Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 383 (2007) 140–152
Theoretical Computer Science
www.elsevier.com/locate/tcs

State complexity of combined operations

Arto Salomaa[a], Kai Salomaa[b], Sheng Yu[c,*]

[a] Turku Centre for Computer Science, Lemminkäisenkatu 14, 20520 Turku, Finland
[b] School of Computing, Queen's University, Kingston, Ontario K7L 3N6, Canada
[c] Department of Computer Science, The University of Western Ontario, London, Ontario, N6A 5B7, Canada

Abstract

We study the state complexity of combined operations. Two particular combined operations are studied: star of union and star of intersection. It is shown that the state complexity of a combined operation is not necessarily similar to the combination of the individual state complexities of the participating operations.

Keywords: State complexity; Combined operations; Regular languages

1. Introduction

State complexity is a topic of many recent publications. State complexity is not only a fundamental topic in theoretical computer science, but also having important practical implications in automata applications [20]. There were a number of papers related to state complexity published in the past. However, since the publication of [21], a much larger number of papers in the area of state complexity have appeared. The following, for example, is a list of some of the papers published recently: [2–8,11–16,18–21]. Notice that in all those papers, state complexity is considered for only individual operations, e.g. union, intersection, catenation, and Kleene star. We know that, in practice, not only individual operations but also combinations of operations are often required to be performed on finite automata. The state complexity of combined operations should also be studied.

The state complexity of a combination of operations may not necessarily equal to the composition of the state complexities of the individual operations. For example, given an m-state DFA language L_1 and an n-state DFA language L_2, what is the state complexity of $(L_1 \cup L_2)^*$ (i.e. the number of states of a minimal DFA that accepts $(L_1 \cup L_2)^*$ in the worst case)? It is known that the state complexity of the union of an m-state DFA language and an n-state DFA language is mn, and the state complexity of the (Kleene) star of an n-state DFA language is $2^{n-1} + 2^{n-2}$. Then is it true that the state complexity of $(L_1 \cup L_2)^*$ would be $2^{mn-1} + 2^{mn-2}$? In this paper, we will show that it is not true for this combination of operations. The result is even in a different order. However, in some other cases, the state complexity of a combination of operations may be very similar to the composition of the state complexities of

* Corresponding author.
E-mail addresses: asalomaa@utu.fi (A. Salomaa), ksalomaa@cs.queensu.ca (K. Salomaa), syu@csd.uwo.ca (S. Yu).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2007.04.015

individual operations. We will show in this paper that $(L_1 \cap L_2)^*$, where L_1 and L_2 are regular languages, is in this category.

In the case of combination of two operations, the first operation may restrict its result to a special type of DFA. Then the worst cases for the second operation in the general setting may or may not be among the outputs of the first operation. Therefore, the state complexity of a combination of operations may or may not be the same as the composition of the state complexities of the individual operations. Each case has to be studied individually.

In this paper, we study only two different cases, i.e. $(L_1 \cup L_2)^*$ and $(L_1 \cap L_2)^*$ for two regular languages L_1 and L_2. We hope that results on the state complexity of other combined operations on regular languages will be obtained in the near future.

In the next section, we introduce the basic notations that are necessary for this paper and review the definition of state complexity. In Section 3, we study the state complexity of the combination of union and star. In Section 4, we study the state complexity of the combination of intersection and star. We conclude the paper in Section 5.

2. Preliminaries

A deterministic finite automaton (DFA) is denoted by a 5-tuple $A = (Q, \Sigma, \delta, q_0, F)$, where Q is the finite set of states, Σ is the finite input alphabet, $\delta : Q \times \Sigma \to Q$ is the state transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. A DFA is said to be complete if $\delta(q, a)$ is defined for all $q \in Q$ and $a \in \Sigma$.

In this paper, the state transition function δ is often extended to $\hat{\delta} : 2^Q \times \Sigma \to 2^Q$, where 2^Q denotes the power set of Q, i.e. the set of all subsets of Q. The function $\hat{\delta}$ is defined by $\hat{\delta}(R, a) = \{\delta(r, a) \mid r \in R\}$, for $R \subseteq Q$ and $a \in \Sigma$. We just write δ instead of $\hat{\delta}$ if there is no confusion.

For a rather complete background knowledge in automata theory, the reader may refer to [9,17,22].

State complexity is a descriptional complexity measure for regular languages based on the deterministic finite automaton model. The state complexity of a regular-language operation also gives a lower bound for the space as well as the time complexity of the same operation. In many cases, the bounds given are tight.

The *state complexity* of a regular language L, denoted $sc(L)$, is the number of states in the minimal complete DFA accepting L. The state complexity of a class $\mathcal{L}$ of regular languages, denoted $sc(\mathcal{L})$, is the supremum among all $sc(L)$, $L \in \mathcal{L}$. When we speak about the state complexity of an operation on regular languages, we mean the state complexity of the languages resulting from the operation. For example, we say that the state complexity of the intersection of an m-state DFA language, i.e. a language accepted by an m-state complete DFA, and an n-state DFA language is exactly mn. This means that mn is the state complexity of the class of languages each of which is the intersection of an m-state DFA language and an n-state DFA language. In other words, there exist two regular languages that are accepted by an m-state DFA and an n-state DFA, respectively, such that the intersection of them is accepted by a minimal DFA of mn states, and this is the worst case. So, in a certain sense, state complexity is a worst-case complexity.

3. Star of union

We first consider the state complexity of the star-of-union combined operation, i.e. the combination that includes first the union of two regular languages and then the star of the resulting language from the union. In this section, we prove an upper bound first, and then provide examples that can reach the bound.

3.1. An upper bound

Let L_i be recognized by a DFA of size m_i, $i = 1, 2$. From the state complexities of the individual operations of union and star [21,22] we know that

$$\mathrm{sc}((L_1 \cup L_2)^*) \le 2^{m_1 m_2 - 1} + 2^{m_1 m_2 - 2}.$$

Here we show that the state complexity of $(L_1 \cup L_2)^*$, in fact, is always considerably less than the above bound obtained directly from state complexities of the individual operations.

Let $L_i = L(A_i)$ where A_i is a *complete* DFA $A_i = (Q_i, \Sigma, \delta_i, q_{0,i}, F_i)$, $i = 1, 2$. Note that we assume that Q_1 and Q_2 are disjoint. We construct a DFA

$$A = (Q, \Sigma, \delta, q_0, F) \tag{1}$$

for the language $(L_1 \cup L_2)^*$. As the state set we choose

$$Q = \{q_0\} \cup \mathcal{P} \cup \mathcal{R} \tag{2}$$

where

$$\mathcal{P} = \{P_1 \cup P_2 \mid \emptyset \neq P_i \subseteq Q_i - F_i,\ i = 1, 2\},$$

and

$$\mathcal{R} = \{R \subseteq Q_1 \cup Q_2 \mid q_{0,1}, q_{0,2} \in R,\ R \cap (F_1 \cup F_2) \neq \emptyset\}.$$

If $q_{0,1} \notin F_1$ and $q_{0,2} \notin F_2$, the initial state q_0 is a new symbol. (In this case the empty word is not in $L_1 \cup L_2$.) If $q_{0,1} \in F_1$ or $q_{0,2} \in F_2$, we choose $q_0 = \{q_{0,1}, q_{0,2}\}$ in which case q_0 is an element of $\mathcal{R}$.

