19] studied classes of languages for which the reversal operation reaches the exponential upper bound. As special cases of the state complexity, researchers examined the state complexity of finite languages [1,7], the state complexity of unary language operations [17] and the nondeterministic descriptional complexity of regular languages [10]. For regular language codes, Han et al. [9] studied the state complexity of prefix-free regular languages. Similarly, based on suffix-freeness, Han and Salomaa [8] looked at the state complexity of suffix-free regular languages. There are several other results with respect to the state complexity of different operations [2,4,5,11,12,16].

While people mainly looked at the state complexity of single operations (union, intersection, catenation and so on), Yu and his co-authors [6,18,20] recently started investigating the state complexity of combined operations (star-of-union, star-of-intersection and so on). They showed that the state complexity of a combined operation is usually not equal to the composition of the state complexities of the participating individual operations. They also observed that in a few cases, the

* Corresponding author.
E-mail addresses: haesung@cs.yonsei.ac.kr (H.-S. Eom), emmous@cs.yonsei.ac.kr (Y.-S. Han).

0304-3975/$ – see front matter
http://dx.doi.org/10.1016/j.tcs.2013.09.014

state complexity of a combined operation is very close to the composition of the state complexities. Lately, Jirásková and Okhotin [13] established the tight bound for star-of-intersection and star-of-union of regular languages.

We consider the state complexity of combined operations on suffix-free regular languages, in particular, combined operations involving Kleene star. Note that the state complexity of operations on suffix-free regular languages is very different from the state complexity of arbitrary regular languages because suffix-freeness gives a structural property in a suffix-free DFA; there always exists a sink state and the start state has no in-transitions. Han and Salomaa [8] noticed that the second property is necessary but not sufficient to guarantee that the language is suffix-free and examined the state complexity of suffix-free regular languages. We further investigate the state complexity of combined operations for suffix-free regular languages.

In Section 2, we define some basic notions. Then we present the state complexities of four combined operations in the following sections: We establish the tight bound for star-of-union, star-of-reversal and star-of-catenation. In the case of star-of-intersection, we only have an upper bound. We compare the state complexity of basic operations and the state complexity of combined operations for suffix-free regular languages, and conclude the paper in Section 7.

2. Preliminaries

Let Σ denote a finite alphabet of characters and Σ^* denote the set of all strings over Σ. The size $|\Sigma|$ of Σ is the number of characters in Σ. A language over Σ is any subset of Σ^*. The symbol $\emptyset$ denotes the empty language and the symbol λ denotes the null string. For strings x, y and z, we say that y is a *suffix* of z if $z = xy$. We define a language L to be suffix-free if for any two distinct strings x and y in L, x is not a suffix of y. For a string x, let x^R be the reversal of x and for a language L we denote $L^R = \{x^R \mid x \in L\}$.

A DFA A is specified by a tuple $(Q, \Sigma, \delta, s, F)$, where Q is a finite set of states, Σ is an input alphabet, $\delta : Q \times \Sigma \to Q$ is a transition function, $s \in Q$ is the start state and $F \subseteq Q$ is a set of final states. If F consists of a single state f, then we use f instead of $\{f\}$ for simplicity. Given a DFA A, we assume that A is complete; namely, each state has $|\Sigma|$ out-transitions and, therefore, A may have a sink state that is, a state, from which no string is accepted. We assume that A has a unique sink state since all sink states are equivalent and can be merged into a single state. Let $|Q|$ be the number of states in Q. The size $|A|$ of A is $|Q|$. For a transition $\delta(p, a) = q$ in A, we say that p has an *out-transition* and q has an *in-transition*. We say that A is *non-returning* if the start state of A does not have any in-transitions. An NFA is specified by a tuple $(Q, \Sigma, \delta, s, F)$, where Q, Σ, s and F have the same meaning as for a DFA and δ is a transition function for $Q \times \Sigma$ to 2^Q, which is the set of all subsets of Q. The subset construction produces an equivalent DFA $M' = (2^Q, \Sigma, \delta', s, F')$ for an NFA $M = (Q, \Sigma, \delta, s, F)$, where $d'(R, a) = \bigcup_{r \in R} \delta(r, a)$ and $F' = \{R \in 2^Q \mid R \cap F \neq \emptyset\}$. Note that the subset DFA M' is not necessarily minimal since some of its states may be unreachable or equivalent.

