State complexity of union and intersection of square and reversal on k regular languages[☆]

Yuan Gao[a], Lila Kari[a], Sheng Yu[a]

[a] *Department of Computer Science, The University of Western Ontario, London, Ontario, Canada N6A 5B7*

Abstract

In the paper, we continue our study on the state complexity of combined operations on regular languages. We study the state complexities of $\bigcup_{i=1}^{k} L_i^2$, $\bigcap_{i=1}^{k} L_i^2$, $\bigcup_{i=1}^{k} L_i^R$, and $\bigcap_{i=1}^{k} L_i^R$, for regular languages L_i, $1 \le i \le k$. We obtain the exact bounds for these combined operations and show that the state complexities of $\bigcup_{i=1}^{k} L_i^2$ and $\bigcap_{i=1}^{k} L_i^2$ are the same as the mathematical compositions of the state complexities of their component individual operations, while, on the other hand, the state complexities of $\bigcup_{i=1}^{k} L_i^R$ and $\bigcap_{i=1}^{k} L_i^R$ are lower than the corresponding mathematical compositions.

Keywords: state complexity, combined operations, regular languages, finite automata

1. Introduction

State complexity of finite automata which is the number of states of finite automata, is an important, ongoing topic in formal languages and automata theory. Nowadays, finite automata of very large sizes are widely used in software engineering, programming languages, natural language and speech processing, and other practical areas. These applications make the research on state complexity essential and well-motivated.

The earliest research on state complexity dates back to the 1950s [20]. However, most results were obtained after 1990 with the help of powerful computers

[☆] This research was supported by Natural Science and Engineering Council of Canada Discovery Grant R2824A01, Canada Research Chair Award to L. K, and Natural Science and Engineering Council of Canada Discovery Grant 41630 to S. Y.

Email addresses: `ygao72@csd.uwo.ca` (Yuan Gao), `lila@csd.uwo.ca` (Lila Kari), `syu@csd.uwo.ca` (Sheng Yu)

and software for experiments, e.g. Grail+ [29]. Existing literature includes studies of the state complexity of individual operations, such as catenation, union, intersection, star, reversal, shuffle, power, proportional removal, cyclic shift, etc [1, 4, 5, 6, 11, 13, 14, 15, 19, 25, 26, 27].

However, in practice, it is often the case that the operation to be performed on finite automata is not just a single individual operation, but a combination of several individual operations in some specific order. This motivated the study of state complexity of combined operations which started in 2007 [23]. In [23], the state complexities of $(L_1 \cup L_2)^*$ and $(L_1 \cap L_2)^*$ are investigated, and it is pointed out that the mathematical composition of the state complexities of the component individual operations of a combined operation cannot be directly used as the state complexity of the combined operation. Indeed, the state complexity of the combined operation can be much lower than its corresponding mathematical composition. For example, let L_1 and L_2 be two regular languages accepted by m- and n-state deterministic finite automata (DFAs), respectively. The state complexity of L_1^* is known to be $\frac{3}{4}2^m$ and the state complexity of L_1L_2 is $m2^n - 2^{n-1}$ [18, 27]. Then the mathematical composition of these two state complexities for the combined operation $(L_1L_2)^*$ is

$$\frac{3}{4}2^{2^{m2^n-2^{n-1}}}.$$

However, the state complexity of $(L_1L_2)^*$ is only [8]

$$2^{m+n-1} + 2^{m+n-4} - 2^{m-1} - 2^{n-1} + m + 1.$$

From this example, we can see that although the mathematical composition of the state complexities of component individual operations does serve as an upper bound of the state complexity of the combined operation, this upper bound usually cannot be reached. Recently, it has also been shown that there does not exist a general algorithm to compute the state complexities of combined operations even if all the state complexities of individual operations are known [24]. Thus, the state complexity of each combined operation should be investigated separately.