The set of accepting states F is chosen to be $\mathcal{R} \cup \{q_0\}$.

The transition function δ is defined as follows:

(i) For all $b \in \Sigma$, $\delta(q_0, b) = \begin{cases} \{\delta_1(q_{0,1}, b), \delta_2(q_{0,2}, b)\} \text{ if } \delta_i(q_{0,i}, b) \cap F_i = \emptyset, i = 1, 2, \\ \{\delta_1(q_{0,1}, b), \delta_2(q_{0,2}, b), q_{0,1}, q_{0,2}\} \text{ otherwise.} \end{cases}$

(ii) Let $R_i \subseteq Q_i$, $i = 1, 2$, be such that $R_1 \cup R_2 \in Q$. For all $b \in \Sigma$ we define

$$\delta(R_1 \cup R_2, b) = \begin{cases} \delta_1(R_1, b) \cup \delta_2(R_2, b) \text{ if } \delta_i(R_i, b) \cap F_i = \emptyset, i = 1, 2, \\ \delta_1(R_1, b) \cup \delta_2(R_2, b) \cup \{q_{0,1}, q_{0,2}\} \text{ otherwise.} \end{cases} \tag{3}$$

Note that if $q_0 = \{q_{0,1}, q_{0,2}\}$, the transitions defined for q_0 in (i) and (ii) coincide.

The computation of A begins by simulating both the computation of A_1 and A_2. Always when one of A_1 or A_2 enters a final state, A should also enter both start states $q_{0,1}$ and $q_{0,2}$. It is easy to verify that $L(A) = (L(A_1) \cup L(A_2))^*$.

In order to obtain an upper bound for the operation "star of union" we count the number of states of A. Denote $|Q_i| = m_i$ and $|F_i| = k_i$, $i = 1, 2$. In the following we assume that:

$$1 \le k_i < m_i, \quad i = 1, 2. \tag{4}$$

Note that (4) holds in all but the trivial cases where $L(A_i)$ is Σ^* or $\emptyset$.

The cardinality of $\mathcal{P}$ is $(2^{m_1-k_1} - 1)(2^{m_2-k_2} - 1)$. Thus we obtain

$$|Q| = (2^{m_1-k_1} - 1)(2^{m_2-k_2} - 1) + X + Y, \tag{5}$$

where X is the cardinality of $\mathcal{R}$ and has the value

$$X = \begin{cases} 2^{m_1+m_2-2} \text{ if } q_{0,1} \in F_1 \text{ or } q_{0,2} \in F_2, \\ 2^{m_1+m_2-2} - 2^{m_1-k_1-1+m_2-k_2-1} \text{ if } q_{0,1} \notin F_1 \text{ and } q_{0,2} \notin F_2. \end{cases}$$

Above, in the case where $q_{0,1} \notin F_1$ and $q_{0,2} \notin F_2$, $2^{m_1-k_1-1+m_2-k_2-1}$ is the number of all sets $Z \subseteq (Q_1 \cup Q_2) - \{q_{0,1}, q_{0,2}\}$ that do not contain any states of $F_1 \cup F_2$, and hence the corresponding set $\{q_{0,1}, q_{0,2}\} \cup Z$ is not in $\mathcal{R}$. Note that $\mathcal{P}$ and $\mathcal{R}$ are always disjoint and in this case $\{q_{0,1}, q_{0,2}\} \cup Z$ is in $\mathcal{P}$.

Also we observe that $\{q_0\}$ is part of $\mathcal{R}$ except in the case when $q_{0,1} \notin F_1$ and $q_{0,2} \notin F_2$. The value of the term Y in the sum (5) is

$$Y = \begin{cases} 1 \text{ if } q_{0,1} \notin F_1 \text{ and } q_{0,2} \notin F_2, \\ 0 \text{ otherwise.} \end{cases}$$

Since $1 \le k_i < m_i$, $i = 1, 2$, the upper-bound (5) reaches the worst case

$$(2^{m_1-1} - 1)(2^{m_2-1} - 1) + 2^{m_1+m_2-2} = 2^{m_1+m_2-1} - 2^{m_1-1} - 2^{m_2-1} + 1 \tag{6}$$

when A_1 and A_2 both have one final state and at least one of the start states of A_1 and A_2 is a final state. Therefore, we have the following theorem:

Theorem 3.1. *Let $L_i = L(A_i)$ and A_i be a complete DFA of m_i states $i = 1, 2$. Then $(L_1 \cup L_2)^*$ is accepted by a complete DFA of no more than $2^{m_1+m_2-1} - 2^{m_1-1} - 2^{m_2-1} + 1$ states.*

The direct composition of state complexities for the star of union of an m_1-state DFA language and an m_2-state DFA language (see [21]) gives a bound $2^{m_1 m_2 - 1} + 2^{m_1 m_2 - 2}$ which is considerably worse than (6).

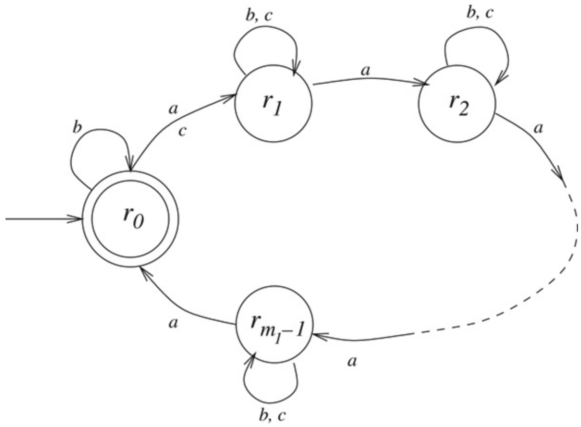

Fig. 1. DFA A_1.

3.2. Worst-case example

3.2.1. Case $m_1 \geq 3$ and $m_2 \geq 3$

We construct DFAs of size $m_i \geq 3$, $i = 1, 2$, over the alphabet $\Sigma = \{a, b, c\}$ such that the state complexity of $(L(A_1) \cup L(A_2))^*$ reaches the upper-bound (6). In the next two subsections we explain how the construction can be modified for the case where m_1 or m_2 is less than three.