A string x over Σ is accepted by A if there is a labeled path from s to a final state such that this path reads x. We call this path an *accepting path*. Then the language $L(A)$ of A is the set of all strings spelled out by accepting paths in A. We say that A is suffix-free if $L(A)$ is suffix-free. If A is suffix-free, the start state of A cannot be final state, except for special case $L = \{\lambda\}$. We define a state q of A to be *reachable* if there is a path from the start state to q. In the following, we assume that all states are reachable and a DFA has at most one sink state. The state complexity $\mathcal{SC}(L)$ of a regular language L is defined to be the size of the minimal DFA recognizing L. The state complexity of an operation is the number of states that is sufficient and necessary in the worst case for a DFA to accept the language resulting from the operation, taken as a function of the state complexities of operands. Formally, if $\odot$ is a binary regular operation, then its state complexity is given by a function $f : N \times N \to N$ defined as follows:

$$f(m, n) = \max\big\{\mathcal{SC}(K \odot L) \,\big|\, \mathcal{SC}(K) = m \text{ and } \mathcal{SC}(L) = n\big\}.$$

For a unary operation, the definition is analogous.

For complete background knowledge in automata theory, the reader may refer to Wood [21].

We describe two known results that are useful to tackle the state complexity problem for suffix-free regular languages.

Proposition 1. *Let L be accepted by a non-returning NFA of n states. Then the minimal DFA for L^* has at most $2^{n-1} + 1$ states.*

Proof. Let $A = (Q, \Sigma, \delta, s, F)$ be a non-returning NFA for L. Then we construct an NFA $A' = (Q, \Sigma, \delta', s, F')$ for L^* from A as follows:

$$\delta'(q, a) = \begin{cases} \delta(q, a), & \text{if } q \notin F; \\ \delta(q, a) \cup \delta(s, a), & \text{if } q \in F. \end{cases}$$

$$F' = F \cup \{s\}.$$

Namely, we make s to be final and add all out-transitions of s to the set of out-transitions of each final state. The resulting NFA A' is still non-returning and has n states. Therefore, the subset construction for A' gives rise to a DFA of at most $2^{n-1} + 1$ reachable states. □

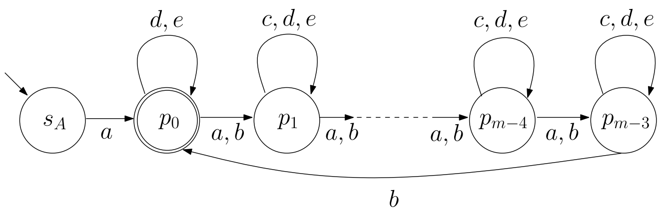

Fig. 1. DFA A for star-of-union. The sink state d_A is omitted.

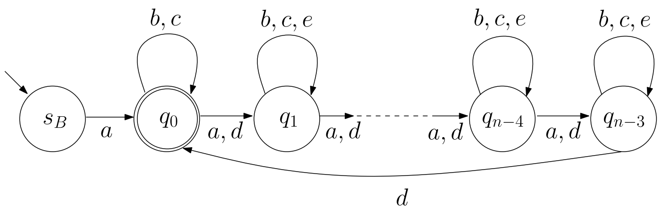

Fig. 2. DFA B for star-of-union. The sink state d_B is omitted.

Lemma 2. *(See Cmorik and Jirásková [3].) Consider a non-returning DFA A with the sink state and the sole final state. If no two distinct states of A go to a non-sink state by the same symbol, then A is suffix-free.*

Lemma 3. *Let NFA N and let D be the DFA obtained from N by the subset construction. Assume that for each state q of the NFA N, there exists a string w_q in Σ^* such that w_q is accepted by N from the state q and is not accepted by N from any other state. Then the states of the DFA D are pairwise distinguishable.*

Proof. Recall that a state of D is a subset of states of N. Two distinct states of the DFA D have to differ in a state q of the NFA N, and therefore the string w_q distinguishes the two states. □

3. Star-of-union

We first consider the state complexity of $(K \cup L)^*$ for suffix-free regular languages K and L.