A number of results on the state complexity of combined operations have been obtained in the past four years. Most of these results are concerned with the combined operations that consist of two different individual operations, e.g. $(L_1 \cup L_2)^*$, $(L_1 \cap L_2)^*$, $(L_1L_2)^*$, $(L_1^R)^*$ $(L_1 \cup L_2)^R$, $(L_1 \cap L_2)^R$, $(L_1L_2)^R$, etc [2, 3, 8, 10, 16, 17, 23]. Besides these basic combined operations, only a few combined operations composed of arbitrarily many individual operations have been investigated, including L^k, $L_1L_2 \cdots L_k$, and combined Boolean operations on $L_1, L_2, \ldots, L_k$ [6, 7, 9]. Clearly, combined operations with arbitrarily many individual operations are more general than basic combined operations because the latter can be viewed as the special cases of the former. Therefore, combined operations with arbitrarily many individual operations should be the emphasis of the study of state complexity of combined operations.

In this paper, we study the state complexities of four particular combined operations on k operand languages, $\bigcup_{i=1}^{k} L_i^2$, $\bigcap_{i=1}^{k} L_i^2$, $\bigcup_{i=1}^{k} L_i^R$, and $\bigcap_{i=1}^{k} L_i^R$, where L_i is a regular language accepted by an n_i-state DFA, $1 \leq i \leq k$. We show that the state complexities of $\bigcup_{i=1}^{k} L_i^2$ and $\bigcap_{i=1}^{k} L_i^2$ are both

$$\prod_{i=1}^{k}(n_i 2^{n_i} - 2^{n_i-1})$$

for $n_i \geq 3$ and $k \geq 2$, the same as the mathematical compositions of the state complexities of their component operations.

For $\bigcup_{i=1}^{k} L_i^R$ and $\bigcap_{i=1}^{k} L_i^R$, we prove that their state complexities are both

$$\prod_{i=1}^{k}(2^{n_i} - 1) + 1$$

for $n_i \geq 3$ and $k \geq 2$. In contrast to the other two combined operations, in this case the state complexities of these two combined operations are lower than the mathematical compositions of the state complexities of their component operations.

In the next section, we introduce the basic definitions and notations used in the paper. In Sections 3 and 4, we investigate the state complexities of $\bigcup_{i=1}^{k} L_i^2$, $\bigcap_{i=1}^{k} L_i^2$, respectively. In Section 5, the state complexities of $\bigcap_{i=1}^{k} L_i^R$ and $\bigcup_{i=1}^{k} L_i^R$ are shown. In Section 6, we conclude the paper.

2. Preliminaries

A DFA is denoted by a 5-tuple $A = (Q, \Sigma, \delta, s, F)$, 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, $s \in Q$ 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 $a \in \Sigma$. All the DFAs we mention in this paper are assumed to be complete. We extend δ to $Q \times \Sigma^* \to Q$ in the usual way.

A *non-deterministic finite automaton* (NFA) is denoted by a 5-tuple $A = (Q, \Sigma, \delta, s, F)$, where the definitions of Q, Σ, s, and F are the same to those of DFAs, but the state transition function δ is defined as $\delta : Q \times \Sigma \to 2^Q$, where 2^Q denotes the power set of Q, i.e. the set of all subsets of Q. An NFA can have multiple initial states, which is not the usual convention. In this case, the NFA can be denoted by a 5-tuple $A = (Q, \Sigma, \delta, S, F)$, where S is the set of the initial states.

In this paper, the state transition function δ of a DFA is often extended to $\hat{\delta} : 2^Q \times \Sigma \to 2^Q$. 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.

A string $w \in \Sigma^*$ is accepted by a DFA (an NFA) if $\delta(s, w) \in F$ ($\delta(s, w) \cap F \neq \emptyset$). Two states in a finite automaton A are said to be *equivalent* if and only if for every string $w \in \Sigma^*$, if A is started in either state with w as input, it either accepts in both cases or rejects in both cases. It is well-known that a language which is accepted by an NFA can be accepted by a DFA, and such a language is said to be *regular*. The language accepted by a DFA A is denoted by $L(A)$. The reader may refer to [12, 22, 28] for more details about regular languages and finite automata.

The *state complexity* of a regular language L, denoted by $sc(L)$, is the number of states of the minimal complete DFA that accepts L. The state complexity of a class S of regular languages, denoted by $sc(S)$, is the supremum among all $sc(L)$, $L \in S$. The state complexity of an operation on regular languages is the state complexity of the resulting languages from the operation as a function of the state complexity of the operand languages. Thus, in a certain sense, the state complexity of an operation is a worst-case complexity.