We choose

$$A_1 = (R, \Sigma, \delta_1, r_0, \{r_0\}) \text{ where } R = \{r_0, r_1, \ldots, r_{m_1-1}\}, \tag{7}$$

- $\delta_1(r_j, a) = r_{(j+1) \pmod{m_1}}$, $j = 0, 1, \ldots, m_1 - 1$,
- $\delta_1(r_j, b) = r_j$, $j = 0, 1, \ldots, m_1 - 1$,
- $\delta_1(r_0, c) = r_1$, $\delta_1(r_j, c) = r_j$, $j = 1, 2, \ldots, m_1 - 1$,

and,

$$A_2 = (P, \Sigma, \delta_2, p_0, \{p_0\}) \text{ where } P = \{p_0, p_1, \ldots, p_{m_2-1}\}, \tag{8}$$

- $\delta_2(p_j, b) = p_{(j+1) \pmod{m_2}}$, $j = 0, 1, \ldots, m_2 - 1$,
- $\delta_2(p_j, a) = p_j$, $j = 0, 1, \ldots, m_2 - 1$,
- $\delta_2(p_0, c) = p_1$, $\delta_2(p_j, c) = p_j$, $j = 1, 2, \ldots, m_2 - 1$.

Above we assume that $R \cap P = \emptyset$. DFA A_1 and A_2 are shown in Figs. 1 and 2, respectively.

Let $A = (Q, \Sigma, \delta, q_0, F)$ be constructed from A_1 and A_2 as in (1). Here Q is as in (2). Since r_0 and p_0 are accepting states of A_i, for $i = 1, 2$, respectively, the start state q_0 is the set $\{r_0, p_0\}$. The set of accepting states consists of all subsets of $R \cup P$ such that $\{r_0, p_0\} \subseteq R \cup P$. Note that r_0 and p_0 are the only accepting states of A_1 and A_2, respectively.

It is sufficient to show that all states of Q are pairwise inequivalent and that all states of Q are reachable from the start state $\{r_0, p_0\}$.

First we show that all states are pairwise inequivalent. Let

$$U = U_1 \cup U_2 \quad \text{and} \quad V = V_1 \cup V_2 \tag{9}$$

be distinct states of Q, $U_1, V_1 \subseteq R$, $U_2, V_2 \subseteq P$. We consider the case where $U_1 \neq V_1$. The other possibility where $U_2 \neq V_2$ is completely symmetrical since the a- and b-transitions play symmetrical roles in A_1 and A_2, and c-transitions are defined in the same way in A_1 and A_2.

Without loss of generality we can assume that there exists an element $x \in U_1 - V_1$, the other possibility again being symmetrical. If $x = r_0$, then U is an accepting state of A. Note that $r_0 \in U_1$ and $U_1 \cup U_2 \in Q$ (where Q is as in (2)) imply that $p_0 \in U_2$. On the other hand, since V is an element of Q and $r_0 \notin V$, it follows that also $p_0 \notin V$. Thus V is not an accepting state, and consequently U and V are inequivalent.

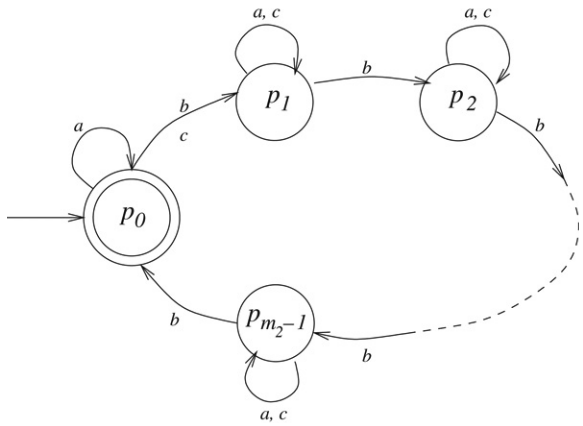

Fig. 2. DFA A_2.

In the following we consider the case where

$$x = r_i, \quad 1 \le i \le m_1 - 1. \tag{10}$$

We claim that

$$\delta(U, a^{m_1-i-1}ca) \in F. \tag{11}$$

Note that in A_1 the transition function δ_1 on input a^{m_1-i-1} takes the state r_i to r_{m_1-1}. The input c does not change state r_{m_1-1} and input a takes r_{m_1-1} to r_0. The last δ transition on input a according to (3) then adds also p_0 to the current set. Thus (11) holds.

On the other hand, we claim that

$$\delta(V, a^{m_1-i-1}ca) \notin F. \tag{12}$$

In A_2 the transitions corresponding to input a just cycle the states. Since $r_i \notin V_1$, it follows that $r_{m_1-1} \notin \delta_1(V_1, a^{m_1-i-1})$. In addition to simulating transitions of δ_1 (and δ_2), the transitions of δ, according to (3), sometimes can add the state r_0 (or p_0) to the current set of states. However, since $i \ge 1$, a computation started from an "added" state r_0 cannot on input a^{m_1-i-1} reach the state r_{m_1-1}. Thus we conclude that

$$r_{m_1-1} \notin \delta(V, a^{m_1-i-1}).$$

We note that c-transitions change r_0 and p_0 to, respectively, r_1 and p_1. The c-transitions keep all the other states fixed. Thus it follows that:

$$\delta(V, a^{m_1-i-1}c) \cap \{r_0, p_0, r_{m_1-1}\} = \emptyset. \tag{13}$$

Note that here we are using the property that

$$m_1 \ge 3, \qquad \text{that is, } r_{m_1-1} \ne r_1. \tag{14}$$

Now we use one more a-transition. Since a-transitions according to δ_2 do not change any states of $V_2 \subseteq P$, the Eq. (13) implies that

$$r_0, p_0 \notin \delta(V, a^{m_1-i-1}ca),$$

and (12) holds. Now (11) and (12) mean that U and V are inequivalent.

It remains to show that all states of Q (as in (2)) are reachable from the start state of A, $\{r_0, p_0\}$. First we establish some claims. The first claim says that all elements of Q that contain from P only the state p_0 or p_1 (respectively, from R only the state r_0 or r_1) are reachable.

Claim 1. (i) (a) *Let $r_0 \in R' \subseteq R$. Then $R' \cup \{p_0\}$ is reachable.*

(b) *Let $r_0 \notin R' \subseteq R$. Then $R' \cup \{p_1\}$ is reachable.*

(ii) (a) *Let* $p_0 \in P' \subseteq P$. *Then* $\{r_0\} \cup P'$ *is reachable.*

(b) *Let* $p_0 \notin P' \subseteq P$. *Then* $\{r_1\} \cup P'$ *is reachable.*

Proof. We consider only case (i). Case (ii) is completely symmetric due to the symmetry in the definitions of A_1 and A_2. We use induction on $|R'|$. First let $|R'| = 1$. Now in case (a), $R' \cup \{p_0\} = \{r_0, p_0\}$ and there is nothing to prove. For case (b) we observe that $\delta(\{r_0, q_0\}, c) = \{r_1, q_1\}$ and from $\{r_1, q_1\}$ we can reach any state $\{r_i, q_1\}$, $1 \le i \le m_1 - 1$, using only a-transitions.