Theorem 4. *Let Σ be an alphabet with $|\Sigma| \geqslant 5$. The state complexity of the star-of-union on suffix-free regular languages over Σ is given by the function*

$$f(m,n) = \begin{cases} 2, & \text{if } m = n = 1; \\ 2^{n-2}+1, & \text{if } m \in \{1,2\} \text{ and } n \geqslant 2; \\ 2^{m-2}+1, & \text{if } m \geqslant 2 \text{ and } n \in \{1,2\}; \\ 2^{m+n-4}+1, & \text{if } m,n \geqslant 3. \end{cases}$$

Proof. First we consider small cases of m and n. Note that $\mathcal{SC}(K) = 1$ means $K = \emptyset$. This follows that, for $m = n = 1$, $(K \cup L)^* = \emptyset^* = \{\lambda\}$, and, thus the state complexity is 2.

Next we examine the case where one of the DFAs has one or two states, say $m = 1$ or 2. This implies that $K = \emptyset$ (when $m = 1$) or $K = \{\lambda\}$ (when $m = 2$). Therefore, $(K \cup L)^* = L^*$ and $\mathcal{SC}((K \cup L)^*) = \mathcal{SC}(L^*) = 2^{n-2} + 1$, since, by Han and Salomaa [8], the state complexity of the star of an n-state suffix-free language is at most $2^{n-2} + 1$, and the bound is tight. The case of $n = 1$ or 2 is symmetric.

Let $m, n \geqslant 3$. Let $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ be a DFA for K and $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ be a DFA for L, where $|Q_A| = m$ and $|Q_B| = n$. Let $d_A \in Q_A$ and $d_B \in Q_B$ be the corresponding sink states of A and B, respectively. From A and B, we obtain a non-returning NFA N for $K \cup L$ by removing d_A, d_B and their in-transitions, and merging s_A and s_B into a new start state. Then N has $(m-2)+(n-2)+1$ states and, thus, the minimal DFA for $(K \cup L)^*$ has at most $2^{m+n-4} + 1$ states by Proposition 1.

To prove the tightness, consider the languages K and L accepted by the DFAs A and B shown in Fig. 1 and Fig. 2, respectively.

The languages accepted by the DFAs A and B are suffix-free by Lemma 2. Let $P = \{p_0, p_1, \ldots, p_{m-3}\}$ and $Q = \{q_0, q_1, \ldots, q_{n-3}\}$. To get an NFA N for $(K \cup L)^*$, we merge s_A and s_B into the new start state s that goes to $\{p_0, q_0\}$ by a. Then, we add transitions on a from p_0 to $\{p_0, q_0\}$ and from q_0 to $\{p_0, q_0\}$. The final states of the NFA N are s, p_0 and q_0. The aim is to prove that the DFA D obtained from the NFA N by the subset construction has $2^{m+n-4} + 1$ reachable and pairwise distinguishable states.

Let us prove that $\{s\}$ and an arbitrary subset X of $P \cup Q$ are reachable in the DFA D. The subset $\{s\}$ is the start state of the DFA D and it goes to $\{p_0, q_0\}$ by a. Then, the state $\{p_0, q_0\}$ goes to the state $P \cup Q$ by $a^{\max\{m,n\}}$. Now let X be a subset of $P \cup Q$ and $p_i \in X$. Then the set $X \setminus \{p_i\}$ is reached from the set X by the string $b^{m-2-i}cb^i$. Symmetrically, the set $X \setminus \{q_j\}$ is reached from the set X by the string $d^{n-2-j}ed^j$. This proves the reachability of all the subsets of $P \cup Q$ by induction.

In the NFA N, the string e is accepted only from state p_0, the string $b^{m-2-i}e$ is accepted only from state p_i, and symmetrically, c is accepted only from q_0, and $d^{n-2-j}c$ is accepted only from q_j. It follows that the subsets of $P \cup Q$ are pairwise distinguishable by Lemma 3. The start and final state $\{s\}$ is distinguished from any other final state either by e or by c.

Therefore, $\{s\}$ and all subsets of $P \cup Q$ are reachable and pairwise inequivalent. $\square$

4. Star-of-intersection

We consider the state complexity of $(K \cap L)^*$ for suffix-free regular languages K and L. We first examine special cases that K is small and after that investigate the general case.

4.1. Special cases: $m = 2, 3$

We consider when one suffix-free DFA has only two states.