3. State complexity of $L_1^2 \cup L_2^2 \cup \cdots \cup L_k^2$

We first consider the state complexity of $\bigcup_{i=1}^{k} L_i^2$, where L_i, $1 \leq i \leq k$ is a regular language accepted by an n_i-state DFA. It has been proved that the state complexity of L_1^2 is $n_1 2^{n_1} - 2^{n_1 - 1}$ [21] and the state complexity of $L_1 \cup L_2$ is $n_1 n_2$ [18, 27]. Their mathematical composition is

$$\prod_{i=1}^{k} (n_i 2^{n_i} - 2^{n_i - 1}).$$

In the following, we show that this upper bound of the state complexity of $\bigcup_{i=1}^{k} L_i^2$ can be reached.

Theorem 3.1. *For integers $n_i \geq 3$ and $k \geq 2$, there exists a DFA N_i of n_i states such that any DFA accepting $\bigcup_{i=1}^{k} L(N_i)^2$ needs at least*

$$\prod_{i=1}^{k} (n_i 2^{n_i} - 2^{n_i - 1})$$

states.

Proof. Let $N_i = (Q_{N_i}, \Sigma, \delta_{N_i}, 0, F_{N_i})$ be a DFA, where $Q_{N_i} = \{0, 1, \ldots, n_i - 1\}$, $n_i \geq 3$, $\Sigma = \{a_{i,j} \mid 1 \leq i \leq k, j \in \{1, 2\}\}$, $F_{N_i} = \{n_i - 1\}$, and the transitions of N_i are

$$\delta_{N_i}(p, a_{i,1}) = p + 1 \bmod n_i,\ p = 0, 1, \ldots, n_i - 1,$$
$$\delta_{N_i}(1, a_{i,2}) = 0,\ \delta_{N_i}(p, a_{i,2}) = p,\ p = 0, 2, 3 \ldots, n_i - 1,$$
$$\delta_{N_i}(p, c) = p,\ c \in \Sigma - \{a_{i,1}, a_{i,2}\},\ p = 0, 1, \ldots, n_i - 1.$$

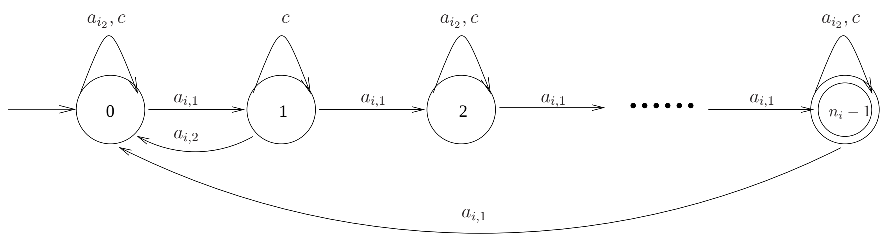

Figure 1: Witness DFA N_i for Theorems 3.1.

The transition diagram of N_i is shown in Figure 1.

It has been shown in [21] that the minimal DFA that accepts the square of an n_i-state DFA language has $n_i 2^{n_i} - 2^{n_i-1}$ states in the worst case. The DFA N_i is a modification of the witness DFA used in [21] by adding c-loops to every state, where $c \in \Sigma - \{a_{i,1}, a_{i,2}\}$. So we can similarly design an $(n_i 2^{n_i} - 2^{n_i-1})$-state, minimal DFA $N'_i = (Q_{N'_i}, \Sigma, \delta_{N'_i}, s_{N'_i}, F_{N'_i})$ that accepts $L(N_i)^2$, where

$$\begin{aligned} &Q_{N'_i} = Q_{N_i} \times 2^{Q_{N_i}} - F_{N_i} \times 2^{Q_{N_i} - \{0\}}, \\ &s_{N'_i} = \langle 0, \emptyset \rangle, \\ &F_{N'_i} = \{\langle u, V \rangle \in Q_{N'_i} \mid V \cap F_{N_i} \neq \emptyset\}, \end{aligned}$$

and for $\langle u, V \rangle \in Q_{N'_i}$ and $a \in \Sigma$,

$$\delta_{N'_i}(\langle u, V \rangle, a) = \begin{cases} \langle \delta_{N_i}(u, a), \delta_{N_i}(V, a) \rangle, & \text{if } n_i - 1 \notin \delta_{N_i}(u, a); \\ \langle \delta_{N_i}(u, a), \delta_{N_i}(V, a) \cup \{0\} \rangle, & \text{otherwise.} \end{cases}$$