Now inductively we assume that (i) holds for all subsets R' of R of cardinality at most k, $1 \le k < m_1$, and consider the case where

$$R' = \{r_{i_1}, r_{i_2}, \ldots, r_{i_{k+1}}\}, \quad 0 \le i_1 < i_2 < \ldots i_{k+1} \le m_1 - 1.$$

First in (a) we have that $i_1 = 0$. By the inductive assumption the set

$$X = \{r_{i_2}, \ldots, r_{i_{k+1}}\} \cup \{p_1\}$$

is reachable. We note that $r_0 \notin X$ since $i_2 > 0$. Now

$$\delta(X, b^{m_2-1}) = R' \cup \{p_0\}. \tag{15}$$

Note that the transitions of δ_2 on input b^{m_2-1} take p_1 to p_0. When reaching state p_0 the transitions of δ according to (3) add the state r_0 to the current set. Otherwise b-transitions do not change any of the elements $r_{i_2}, \ldots, r_{i_{k+1}}$, and (15) holds.

Second for (b) we consider the case $i_1 > 0$. By the inductive assumption the set

$$Y = \{r_{i_2-i_1+1}, \ldots, r_{i_{k+1}-i_1+1}\} \cup \{p_1\}$$

is reachable. Again $r_0 \notin Y$. We note that

$$\delta(Y, b^{m_2-1}c) = Y \cup \{r_1, p_1\}.$$

On input b^{m_2-1} everything works as above in (15), that is, δ_2 takes p_1 to p_0, and δ adds r_0 to the current set. Then c takes r_0 and p_0 to r_1 and p_1, respectively. Finally,

$$\delta(Y \cup \{r_1, p_1\}, a^{i_1-1}) = R' \cup \{p_1\}. \quad \blacksquare$$

We introduce the following notions: let

$$R_1 = \{r_{i_1}, \ldots, r_{i_k}\}, \quad 0 \le i_1 < \cdots < i_k < m_1,$$

be a subset of R. By the *cyclic* j*-shift* of R_1, $0 \le j \le m_1 - 1$, we mean the set $\{r_{i'_1}, \ldots, r_{i'_k}\}$ where

$$i'_s = i_s + j \pmod{m_1}, \quad s = 1, \ldots, k.$$

The cyclic j-shift of R_1 is denoted as $R_1[j]$. In the same way we define the cyclic j-shift of a subset P_1 of P, $0 \le j \le m_2 - 1$.

Intuitively, the following lemma says that if the current subset of P contains p_1, then the current subset of R can perform an arbitrary cyclic shift, and this operation does not change the subset of P except by adding/removing the state p_0. Recall that according to transitions of A as defined in (3) the current state must always have both r_0 and p_0, or neither one of them.

Claim 2. *Let* $R_1 \subseteq R$ *and* $P_1 \subseteq P$ *and assume that* $p_1 \in P_1$. *Let* $j \in \{1, \ldots, m_1 - 1\}$. *Denote*

$$P'_1 = \begin{cases} P_1 \cup \{p_0\} \text{ if } r_0 \in R_1[j], \\ P_1 - \{p_0\} \text{ if } r_0 \notin R_1[j]. \end{cases} \tag{16}$$

Then $R_1[j] \cup P'_1$ *is reachable from* $R_1 \cup P_1$.

Proof. Assume originally $p_0 \notin P_1$ and $r_0 \notin R_1$. We can cycle the set R_1 using a-transitions. When a state of R_1 reaches r_0 this adds p_0 to the set P_1. As long as consecutive elements of R_1 are "passing by" r_0 we continue to cycle just using a-transitions. When after a sequence of consecutive elements of R_1 have "passed by" r_0, the last a-transition adds a "new" element r_0 to the subset of R (since p_0 is in the current set). Then we apply a c-transition that removes both r_0 and p_0. After this the shifted version of R_1 again has the same number of elements as R_1. Since p_1 is in the original set P_1, after the c-transition the current subset of P is again P_1.

In case originally $p_0 \in P_1$ and $r_0 \in R_1$, we again cycle using a-transitions as long as elements of R_1 are passing by r_0, and after that apply a c-transition. This transition removes the element r_0 that was added by the last a-transition. The c-transition removes also p_0 and this is in accordance with the definition of P_1' in (16). ■

Now we show that all states of Q of the form $R_1 \cup P_1$, where

$$R_1 \neq R \quad \text{and} \quad P_1 \neq P \tag{17}$$

are reachable. The case where $R_1 = R$ or $P_1 = P$ is handled separately afterwards. Let R_1 and P_1 be arbitrary as in (17).

Let $R_1[j]$, $1 \leq j \leq m_1 - 1$, be a cyclic shift of R_1 such that $r_0 \notin R_1[j]$ and $r_1 \in R_1[j]$. We can find j as above since $R_1 \neq R$. By Claim 1 (i), $R_1[j] \cup \{p_1\}$ is reachable.

Denote

$$P_1 = \{p_{i_1}, \ldots, p_{i_k}\}, \quad 0 \leq i_1 < \cdots < i_k \leq m_2 - 1, \quad 1 \leq k < m_2.$$

We show that the set $R_1[j] \cup P_1$ is reachable from $R_1[j] \cup \{p_1\}$. Intuitively, this is done by "generating" the elements of P_1 one-by-one starting from p_k, i.e. p_1 becomes p_k and the rest of the elements are added using transition rules (3) of A.

We begin the computation from the state $R_1[j] \cup \{p_1\}$ by reading $i_k - i_{k-1} - 1$ symbols b. These transitions do not add any states of P and just move p_1 to "position $p_{i_k - i_{k-1}}$".[1] Next we apply a-transitions until some element of $R_1[j]$ reaches position r_0, we denote the resulting cyclic shift of R_1 by $R_1[j']$. Since $r_0 \in R_1[j']$, the last a-transition adds also p_0 to the current set of states. Now we apply a b-transition, this shifts p_0 and $p_{i_k - i_{k-1}}$ by one, and adds a "new" element p_0 to the current set (since r_0 remains in the current set). Thus the current set has the following elements of P:

$$\{p_0, p_1, p_{i_k - i_{k-1} + 1}\} \tag{18}$$

Now by Claim 2 we can cyclically shift $R_1[j']$ back to the set $R_1[j]$. According to Claim 2, this operation changes the current subset (18) of P just by removing p_0 (since $r_0 \notin R_1[j]$).

We continue in the same way to add new elements of P. After the next step the current set has elements p_1, $p_{i_{k-1} - i_{k-2} + 1}$ and $p_{i_k - i_{k-2} + 1}$ of P. If $i_1 > 0$ (that is, $p_0 \notin P_1$), in this way we add all k elements of P_1 but keep them in a position that is shifted "backwards" $i_1 - 1$ steps so that p_{i_1} is in position p_1, p_{i_2} is in position $p_{i_2 - i_1 + 1}$, and so on. Then using Claim 2, we cyclically shift $R_1[j]$ to R_1. According to Claim 2 this does not change the current elements of P. Note that since $p_0 \notin P_1$ and $R_1 \cup P_1 \in Q$, it has to be that $r_0 \notin R_1$. Finally using input word $b^{i_1 - 1}$ we shift all the elements of P into correct positions. This does not change the elements of R_1 and the resulting set is $R_1 \cup P_1$.