Lemma 5. *Given a 2-state suffix-free minimal DFA A and an n-state suffix-free minimal DFA B, $\mathcal{SC}((L(A) \cap L(B))^*) = 2$.*

Proof. Since A has two states, $L(A) = \{\lambda\}$ and thus $L(A) \cap L(B)$ is either $L(A)$ or $\emptyset$. It follows that $\mathcal{SC}((L(A) \cap L(B))^*) = \mathcal{SC}(\{\lambda\}) = 2$. $\square$

We next consider when one suffix-free DFA has only three states.

Lemma 6. *Given a 3-state suffix-free minimal DFA A and an n-state suffix-free minimal DFA B, $\mathcal{SC}((L(A) \cap L(B))^*) \leqslant n$, and the bound is tight if $|\Sigma| \geqslant 2$.*

Proof. If K is suffix-free regular language over Σ with $\mathcal{SC}(K) = 3$, then

$$K = ST^*,$$

where S and T are disjoint subalphabets of Σ. Since the state complexity of intersection for suffix-free regular languages is $mn - 2(m+n) + 6$, the suffix-free regular language $K \cap L$ has at most n states ($= mn - 2(m+n) + 6$ if $m = 3$). Let the language be accepted by a DFA C with the start state s and the transition function δ. Notice that all the transitions on symbols in S from any non-start state of C go to the sink state. To get the DFA for $(K \cap L)^*$ from the DFA C we only need to redirect the transitions on every symbol $x \in S$ from every final state of C to the state $\delta(s, x)$.

We next show that the upper bound can be reached. Let A be the minimal DFA for $L(A) = \{\#a^n \mid n \geqslant 0\}$ and B be the minimal DFA for $L(B) = \{\#a^i \mid i \equiv 0 \pmod{n-2}\}$ over $\Sigma = \{a, \#\}$. By Lemma 2, $L(A)$ and $L(B)$ are suffix-free. It is easy to verify that $|A| = 3$ and $|B| = n$. Let $L = (L(A) \cap L(B))^*$. We claim that the minimal DFA for L needs n states by presenting a set R of n strings over Σ that are pairwise inequivalent modulo the right-invariant congruence of L. Let $R = R_1 \cup R_2$, where

$$R_1 = \{\lambda, \#a^{n-3}\#\} \quad \text{and} \quad R_2 = \{\#a^i \mid 0 \leqslant i \leqslant n-3\}.$$

Any string $\#a^i$ from R_2 is inequivalent with λ since $\#a^i \cdot \# \notin L$ but $\lambda \cdot \# \in L$. Similarly, $\#a^i$ is inequivalent with $\#a^{n-3}\#$ since $\#a^i \cdot a^{n-2-i} \in L$ but $\#a^{n-3}\# \cdot a^{n-2-i} \notin L$, for $0 \leqslant i \leqslant n-3$. Thus R_1 and R_2 are inequivalent with each other. The two strings λ and $\#a^{n-3}\#$ of R_1 are inequivalent as well. Next, consider two distinct strings $\#a^i$ and $\#a^j$ from R_2. Since $\#a^i \neq \#a^j$, $\#a^i \cdot a^{n-2-i} \in L$ but $\#a^j \cdot a^{n-2-i} \notin L$. Therefore, any two distinct strings from R_2 are inequivalent. Thus, all n strings in R are pairwise inequivalent. $\square$

4.2. General case

Lemma 7. *Given an m-state suffix-free minimal DFA A for K and an n-state suffix-free minimal DFA B for L, $2^{(m-2)(n-2)} + 1$ states are sufficient for a DFA to accept $(K \cap L)^*$, where $m, n \geqslant 4$.*

Proof. Let $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ and $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$, where $|Q_A| = m$, $|Q_B| = n$. Let $d_A \in Q_A$ and $d_B \in Q_B$ be the corresponding sink states of A and B. From A and B, we can obtain a non-returning NFA N for $K \cap L$ by removing d_A, d_B and running the cross-product construction. Recall that the states (s_A, q) with $q \neq s_B$ and the states (p, s_B) with $p \neq s_A$ are unreachable in the cross-product automaton. Then, we eliminate unreachable states from N and the resulting NFA has $(m-2)(n-2)+1$ states. Thus, the minimal DFA for $(K \cap L)^*$ has at most $2^{(m-2)(n-2)} + 1$ states by Proposition 1. $\square$