Then we construct the DFA $A = (Q, \Sigma, \delta, s, F)$ that accepts $\bigcup_{i=1}^{k} L(N_i)^2$, where

$$\begin{aligned} &Q = \{\langle p_1, p_2, \ldots, p_k \rangle \mid p_i \in Q_{N'_i}, 1 \leq i \leq k\}, \\ &s = \langle s_{N'_1}, s_{N'_2}, \ldots, s_{N'_k} \rangle, \\ &\delta(\langle p_1, p_2, \ldots, p_k \rangle, a) = \langle \delta_{N'_1}(p_1, a), \delta_{N'_2}(p_2, a), \ldots, \delta_{N'_k}(p_k, a) \rangle,\ a \in \Sigma, \\ &F = \{\langle p_1, p_2, \ldots, p_k \rangle \in Q \mid \exists i (p_i \in F_{N'_i}, 1 \leq i \leq k)\}. \end{aligned}$$

In the following, we show that the DFA A is minimal.

(I) All the states in Q are reachable.
For an arbitrary state $\langle p_1, p_2, \ldots, p_k \rangle$ in Q, there always exists a string $w_1 w_2 \cdots w_k$ such that $\delta(s, w_1 w_2 \cdots w_k) = \langle p_1, p_2, \ldots, p_k \rangle$, where

$$\delta_{N'_i}(s_{N'_i}, w_i) = p_i,\ w_i \in \{a_{i,1}, a_{i,2}\}^*,\ 1 \leq i \leq k.$$

(II) Any two different states $\langle p_1, p_2, \ldots, p_k \rangle$ and $\langle q_1, q_2, \ldots, q_k \rangle$ in Q are distinguishable.

Without loss of generality, we assume that $p_i \neq q_i$, $1 \leq i \leq k$. Let $p_j = \langle u_j, V_j \rangle$ and $q_j = \langle x_j, Y_j \rangle$ for all $1 \leq j \leq k$. Then there exists a string $w = w_1 w_2 \cdots w_k$ such that

$$\delta(\langle p_1, p_2, \ldots, p_k \rangle, w) \in F,$$
$$\delta(\langle q_1, q_2, \ldots, q_k \rangle, w) \notin F,$$

where $w_i \in \{a_{i,1}, a_{i,2}\}^*$, $\delta_{N_i'}(p_i, w_i) \in F_{N_i'}$ and $\delta_{N_i'}(q_i, w_i) \notin F_{N_i'}$, and for $1 \leq l \leq k$, $l \neq i$,

$$w_l = a_{l,1}^{n_l - x_l}(a_{l,1}a_{l,2}a_{l,1}^{n_l})^{n_l - 1} a_{l,2}(a_{l,1}a_{l,2})^{n_l - 2}.$$

Note that

$$\delta_{N_l'}(\langle x_l, Y_l \rangle, a_{l,1}^{n_l - x_l}(a_{l,1}a_{l,2}a_{l,1}^{n_l})^{n_l - 1}) = \langle 0, Q_{N_l} \rangle$$

and

$$\delta_{N_l'}(\langle 0, Q_{N_l} \rangle, a_{l,2}(a_{l,1}a_{l,2})^{n_l - 2}) = \langle 0, \{0\} \rangle \notin F_{N_l'}.$$

Since all the states in A are reachable and pairwise distinguishable, A is a minimal DFA. Thus, any DFA that accepts $\bigcup_{i=1}^{k} L(N_i)^2$ has at least

$$\prod_{i=1}^{k} (n_i 2^{n_i} - 2^{n_i - 1})$$

states, for $k \geq 2$ and $n_i \geq 3$. □

This result gives a lower bound for the state complexity of $\bigcup_{i=1}^{k} L(N_i)^2$. It coincides with the upper bound we stated at the beginning of the section. Therefore, we have the following theorem.