The situation is slightly different when $i_1 = 0$, that is, $p_0 \in P_1$, and consequently also $r_0 \in R_1$. Now as above we first generate the $k - 1$ elements of $P_1 - \{p_0\}$. Then using Claim 2 we shift $R_1[j]$ to a set $R_1[j']$ that has the property that $r_0 \notin R_1[j']$ and R_1 is obtained from $R_1[j']$ by a-transitions in such a way that all the intermediate sets of elements contain r_0. Note that $R_1[j']$ is simply the cyclic shift of R_1 where the continuous segment of states of R_1 that overlaps r_0 is just about to reach r_0. After this operation, using b-transitions we shift the current $k - 1$ elements of P into their correct positions $p_{i_2}, \ldots, p_{i_k}$. Finally using a-transitions we transform $R_1[j']$ to R_1. The first a-transition adds the state p_0 to the current set and the remaining a-transitions do not change any elements of P. By the choice of the cyclic shift $R_1[j']$ this sequence of a-transitions does not at any point add a new element r_0. The resulting set is again $R_1 \cup P_1$.

[1] In order to make it clear that we are referring to a position in the cycle of states of P we slightly abuse terminology by referring to the position as "$p_{i_k - i_{k-1}}$" instead of "position $i_k - i_{k-1}$".

Finally it remains to consider how A can reach the states not covered by (17). We show that any set $(R \cup P') \in Q$, $P' \subseteq P$ is reachable. By Claim 1 (ii) we know that $\{r_0\} \cup P'$ is reachable. Using induction on k it is easy to see that for $0 \le k \le m_1 - 1$,

$$\delta(\{r_0\} \cup P', a^k) = \{r_0, r_1, \ldots, r_k\} \cup P'.$$

Note that since $R \cup P' \in Q$, necessarily $p_0 \in P'$. Hence each a-transition shifts the elements of R by one step and adds a new element r_0. It follows that:

$$\delta(\{r_0\} \cup P', a^{m_1}) = R \cup P'.$$

Completely symmetrically we see that any set $R' \cup P \in Q$, $R' \subseteq R$ is reachable. Thus we have shown that all states of Q are reachable.

3.2.2. Case $m_1 = 2$ and $m_2 \ge 3$

In the previous subsection we constructed DFAs A_i of size $m_i \ge 3$, $i = 1, 2$, such that the state complexity of $(L(A_1) \cup L(A_2)^*)$ reaches the upper bound (6). The only place where the argument for the correctness of the construction used the assumption $m_i \ge 3$ was in (14).[2]

Here we modify the construction of the automata A_1 and A_2 by adding one more alphabet symbol. Let $m_1 = 2$, $m_2 \ge 3$ and $\Sigma = \{a, b, c, d\}$. The DFAs A_1 and A_2 are defined as in (7) and (8) with the following additional d-transitions:

$$\delta_1(r_0, d) = r_1, \quad \delta_1(r_1, d) = r_0, \tag{19}$$

and,

$$\delta_2(p_j, d) = p_1, \quad \text{for all } j = 0, 1, \ldots, m_2 - 1. \tag{20}$$

Let A again be constructed from A_1 and A_2 as in (1). Since we have shown that all the states of A are reachable using the original transitions for $\{a, b, c\}$, the same holds also with one more alphabet symbol. As observed above, the proof showing that all states of A are pairwise inequivalent used the assumption $m_1 \ge 3$ only in (14). We use the notations from the earlier proof, see (9). The two distinct states of A are $U = U_1 \cup U_2$ and $V = V_1 \cup V_2$. It is sufficient to consider the case (10) where $x \in U_1 - V_1$ and $x \ne r_0$. Since $m_1 = 2$, this means that $x = r_1$. We observe that with the added d-transitions

$$\delta(U, d) \in F,$$

since $\delta_1(r_1, d) = r_0$. On the other hand,

$$\delta(V, d) \notin F,$$

since necessarily $V_1 = \{r_0\}$ and in δ_2 the d-transitions take all states to p_1.

Thus when A_1 and A_2 have the additional d-transitions (19) and (20) all states of A will be pairwise inequivalent also in the case $m_1 = 2$.

3.2.3. Case $m_1 = m_2 = 2$

Using the idea from the previous subsection we could handle the case $m_1 = m_2 = 2$ by adding one further alphabet symbol with transitions defined by interchanging the roles of A_1 and A_2. However, the following direct example establishes that this case can be handled with an alphabet of size four:

Example 3.1. Let $\Sigma = \{a, b, c, d\}$. We define the following DFAs:

- $A_1 = (\{r_0, r_1\}, \Sigma, \delta_1, r_0, \{r_0\})$, $\delta(r_0, a) = r_1$, $\delta(r_1, a) = r_0$, $\delta(r_0, b) = r_0$, $\delta(r_1, b) = r_1$, $\delta(r_0, c) = r_1$, $\delta(r_1, c) = r_0$, $\delta(r_0, d) = r_1$, $\delta(r_1, d) = r_1$.
- $A_2 = (\{p_0, p_1\}, \Sigma, \delta_2, p_0, \{p_0\})$, $\delta(p_0, a) = p_0$, $\delta(p_1, a) = p_1$, $\delta(p_0, b) = p_1$, $\delta(p_1, b) = p_0$, $\delta(p_0, c) = p_1$, $\delta(p_1, c) = p_1$, $\delta(p_0, d) = p_1$, $\delta(p_1, d) = p_0$.

[2] It is easy to verify that the automata (7) and (8) with $m_1 = 2$ do not reach the upper bound (6).

A direct computation verifies that the minimal DFA for the language $(L(A_1) \cup L(A_2))^*$ has 5 states. This coincides with the upper bound (6) when $m_1 = m_2 = 2$. ■

3.2.4. The result

Theorem 3.2. *Let L_i be recognized by a minimal DFA with m_i states, $i = 1, 2$.*

(i) *If $m_1, m_2 \geq 2$, then*

$$\mathrm{sc}((L_1 \cup L_2)^*) \leq 2^{m_1+m_2-1} - 2^{m_1-1} - 2^{m_2-1} + 1 \tag{21}$$

and there exist languages L_1, L_2 for which the above relation becomes an equality.

(ii) *If $m_1 = 1$ and $m_2 \geq 2$, then*

$$\mathrm{sc}((L_1 \cup L_2)^*) \leq 2^{m_2-1} + 2^{m_2-2} \tag{22}$$

and there exist languages L_1, L_2 for which the above relation becomes an equality.