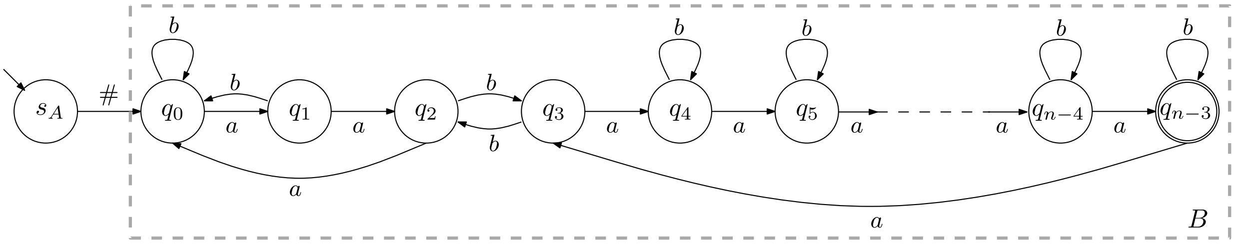

Fig. 3. DFA A for star-of-reversal. The sink state d_A is omitted. Note that B within the dotted box is an example DFA over $\{a,b\}$ such that $\mathcal{SC}(L(B)^R) = 2^{n-2}$ in Jirásková and Šebej [14], where q_0 is the start state and q_{n-3} is the final state.

It is open to find the exact lower bound examples. On the other hand, we notice that the star-of-intersection operation gives rise to an exponential blowup. For instance, when $L(A) = L(B)$, $L(A \cap B)^* = L(A)^*$. Since Han and Salomaa [8] showed that there is a suffix-free regular language L such that $\mathcal{SC}(L^*) = 2^{m-2} + 1$, there exists a suffix-free regular language whose star-of-intersection state complexity is $2^{m-2} + 1$.

5. Star-of-reversal

We consider the state complexity of $(L^R)^*$ for a suffix-free regular language L.

Theorem 8. *Let Σ be an alphabet with $|\Sigma| \geqslant 3$. The state complexity of the star-of-reversal on suffix-free regular languages over Σ is given by the function*

$$f(n) = \begin{cases} 2, & \text{if } n \in \{1,2,3\}; \\ 2^{n-2} + 1, & \text{if } n \geqslant 4. \end{cases}$$

Proof. First we consider small cases of n. We tackle three cases separately:

(1) For $n = 1$, $L(A) = \emptyset$ and $(L(A)^R)^* = \{\lambda\}$.
(2) For $n = 2$, $L(A) = \{\lambda\}$ and $(L(A)^R)^* = \{\lambda\}$.
(3) For $n = 3$, let $A = (Q_A, \Sigma, \delta_A, q_0, q_1)$ be the minimal DFA for a suffix-free regular language and $Q_1 = \{q_0, q_1, d_A\}$, where d_A is the sink state. We flip the transition directions in δ_A, make q_0 final and q_1 non-final. Furthermore, we use q_1 as a new start state. Namely, we construct a new FA $A^R = (Q_A, \Sigma, \delta_A^R, q_1, q_0)$. It is easy to verify that A^R is deterministic and $L(A^R) = L(A)^R$. The DFA $A^{R,*} = (Q_A, \Sigma, \delta_A^{R,*}, q_0, q_0)$ recognizes $L(A^R)^*$. Moreover q_1 is now unreachable from q_0 in $A^{R,*}$. This follows that two states (the start and final state, and the sink state) are sufficient for the minimal DFA $A^{R,*}$. For the necessary case, we use a suffix-free regular language $\{a\}$ over $\Sigma = \{a,b\}$ whose star-of-reversal minimal DFA has 2 states.

If L is suffix-free with $\mathcal{SC}(L) = n$, when $n \geqslant 4$, then L^R is prefix-free with $\mathcal{SC}(L^R) \leqslant 2^{n-2} + 1$ [8]. If K is prefix-free with $\mathcal{SC}(K) = n$, then $\mathcal{SC}(K^*) = n$ [9]. Combining these two results, we have $\mathcal{SC}((L^R)^*) \leqslant 2^{n-2} + 1$ for a suffix-free language L with $\mathcal{SC}(L) = n$.

Recently, Jirásková and Šebej [14] showed that there exists a binary DFA A such that $\mathcal{SC}(L(A)) = n$ and $\mathcal{SC}(L(A)^R) = 2^n$. We modify A to be suffix-free and demonstrate that the upper bound in Theorem 8 can be reached.