Theorem 3.2. *For integers $n_i \geq 3$ and $k \geq 2$, $\prod_{i=1}^{k} (n_i 2^{n_i} - 2^{n_i - 1})$ states are both sufficient and necessary in the worst case for a DFA to accept $\bigcup_{i=1}^{k} L(N_i)^2$, where N_i is an n_i-state DFA.*

4. State complexity of $L_1^2 \cap L_2^2 \cap \cdots \cap L_k^2$

In this section, we study the state complexity of $\bigcap_{i=1}^{k} L_i^2$, where L_i is a regular language accepted by an n_i-state DFA, $1 \leq i \leq k$. Since $\overline{L}^2 \neq \overline{L^2}$, we cannot directly obtain the state complexity of $\bigcap_{i=1}^{k} L_i^2$ from that of $\bigcup_{i=1}^{k} L_i^2$ through De

Morgan's laws. The mathematical composition of the state complexities of square and intersection is also

$$\prod_{i=1}^{k}(n_i 2^{n_i} - 2^{n_i-1})$$

because the state complexity of $L_1 \cap L_2$ is the same as that of $L_1 \cup L_2$. We will show that this upper bound of the state complexity of $\bigcap_{i=1}^{k} L_i^2$ can be reached by some worst-case examples.

Theorem 4.1. *For integers $n_i \geq 3$ and $k \geq 2$, there exists a DFA N_i of n_i states such that any DFA accepting $\bigcap_{i=1}^{k} L(N_i)^2$ needs at least*

$$\prod_{i=1}^{k}(n_i 2^{n_i} - 2^{n_i-1})$$

states.

Proof. We use the same DFA N_i as in the proof of Theorem 3.1. Construct an $(n_i 2^{n_i} - 2^{n_i-1})$-state, minimal DFA $N_i' = (Q_{N_i'}, \Sigma, \delta_{N_i'}, s_{N_i'}, F_{N_i'})$ for $L(N_i)^2$ in the same way as in the proof of Theorem 3.1.

Then we construct the DFA $A = (Q, \Sigma, \delta, s, F)$ that accepts $\bigcap_{i=1}^{k} L(N_i)^2$ exactly as described in the proof of Theorem 3.1 except that

$$F = \{\langle p_1, p_2, \ldots, p_k \rangle \in Q \mid \forall i(p_i \in F_{N_i'}, 1 \leq i \leq k)\}.$$

Next, we will show that A is minimal. The proof for the reachability of an arbitrary state in A is omitted, because it is the same as that in the proof of Theorem 3.1.

We now prove that any two different states $\langle p_1, p_2, \ldots, p_k \rangle$ and $\langle q_1, q_2, \ldots, q_k \rangle$ of A are distinguishable. We may assume, without loss of generality that $p_i \neq q_i$, $1 \leq i \leq k$. Then there exists a string $w = w_1 w_2 \cdots w_k$ such that

$$\delta(\langle p_1, p_2, \ldots, p_k \rangle, w) \in F,$$
$$\delta(\langle q_1, q_2, \ldots, q_k \rangle, w) \notin F,$$

where $w_i \in \{a_{i,1}, a_{i,2}\}^*$,

$$\delta_{N_i'}(p_i, w_i) \in F_{N_i'},$$
$$\delta_{N_i'}(q_i, w_i) \notin F_{N_i'},$$

and for $1 \leq l \leq k$, $l \neq i$, $w_l \in \{a_{l,1}, a_{l,2}\}^*$ and $\delta_{N_l'}(p_l, w_l) \in F_{N_l'}$.

Since all the states in A can be reached from the initial state and are pairwise distinguishable, the DFA A is minimal. Thus, any DFA that accepts $\bigcap_{i=1}^{k} L(N_i)^2$ has at least

$$\prod_{i=1}^{k}(n_i 2^{n_i} - 2^{n_i-1})$$

states, for $n_i \geq 3$ and $k \geq 2$. $\square$

The lower bound shown in Theorem 4.1 coincides with the mathematical composition of the state complexities of square and intersection. Thus, the following theorem holds.