Proof. First consider the case where $m_1, m_2 \geq 2$. Now in (21) the inequality holds by (6). The construction of the DFAs A_1 and A_2 in subsections 3.2.1, 3.2.2 and 3.2.3 (corresponding to the different cases where $m_1, m_2 \geq 3$ or one or both of m_1, m_2 may be equal to two) establishes that the bound (21) may have the equality in the worst case. We have not explicitly shown that the DFAs A_1 and A_2 used in the construction are minimal. This fact follows from the observation that otherwise the minimal DFA A constructed for $(L(A_1) \cup L(A_2))^*$ would violate the upper bound (6).

For case (ii), if $m_1 = 1$ then $L_1 = \Sigma^*$ or $L_1 = \emptyset$. In the former case $\mathrm{sc}(L_1 \cup L_2)^* = 1$. In the latter case $(L_1 \cup L_2)^* = L_2^*$ and (22) follows by the state complexity upper bound for Kleene-star [21,22]. For the worst-case example we can choose $L_1 = \emptyset$ and for L_2 use the worst-case example for the state complexity of the Kleene-star operation [21,22]. ■

In the last remaining case, if $m_1 = m_2 = 1$ then, assuming that L_1 and L_2 are over the same alphabet Σ, $(L_1 \cup L_2)^*$ is necessarily Σ^* or $\{\varepsilon\}$. Note that if we take the union of languages over distinct alphabets, the state complexity of the operation would change since we require that all DFAs are complete.

The worst-case example for the tight upper bound of Theorem 3.2 uses a three-letter alphabet when $m_i \geq 3$, $i = 1, 2$, and a four-letter alphabet when one or both of m_1 and m_2 can be equal to two. It remains an open question whether the upper bound (6) can be reached by regular languages over a two-letter alphabet.

4. Star of intersection

If L_1 is accepted by a DFA with m states and L_2 is accepted by a DFA with n states, from [21] we know that the state complexity of $(L_1 \cap L_2)^*$ is at most $2^{mn-1} + 2^{mn-2}$. In contrast to the above section we show that the state complexity of star of intersection may become at least reasonably close to the composition of the worst-case state complexities of the individual operations.

In the following, we first consider the general cases, i.e. when $m, n \geq 3$, and then consider small values of m and n at the end of the section.

4.1. The general cases

Let $m, n \geq 3$ and L_1 and L_2 be accepted by DFAs of m states and n states, respectively. First we give a construction where $\mathrm{sc}((L_1 \cap L_2)^*)$ reaches at least

$$2^{m(n-2)} \tag{23}$$

Afterwards we describe how this bound can be improved.

Let $A = (P, \Sigma, \delta_A, 0, \{m-1\})$ and $B = (Q, \Sigma, \delta_B, 0, \{n-1\})$,

$$P = \{0, 1, \ldots, m-1\}, \quad Q = \{0, 1, \ldots, n-1\}, \quad m, n \geq 3,$$

$$\Sigma = \{a, b, c, d, e\}$$

For easier readability transitions defined by δ_A and δ_B will be listed later (when they are used).

The language $(L_1 \cap L_2)^*$ is recognized by the DFA $C = (U, \Sigma, \delta_C, r_0, F)$ where

$$\begin{aligned} U\ &= \{r_0\} \cup \{X \mid \emptyset \neq X \subseteq P \times Q - \{(m-1, n-1)\}\} \\ &\cup \{Y \subseteq P \times Q \mid (0,0), (m-1, n-1) \in Y\} \end{aligned} \tag{24}$$

Here $r_0 \notin P \times Q$ is a new symbol, and δ_C is defined by the following: let $x \in \Sigma$.

Then $\delta_C(r_0, x) = \{(\delta_A(0, x), \delta_B(0, x)\}$, and for $Z \subseteq P \times Q$ (such that Z is in U)

$$\delta_C(Z, x) = \begin{cases} \{(\delta_A(z_1, x), \delta_B(z_2, x) \mid (z_1, z_2) \in Z\} \cup \{(0,0)\} \\ \qquad \text{if } (m-1, n-1) = (\delta_A(z_1, x), \delta_B(z_2, x)) \text{ for some } (z_1, z_2) \in Z, \\ \{(\delta_A(z_1, x), \delta_B(z_2, x)) \mid (z_1, z_2) \in Z\}, \text{ otherwise.} \end{cases} \tag{25}$$

and $F = \{r_0\} \cup \{X \in U \mid (m-1, n-1) \in X\}$.

In the following when there is no danger of confusion, we identify a singleton set $\{(i, j)\}$ with (i, j), $(i, j) \in P \times Q$. For example, we can write $\delta_C((i, j), x)$ instead of $\delta_C(\{(i, j)\}, x)$, $x \in \Sigma$.

First we show that all the states of C are inequivalent. For this purpose we define the transitions for input elements a and b as follows:

$$\delta_A(i, a) = i + 1 (\text{mod } m), \quad i = 0, \ldots, m-1. \tag{26}$$

$$\delta_A(i, b) = i, \quad i = 0, \ldots, m-1. \tag{27}$$

$$\delta_B(j, a) = j, \quad j = 0, \ldots n-1. \tag{28}$$

$$\delta_B(j, b) = j + 1 (\text{mod } n), \quad j = 0, \ldots, n-1. \tag{29}$$

The state r_0 is an accepting state and cannot be equivalent with any state $Z \subseteq P \times Q$ that does not contain the pair $(m-1, n-1)$. On the other hand, if $(m-1, n-1) \in Z$ then $\delta_C(Z, a^m) \in F$ and $\delta_C(r_0, a^m) \notin F$.

Next consider any distinct states $Z_1, Z_2 \in P \times Q$. Without loss of generality there is an element $(i, j) \in Z_1 - Z_2$, $0 \le i \le m-1$, $0 \le j \le n-1$. We observe that $\delta_C(Z_1, a^{m-i-1}b^{n-j-1}) \in F$ and $\delta_C(Z_2, a^{m-i-1}b^{n-j-1}) \notin F$. Note that $\delta_A \times \delta_B$ with input $a^{m-i-1}b^{n-j-1}$ takes state (i, j) to $(m-1, n-1)$, and with this input $\delta_A \times \delta_B$ does not take any state $(i', j') \neq (i, j)$ to the state $(m-1, n-1)$. Also, note that δ_C differs from $\delta_A \times \delta_B$ only in that according to (25) the transitions of δ_C may create additional copies of $(0, 0)$. However, with (any suffix of) $a^{m-i-1}b^{n-j-1}$, δ_C does not take $(0, 0)$ to $(m-1, n-1)$ unless $i = j = 0$. Thus $(m-1, n-1) \notin \delta_C(Z_2, a^{m-i-1}b^{n-j-1})$. This means that Z_1 and Z_2 are not equivalent.