Let K be the language over $\{a,b\}$ accepted by the $(n-2)$-state Šebej's automaton (see Fig. 3). Then, K^R requires 2^{n-2} states. Take $L = \#K$. Then L is accepted by an n-state DFA, and $L^R = K^R\#$. Since K^R requires 2^{n-2} states, there exist strings $\{x_1, x_2, \ldots, x_{2^{n-2}}\}$ that are pairwise distinguishable in the right-invariant congruence defined by K^R. Consider the set of strings $\{x_1, x_2, \ldots, x_{2^{n-2}}\} \cup \{w\#\}$, where w is an arbitrary string in K^R. For arbitrary two symbols $i \neq j$, there is a string y such that exactly one of the strings $x_i y$ and $x_j y$ is in K^R. Then exactly one of the strings $x_i y\#$ and $x_j y\#$ is in L^R. The string $w\#$ is in L^R while the other strings in the above set are not. Thus the strings in $\{x_1, x_2, \ldots, x_{2^{n-2}}\} \cup \{w\#\}$ are pairwise distinguishable in the right-invariant congruence defined by L^R. $\square$

6. Star-of-catenation

We consider the state complexity of $(K \cdot L)^*$ for suffix-free regular languages K and L.

Theorem 9. *Let Σ be an alphabet with $|\Sigma| \geqslant 6$. The state complexity of the star-of-catenation on suffix-free regular languages over Σ is given by the function*

$$f(m,n) = \begin{cases} 2, & \text{if } m = 1 \text{ or } n = 1; \\ 2^{n-2} + 1, & \text{if } m = 2 \text{ and } n \geqslant 2; \\ 2^{m-2} + 1, & \text{if } m \geqslant 2 \text{ and } n = 2; \\ 2^{m+n-4} + 1, & \text{if } m, n \geqslant 3. \end{cases}$$

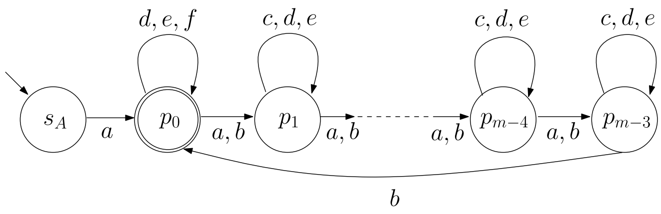

Fig. 4. DFA A for star-of-catenation. The sink state d_A is omitted.

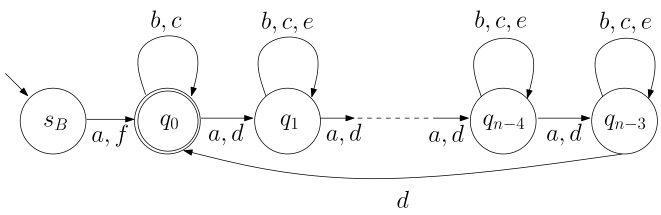

Fig. 5. DFA B for star-of-catenation. The sink state d_B is omitted.

Proof. First we consider small cases of m and n. Note that, for $\mathcal{SC}(K) = 1$, $K = \emptyset$. Therefore, when $m = 1$ or $n = 1$, $(K \cdot L)^* = \emptyset^* = \{\lambda\}$, and, thus the state complexity is 2.

Next we examine the case where one of the DFAs has two states, say $m = 2$. This implies that $K = \{\lambda\}$. Therefore, $(K \cdot L)^* = L^*$ and $\mathcal{SC}((K \cdot L)^*) = \mathcal{SC}(L^*) = 2^{n-2} + 1$, since, by [8], the state complexity of the star of an n-state suffix-free language is at most $2^{n-2} + 1$, and the bound is tight. The case of $n = 2$ is symmetric.

Let $m, n \geqslant 3$. From $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ be a DFA for K and $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ be a DFA for L, we can construct a non-returning NFA N for $K \cdot L$ as follows: First we make final states of A non-final and F_B is the new set of final states. Then, for $z \in \Sigma$, we add z-transition from every state in F_A to $\delta_B(s_B, z)$. Now s_A is the new start state and this makes s_B unreachable. Thus, we remove s_B and two sink states d_A, d_B of A and B. Since s_A still has no in-transitions in N, N is non-returning and has $1 + (m - 2) + (n - 2)$ states. Therefore, the minimal DFA for $(K \cdot L)^*$ has at most $2^{m+n-4} + 1$ states by Proposition 1.