Theorem 4.2. *For integers $n_i \geq 3$ and $k \geq 2$, $\prod_{i=1}^{k}(n_i 2^{n_i} - 2^{n_i-1})$ states are both sufficient and necessary in the worst case for a DFA to accept $\bigcap_{i=1}^{k} L(N_i)^2$, where N_i is an n_i-state DFA.*

5. State complexity of $L_1^R \cap L_2^R \cap \cdots \cap L_k^R$ and $L_1^R \cup L_2^R \cup \cdots \cup L_k^R$

In this section, we investigate the state complexity of $\bigcap_{i=1}^{k} L_i^R$, where L_i, $1 \leq i \leq k$ is a regular language accepted by an n_i-state DFA. It has been shown that the state complexity of L_1^R is 2^{n_1} and the state complexity of $L_1 \cap L_2$ is $n_1 n_2$ [18, 27]. Then their mathematical composition is $2^{\sum_{i=1}^{k} n_i}$ which is an upper bound of the state complexity of $\bigcap_{i=1}^{k} L_i^R$. In the following, we will show this upper bound can be lowered.

Theorem 5.1. *For any n_i-state DFA $N_i = (Q_{N_i}, \Sigma, \delta_{N_i}, s_{N_i}, F_{N_i})$, $1 \leq i \leq k$, $k \geq 2$, there exists a DFA of at most*

$$\prod_{i=1}^{k}(2^{n_i} - 1) + 1$$

states that accepts $\bigcap_{i=1}^{k} L(N_i)^R$.

Proof. Let $N_i = (Q_{N_i}, \Sigma, \delta_{N_i}, s_{N_i}, F_{N_i})$ be a DFA of n_i states, $1 \leq i \leq k$, $k \geq 2$. Let $N_i' = (Q_{N_i}, \Sigma, \delta_{N_i'}, s_{N_i'}, F_{N_i'})$ be an NFA with multiple initial states, where

$$s_{N_i'} = F_{N_i},$$
$$F_{N_i'} = \{s_{N_i}\},$$
$$\delta_{N_i'}(p, a) = \{q \mid \delta_{N_i}(q, a) = p\},\ a \in \Sigma \text{ and } p, q \in Q_{N_i}.$$

Clearly, the NFA N_i' accepts $L(N_i)^R$. By performing the subset construction on the NFA N_i', we can get an equivalent, 2^{n_i}-state DFA $A_i = (Q_{A_i}, \Sigma, \delta_{A_i}, s_{A_i}, F_{A_i})$ such that $L(A_i) = L(N_i)^R$. Note that $\emptyset$ is a state in Q_{A_i}.

Now let $A = (Q, \Sigma, \delta, s, F)$ be another DFA, where

$$\begin{aligned}
&s = \langle s_{A_1}, s_{A_2}, \ldots, s_{A_k}\rangle,\\
&Q = \{\langle p_1, p_2, \ldots, p_k\rangle \mid p_i \in Q_{A_i}, 1 \le i \le k\},\\
&\delta(\langle p_1, p_2, \ldots, p_k\rangle, a) = \langle \delta_{A_1}(p_1, a), \delta_{A_2}(p_2, a), \ldots, \delta_{A_k}(p_k, a)\rangle,\ a \in \Sigma,\\
&F = \{\langle p_1, p_2, \ldots, p_k\rangle \in Q \mid \forall i (p_i \in F_{N_i'}, 1 \le i \le k)\}.
\end{aligned}$$

It is easy to see that

$$L(A) = \bigcap_{i=1}^{k} L(A_i) = \bigcap_{i=1}^{k} L(N_i)^R.$$

The number of states in A is $2^{\sum_{i=1}^{k} n_i}$. However, some of these states are indeed equivalent. Consider two different states $\langle \emptyset, p_2, \ldots, p_k\rangle$ and $\langle q_1, \emptyset, \ldots, q_k\rangle$. Clearly,

$$\begin{aligned}
\langle \emptyset, p_2, \ldots, p_k\rangle \notin F,\\
\langle q_1, \emptyset, \ldots, q_k\rangle \notin F,
\end{aligned}$$

and for any string $w \in \Sigma^*$,

$$\begin{aligned}
\delta(\langle \emptyset, p_2, \ldots, p_k\rangle, w) = \langle \emptyset, p_2', \ldots, p_k'\rangle \notin F,\\
\delta(\langle q_1, \emptyset, \ldots, q_k\rangle, w) = \langle q_1', \emptyset, \ldots, q_k'\rangle \notin F,
\end{aligned}$$

because $\emptyset$ is a sink state in A_i. We can see that the two states $\langle \emptyset, p_2, \ldots, p_k\rangle$ and $\langle q_1, \emptyset, \ldots, q_k\rangle$ are equivalent. Thus, all the states $\langle p_1, p_2, \ldots, p_k\rangle$ such that $p_i = \emptyset$, $1 \le i \le k$, can be merged into one state. The number of states $\langle t_1, t_2, \ldots, t_k\rangle \in Q$ such that none of $t_1, t_2, \ldots, t_k$ is $\emptyset$, is $\prod_{i=1}^{k}(2^{n_i} - 1)$. Then there are in total

$$\prod_{i=1}^{k}(2^{n_i} - 1) + 1$$

states in A. Thus, we obtain the upper bound in the statement of Theorem 5.1. $\square$