The total number of states of C is $2^{mn-1} + 2^{mn-2}$. Since all the states are pairwise inequivalent, the state complexity of $(L_1 \cap L_2)^*$ is at least the number of reachable states of C. In order to establish the lower bound (23), it is sufficient to show that any set $X \subseteq P \times Q$ such that

$$\{(i, n-2) \mid 0 \le i \le m-1\} \subseteq X \quad \text{and} \quad \{(i, n-1) \mid 0 \le i \le m-1\} \cap X = \emptyset \tag{30}$$

is reachable in C. This is done in a sequence of claims.

Claim 3. *$P \times Q$ is reachable.*

Proof. Let $(i, j) \in P \times Q$ be any state. When performing the transitions of $\delta_A \times \delta_B$ starting from (i, j) and using input word $(a^{m-1}b)^n$, the state cycles through all the elements of $P \times Q$. Thus if $Z \subset P \times Q$ (that is, Z is a proper subset of $P \times Q$) when computing $\delta_C(Z, (a^{m-1}b)^n)$ necessarily the computation starting from some element of Z reaches the state $(m-1, n-1)$ at a time when no state is "at $(0, 0)$". More formally, there exists a prefix w of $(a^{m-1}b)^n$ such that $(m-1, n-1) \in (\delta_A \times \delta_B)(Z)$ and $(0, 0) \notin (\delta_A \times \delta_B)(Z)$. Let $w = w'x$, $x \in \{a, b\}$. According to (25) this means that the transition $\delta_C(\delta_C(Z, w'), x)$ adds $(0, 0)$ to the current subset of $P \times Q$. On the other hand, for any $Y \subseteq P \times Q$ and $x \in \{a, b\}$, $|\delta_C(Y, x)| \ge |Y|$. Note that $(\delta_A \times \delta_B)(\cdot, x)$ is a permutation of $P \times Q$ and δ_C is the same as $\delta_A \times \delta_B$ except that it may add the state $(0, 0)$.

Thus $\delta_C(r_0, (a^{m-1}b)^{n \cdot (mn-1)} = P \times Q$. ■

Note that so far we have defined the transitions of A and B only for input symbols a and b. Now we continue with the definition of δ_A and δ_B for inputs c, d and e.

$$\delta_A(i, c) = i, \quad i = 0, \ldots, m-1, \tag{31}$$

$$\delta_B(j, c) = n-2, \quad j = 0, , \ldots, n-1. \tag{32}$$

Thus, $\delta_C(\cdot, c)$ will take any element (i, j) to $(i, n-2)$.

$$\delta_A(i, d) = i, \quad i = 0, \ldots, m-1 \tag{33}$$

$$\delta_B(j, d) = j, \ j = 0, \ldots, n-3, \quad \delta_B(n-2, d) = n-1, \ \delta_B(n-1, d) = n-2. \tag{34}$$

The above means that $\delta_C(\cdot, d)$ only interchanges elements $(i, n-2)$ and $(i, n-1)$, $1 \le i \le m-1$.

$$\delta_A(i, e) = i, \quad i = 0, \ldots, m-1, \tag{35}$$

$$\delta_B(j, e) = j+1, \ j = 0, \ldots, n-4, \quad \delta_B(k, e) = k, \ k \in \{n-3, n-2, n-1\}. \tag{36}$$

The operation $\delta_C(\cdot, e)$ changes (i, j) to $(i, j+1)$ when $0 \le j \le n-4$. Otherwise, δ_C is the identity.

We introduce some notation. For $0 \le j \le n-1$ let

$$Y_j = \{(i, j) \mid 0 \le i \le m-1\}. \tag{37}$$

First we note that $\delta_C(P \times Q, c) = \{(i, n-2) \mid 0 \le i \le m-1\} = Y_{n-2}$. Combining this observation with Claim 3 we get:

Claim 4. *The set Y_{n-2} is reachable.* ■

Let $0 \le j \le n-2$. We define the following family of sets:

$$\mathcal{T}_j = \{Z \subseteq P \times Q \mid (\forall k)(j \le k \le n-3) \Rightarrow Y_k \cap Z = \emptyset \ \& \\ Y_{n-2} \subseteq Z \ \& \ Y_{n-1} \cap Z = \emptyset\}.$$

The family $\mathcal{T}_0$ consists of exactly the set Y_{n-2}. For $1 \le j \le n-2$ the family $\mathcal{T}_j$ consists of all sets Z such that $Y_{n-2} \subseteq Z$, Z does not have any elements of sets $Y_j, Y_{j+1}, \ldots, Y_{n-3}$ and no elements of Y_{n-1}. The intersection of Z with $Y_0, \ldots, Y_{j-1}$ can be arbitrary.

Claim 5. *All sets in $\mathcal{T}_j$ are reachable, $j \in \{0, \ldots, n-2\}$.*

Proof. We prove the claim using induction on j. The claim holds when $j = 0$ since $\mathcal{T}_0$ consists of only the set Y_{n-2} which is reachable by Claim 4. Assume now that the claim holds when $j = r$, $0 \le r \le n-3$. An arbitrary subset of $\mathcal{T}_{r+1}$ can be written in the form

$$X = Z_0 \cup Z_1 \cup \cdots \cup Z_r \cup Y_{n-2} \tag{38}$$

where Z_s is an arbitrary subset of Y_s, $s = 0, \ldots, r$. (The notation Y_s is as in (37).) For $s = 0, \ldots, r-1$, denote

$$Z'_s = \{(i, s) \mid (i, s+1) \in Z_{s+1}, 0 \le i \le m-1\}.$$

We denote

$$X' = Z'_0 \cup Z'_1 \cup \cdots \cup Z'_{r-1} \cup Y_{n-2}.$$

We note that $X' \in \mathcal{T}_r$, and by the induction hypothesis X' is reachable. Since the transition $\delta_C(\cdot, e)$ just increases the second component of elements by one when the second component is at most $n-4$, we note that $\delta_C(X', e) = X''$ where

$$X'' = Z_1 \cup \cdots \cup Z_r \cup Y_{n-2}.$$

Now we just need to show how we can add the states of Z_0 to X''. We perform m transitions with input symbol a. After these transitions each Z_s, $1 \le s \le r$, will be the same as before. Always when $(i, 0) \in Z_0$, $0 \le i \le m-1$, we can add $(i, 0)$ to X'' by performing

$$\text{two } d \text{ transitions after the } (m-i)\text{th } a\text{-transition}, 0 \le i \le m-1. \tag{39}$$

Note that the first d-transition "moves" Y_{n-2} to Y_{n-1}. Since $(m-1, n-1) \in Y_{n-1}$, this transition adds $(0, 0)$ to X''. The second d-transition moves Y_{n-1} back to Y_{n-2}. The two consecutive d-transitions do not change the set in any other way except by adding the element $(0, 0)$. When the pairs of d-transitions are inserted in between the m a-transitions according to (39), after this sequence of transitions the set X'' has been changed to $X'' \cup Z_0$ which is the set we wanted in (38). ■

The family $\mathcal{T}_{n-2}$ consists of exactly all the sets (30). It follows that the state complexity of the language recognized by C is at least $2^{m(n-2)}$ as claimed in (23).