To prove the tightness, consider the languages K and L accepted by the DFAs A and B shown in Fig. 4 and Fig. 5, respectively.

The languages accepted by the DFAs A and B are suffix-free by Lemma 2. Let $P = \{p_0, p_1, \ldots, p_{m-3}\}$ and $Q = \{q_0, q_1, \ldots, q_{n-3}\}$. To construct an NFA N for $(K \cdot L)^*$, first omit s_B and the transition by a from s_B to q_0. Next we add transitions on a from p_0 to q_0 and from q_0 to p_0. Next add transitions on f from p_0 to q_0. The start state of NFA N is s_A and final states of NFA N are s_A and q_0. The aim is to prove that the DFA D obtained from the NFA N by the subset construction has $2^{m+n-4} + 1$ reachable and pairwise distinguishable states.

Let us prove that $\{s_A\}$ and an arbitrary subset X of $P \cup Q$ are reachable in the DFA D. The subset $\{s_A\}$ is the start state of the DFA D and it goes to $\{p_0\}$ by a. The state $\{p_0\}$ goes to $\{p_0, q_0\}$ by f. Then, the state $\{p_0, q_0\}$ goes to the state $P \cup Q$ by $a^{\max\{m,n\}}$. Now let X be a subset of $P \cup Q$ and $p_i \in X$. Then the set $X \setminus p_i$ is reached from the set X by the string $b^{m-2-i}cb^i$. Symmetrically, the set $X \setminus q_j$ is reached from the set X by the string $d^{n-2-j}ed^j$. This proves the reachability of all the subsets of $P \cup Q$ by induction.

In the NFA N, the string e is accepted only from state p_0, the string $b^{m-2-i}e$ is accepted only from state p_i, and symmetrically, c is accepted only from q_0, and $d^{n-2-j}c$ is accepted only from q_j. It follows that the subsets of $P \cup Q$ are pairwise distinguishable by Lemma 3. The start and final state $\{s\}$ is distinguished from any other final state either by e or by c.

Therefore, all states in D are reachable and pairwise inequivalent. $\square$

7. Conclusions

We can usually obtain a much lower state complexity for combined operations compared with the compositions of state complexities of individual operations. However, for some cases, the state complexity of combined operations and the composition of state complexities are similar. We have examined suffix-free regular languages and computed the state complexity of combined operations.

Fig. 6 summarizes the state complexity of basic operations and the state complexity for combined operations of suffix-free regular languages. For star-of-union, star-of-reversal and star-of-catenation, we have much lower state complexity compared to the composition of the state complexities.

The lower bound for star-of-intersection is open. We notice that the upper bound of the star-of-intersection for two suffix-free regular languages is the same as the composite function of intersection $(mn - 2(m + n) + 6)$ and Kleene star $(2^{n-2} + 1)$. For regular languages, the state complexity for star-of-intersection is $\frac{3}{4} \cdot 2^{mn}$ [13,18], which is also the

Operation	Complexity	Operation	Complexity
$K \cup L$	$mn - (m+n) + 2$	$(K \cup L)^*$	$2^{m+n-4} + 1$
$K \cap L$	$mn - 2(m+n) + 6$	$(K \cap L)^*$	$\leqslant 2^{(m-2)(n-2)} + 1$
L^R	$2^{n-2} + 1$	$(L^R)^*$	$2^{n-2} + 1$
$K \cdot L$	$(m-1)2^{n-2} + 1$	$(K \cdot L)^*$	$2^{m+n-4} + 1$
L^*	$2^{n-2} + 1$	$(L^*)^* = L^*$	$2^{n-2} + 1$

Fig. 6. Comparison table between the state complexity of basic operations and the state complexity for combined operations of suffix-free regular languages.

same to the compositions of state complexity of intersection and star for regular languages. Therefore, it is our future work to look for a tight lower bound for the star-of-intersection for suffix-free regular languages.

Acknowledgements

We wish to thank the referees (and especially one of them who went way beyond the call of duty) for intuitive upper bound proofs using NFAs and recent references. Their comments are very helpful for improving the paper. We also appreciate the careful reading of the paper and many valuable suggestions.