Theorem 5.2. *For integers $n_i \ge 3$ and $k \ge 2$, there exists a DFA N_i of n_i states such that any DFA accepting $\bigcap_{i=1}^{k} L(N_i)^R$ needs at least*

$$\prod_{i=1}^{k}(2^{n_i} - 1) + 1$$

states.

Proof. Let $N_i = (Q_{N_i}, \Sigma, \delta_{N_i}, 0, F_{N_i})$ be a DFA, where $Q_{N_i} = \{0, 1, \ldots, n_i - 1\}$, $n_i \geq 3$, $\Sigma = \{a_{i,j} \mid 1 \leq i \leq k, j \in \{1, 2, 3\}\}$, $F_{N_i} = \{0\}$, and the transitions of N_i are

$$\begin{aligned}
&\delta_{N_i}(0, a_{i,1}) = n_i - 1,\ \delta_{N_i}(p, a_{i,1}) = p - 1,\ p = 1, \ldots, n_i - 1,\\
&\delta_{N_i}(0, a_{i,2}) = 1,\ \delta_{N_i}(p, a_{i,2}) = p,\ p = 1, 2, 3 \ldots, n_i - 1,\\
&\delta_{N_i}(0, a_{i,3}) = 1,\ \delta_{N_i}(1, a_{i,3}) = 0,\ \delta_{N_i}(p, a_{i,3}) = p,\ p = 2, 3 \ldots, n_i - 1,\\
&\delta_{N_i}(p, c) = p,\ c \in \Sigma - \{a_{i,1}, a_{i,2}, a_{i,3}\},\ p = 0, 1, \ldots, n_i - 1.
\end{aligned}$$

The transition diagram of N_i is shown in Figure 2.

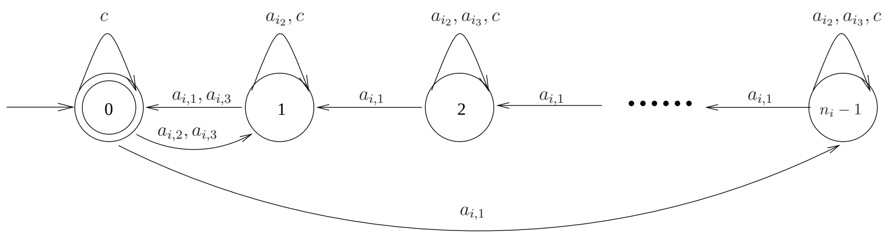

Figure 2: Witness DFA N_i for Theorems 5.2.

It has been shown in [27] that the minimal DFA that accepts the reversal of an n_i-state DFA language has 2^{n_i} states in the worst case. The DFA N_i in this proof is a modification of the witness DFA used in [27] by adding c-loops to every state, where $c \in \Sigma - \{a_{i,1}, a_{i,2}, a_{i,3}\}$. So we can similarly design an 2^{n_i}-state, minimal DFA $A_i = (Q_{A_i}, \Sigma, \delta_{A_i}, s_{A_i}, F_{A_i})$ that accepts $L(N_i)^R$, where

$$\begin{aligned}
&Q_{A_i} = 2^{Q_{N_i}},\\
&s_{A_i} = F_{N_i} = \{0\},\\
&F_{A_i} = \{P \in Q_{A_i} \mid 0 \in P\},
\end{aligned}$$

and for $P \in Q_{A_i}$ and $a \in \Sigma$,

$$\delta_{A_i}(P, a) = \{q \in Q_{N_i} \mid \delta_{N_i}(q, a) \in P\}.$$

Then we construct the DFA $A = (Q, \Sigma, \delta, s, F)$ that accepts $\bigcap\limits_{i=1}^{k} L(N_i)^R$, where