If we are allowed to add three new symbols to Σ, we can add to the DFA C transitions that change the first components of elements of $P \times Q$ analogously as the transitions (31)–(36) change the second components. Let us denote the modified DFA by C'. The new transitions together with the a- and b-transitions (26)–(29) give us $2^{n(m-2)}$ reachable subsets of $P \times Q$. By deducting the subsets that occur also in (30) we obtain the following lower bound for the state complexity of $L(C')$

$$\mathrm{sc}(L(C')) \geq 2^{m(n-2)} + 2^{n(m-2)} - 2^{mn-2(m+n+1)}. \tag{40}$$

Strictly speaking C' will of course have more reachable states than the lower bound given by (40). For example, just by using the a-transitions and b-transitions as given by (26)–(29) from the states in (30) we can obtain further new states. However, computing the precise number of reachable states would be quite involved (since translations into states of Y_{n-1} may contain the final state $(m-1, n-1)$ which will then change also the subset of Y_0). The lower bound (40) has a simple format.

We state the result in the following theorem:

Theorem 4.1. *Let $m_1, m_2 \geq 3$ and Σ be of cardinality eight. Then there exist languages L_i over Σ recognized by an m_i-state DFA, $i = 1, 2$, such that*

$$\mathrm{sc}((L_1 \cap L_2)^*) \geq 2^{m_1(m_2-2)} + 2^{m_2(m_1-2)} - 2^{m_1 m_2 - 2(m_1+m_2+1)}. \quad \blacksquare$$

By increasing the alphabet size the lower bound (40) could be further increased. However, we do not know whether the bound given by the individual state complexities of intersection and star, $2^{m_1 m_2 - 1} + 2^{m_1 m_2 - 2}$, can be reached.

We conjecture that the above bound can be reached if we are allowed to use a variable size alphabet, that is, if Σ may depend on m_1 and m_2.

4.2. Small values of *m* and *n*

By using large alphabets it can be verified that when $m, n \leq 3$, the star-of-intersection can reach exactly $2^{mn-1} + 2^{mn-2}$. We have an example of two three-state DFA's A_1, A_2 such that $\mathrm{sc}(L(A_1) \cap L(A_2))^* = 384$.

5. Conclusion

We have studied the state complexities of two different combinations of operations on regular languages: star of union and star of intersection. In the first combination, the state complexity of the combination is very different from the composition of the state complexities of the individual operations. In the second combination, they are very similar. These two results show that although the composition of the state complexities of individual operations gives an upper bound to the state complexity of the combination of individual operations, this upper bound may or may not be tight. The tight upper bound can be far from this bound. Our first result in this paper has shown this case. The state complexity of a combination of operations has to be studied individually in order to know its result. There are many combinations of operations on regular languages that are worth studying. Hopefully many new results on this topic will be obtained in the near future.

For further reading

[1,10].

References

[1] J.-C. Birget, Partial orders on words, minimal elements of regular languages, and state complexity, Theoretical Computer Science 119 (1993) 267–291.

[2] C. Campeanu, K. Culik, K. Salomaa, S. Yu, State complexity of basic operations on finite languages, in: Proceedings of the Fourth International Workshop on Implementing Automata VIII 1-11, in: LNCS, vol. 2214, 1999, pp. 60–70.

[3] C. Campeanu, K. Salomaa, S. Yu, Tight lower bound for the state complexity of shuffle of regular languages, Journal of Automata, Languages and Combinatorics 7 (3) (2002) 303–310.

[4] C. Campeanu, K. Salomaa, S. Yu, Chapter 5: state complexity of regular languages: finite versus infinite, in: C. Calude, G. Paun (Eds.), Finite vs Infinite — Contributions to an Eternal Dilemma, Springer, 2000, pp. 53–73.
[5] M. Domaratzki, State complexity and proportional removals, Journal of Automata, Languages and Combinatorics 7 (2002) 455–468.
[6] M. Holzer, M. Kutrib, State complexity of basic operations on nondeterministic finite automata, in: Proceedings of International Conference on Implementation and Application of Automata 2002, CIAA 2002, in: LNCS, vol. 2608, Springer, 2002, pp. 148–157.
[7] M. Holzer, M. Kutrib, Unary language operations and their nondeterministic state complexity, in: Developments in Language Theory, DLT 2002, in: LNCS, vol. 2450, Springer, 2002, pp. 162–172.
[8] M. Holzer, K. Salomaa, S. Yu, On the state complexity of k-entry deterministic finite automata, Journal of Automata, Languages and Combinatorics 6 (4) (2001) 453–466.
[9] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison Wesley, Reading, Mass, 1979.
[10] K. Iwama, Y. Kambayashi, K. Takaki, Tight bounds on the number of states of DFAs that are equivalent to n-state NFAs, Theoretical Computer Science 237 (2000) 485–494.
[11] G. Jirásková, State complexity of some operations on regular languages, in: Proceedings of 5th Workshop on Descriptional Complexity of Formal Systems, 2003, pp. 114–125.
[12] G. Jirásková, State complexity of some operations on binary regular languages, Theoretical Computer Science 330 (2005) 287–298.
[13] J. Jirásek, G. Jirásková, A. Szabari, State complexity of concatenation and complementation of regular languages, International Journal of Foundations of Computer Science 16 (2005) 511–529.
[14] G. Jirásková, A. Okhotin, State complexity of cyclic shift, in: Proceedings of DCFS 2005, Como, Italy, June 30–July 2, 2005, pp. 182–193.
[15] C. Nicaud, Average State Complexity of Operations on Unary Automata, in: MFCS'99, in: LNCS, vol. 1672, 1999, pp. 231–240.
[16] G. Pighizzini, J. Shallit, Unary language operations, state complexity and Jacobsthal's function, International Journal of Foundations of Computer Science 13 (1) (2002) 145–159.
[17] A. Salomaa, Theory of Automata, Pergamon Press, Oxford, 1969.
[18] A. Salomaa, D. Wood, S. Yu, On the state complexity of reversals of regular languages, Theoretical Computer Science 320 (2004) 293–313.
[19] K. Salomaa, S. Yu, NFA to DFA transformation for finite languages over arbitrary alphabets, Journal of Automata, Languages and Combinatorics 2 (3) (1997) 177–186.
[20] S. Yu, State complexity of regular languages, Journal of Automata, Languages and Combinatorics 6 (2) (2001) 221–234.
[21] S. Yu, Q. Zhuang, K. Salomaa, The state complexities of some basic operations on regular languages, Theoretical Computer Science 125 (1994) 315–328.
[22] S. Yu, Regular languages, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. 1, Springer-Verlag, 1997, pp. 41–110.