This research was supported by the Basic Science Research Program through NRF funded by MEST (2010-0009168, 2012R1A1A2044562).

References

[1] C. Câmpeanu, K. Culik II, K. Salomaa, S. Yu, State complexity of basic operations on finite languages, in: Proceedings of WIA'99, in: Lecture Notes in Computer Science, vol. 2214, 2001, pp. 60–70.
[2] C. Câmpeanu, K. Salomaa, S. Yu, Tight lower bound for the state complexity of shuffle of regular languages, J. Autom. Lang. Comb. 7 (3) (2002) 303–310.
[3] R. Cmorik, G. Jirásková, Basic operations on binary suffix-free languages, in: Z. Kotásek, et al. (Eds.), MEMICS 2011, in: Lecture Notes in Computer Science, vol. 7119, Springer, Heidelberg, 2012, pp. 94–102.
[4] M. Domaratzki, State complexity of proportional removals, J. Autom. Lang. Comb. 7 (4) (2002) 455–468.
[5] M. Domaratzki, K. Salomaa, State complexity of shuffle on trajectories, J. Autom. Lang. Comb. 9 (2–3) (2004) 217–232.
[6] Y. Gao, K. Salomaa, S. Yu, The state complexity of two combined operations: Star of catenation and star of reversal, Fundam. Inform. 83 (1–2) (2008) 75–89.
[7] Y.-S. Han, K. Salomaa, State complexity of union and intersection of finite languages, Int. J. Found. Comput. Sci. 19 (3) (2008) 581–595.
[8] Y.-S. Han, K. Salomaa, State complexity of basic operations on suffix-free regular languages, Theor. Comput. Sci. 410 (27–29) (2009) 2537–2548.
[9] Y.-S. Han, K. Salomaa, D. Wood, Operational state complexity of prefix-free regular languages, in: Automata, Formal Languages, and Related Topics – Dedicated to Ferenc Gécseg on the Occasion of His 70th Birthday, 2009, pp. 99–115.
[10] M. Holzer, M. Kutrib, Nondeterministic descriptional complexity of regular languages, Int. J. Found. Comput. Sci. 14 (6) (2003) 1087–1102.
[11] M. Hricko, G. Jirásková, A. Szabari, Union and intersection of regular languages and descriptional complexity, in: Proceedings of DCFS'05, 2005, pp. 170–181.
[12] J. Jirásek, G. Jirásková, A. Szabari, State complexity of concatenation and complementation, Int. J. Found. Comput. Sci. 16 (3) (2005) 511–529.
[13] G. Jirásková, A. Okhotin, On the state complexity of star of union and star of intersection, Fundam. Inform. 109 (2) (2011) 161–178.
[14] G. Jirásková, J. Šebej, Reversal of binary regular languages, Theor. Comput. Sci. 449 (2012) 85–92.
[15] A. Maslov, Estimates of the number of states of finite automata, Sov. Math. Dokl. 11 (1970) 1373–1375.
[16] C. Nicaud, Average state complexity of operations on unary automata, in: Proceedings of MFCS'99, in: Lecture Notes in Computer Science, vol. 1672, 1999, pp. 231–240.
[17] G. Pighizzini, J. Shallit, Unary language operations, state complexity and Jacobsthal's function, Int. J. Found. Comput. Sci. 13 (1) (2002) 145–159.
[18] A. Salomaa, K. Salomaa, S. Yu, State complexity of combined operations, Theor. Comput. Sci. 383 (2–3) (2007) 140–152.
[19] A. Salomaa, D. Wood, S. Yu, On the state complexity of reversals of regular languages, Theor. Comput. Sci. 320 (2–3) (2004) 315–329.
[20] K. Salomaa, S. Yu, On the state complexity of combined operations and their estimation, Int. J. Found. Comput. Sci. 18 (2007) 683–698.
[21] D. Wood, Theory of Computation, John Wiley & Sons, Inc., New York, NY, 1987.
[22] S. Yu, State complexity of regular languages, J. Autom. Lang. Comb. 6 (2) (2001) 221–234.
[23] S. Yu, Q. Zhuang, K. Salomaa, The state complexities of some basic operations on regular languages, Theor. Comput. Sci. 125 (2) (1994) 315–328.