$$\begin{aligned}
&Q = \{\langle P_1, P_2, \ldots, P_k\rangle \mid P_i \in Q_{A_i}, P_i \neq \emptyset, 1 \leq i \leq k\} \cup \{\langle \emptyset, \emptyset, \ldots, \emptyset\rangle\},\\
&s = \langle s_{A_1}, s_{A_2}, \ldots, s_{A_k}\rangle,\\
&F = \{\langle P_1, P_2, \ldots, P_k\rangle \in Q \mid \forall i (p_i \in F_{A_i}, 1 \leq i \leq k)\}.
\end{aligned}$$

and for $P = \langle P_1, P_2, \ldots, P_k\rangle \in Q$ and $a \in \Sigma$,

$$\delta(P, a) = \begin{cases} \langle \emptyset, \emptyset, \ldots, \emptyset\rangle, & \text{if } \exists i (\delta_{A_i}(P_i, a) = \emptyset, 1 \leq i \leq k),\\ \langle \delta_{A_1}(P_1, a), \delta_{A_2}(P_2, a), \ldots, \delta_{A_k}(P_k, a)\rangle, & \text{otherwise.} \end{cases}$$

As we mentioned in the proof of Theorem 5.1, the states such that at least one of their components is $\emptyset$, are equivalent. Thus, we can merge them into one state, that is, $\langle \emptyset, \emptyset, \ldots, \emptyset \rangle$ and the number of states in A is

$$\prod_{i=1}^{k} (2^{n_i} - 1) + 1.$$

In the following, we show that the DFA A is minimal.

(I) All the states in Q are reachable.
For an arbitrary state $\langle P_1, P_2, \ldots, P_k \rangle$ in Q, there always exists a string $w_1 w_2 \cdots w_k$ such that $\delta(s, w_1 w_2 \cdots w_k) = \langle P_1, P_2, \ldots, P_k \rangle$, where

$$\delta_{A_i}(s_{A_i}, w_i) = P_i,\ w_i \in \{a_{i,1}, a_{i,2}, a_{i,3}\}^*,\ 1 \le i \le k.$$

(II) Any two different states $\langle P_1, P_2, \ldots, P_k \rangle$ and $\langle R_1, R_2, \ldots, R_k \rangle$ in Q are distinguishable.
When $\langle P_1, P_2, \ldots, P_k \rangle = \langle \emptyset, \emptyset, \ldots, \emptyset \rangle$ and $\langle R_1, R_2, \ldots, R_k \rangle \ne \langle \emptyset, \emptyset, \ldots, \emptyset \rangle$, the two states can be easily distinguished by a string $w = w_1 w_2 \cdots w_k$ where

$$\delta_{A_i}(R_i, w_i) \in F_{A_i},\ 1 \le i \le k,$$

because

$$\delta(\langle \emptyset, \emptyset, \ldots, \emptyset \rangle, w) = \langle \emptyset, \emptyset, \ldots, \emptyset \rangle \notin F,$$
$$\delta(\langle R_1, R_2, \ldots, R_k \rangle, w) \in F.$$

Next, let us consider the case when neither of the two states is $\langle \emptyset, \emptyset, \ldots, \emptyset \rangle$. Without loss of generality, we assume that $P_i \ne R_i$, $1 \le i \le k$. Then there exists a string $w = w_1 w_2 \cdots w_k$ such that

$$\delta(\langle P_1, P_2, \ldots, P_k \rangle, w) \in F,$$
$$\delta(\langle R_1, R_2, \ldots, R_k \rangle, w) \notin F,$$

where $w_i \in \{a_{i,1}, a_{i,2}, a_{i,3}\}^*$,

$$\delta_{A_i}(P_i, w_i) \in F_{A_i},$$
$$\delta_{A_i}(R_i, w_i) \notin F_{A_i},$$

and for $1 \le j \le k$, $j \ne i$, $w_j \in \{a_{j,1}, a_{j,2}, a_{j,3}\}^*$,

$$\delta_{A_j}(P_j, w_j) \in F_{A_j}.$$

Since all the states in A are reachable and pairwise distinguishable, A is a minimal DFA. Thus, we obtain the lower bound stated in Theorem 5.2. $\square$

The lower bound of the state complexity of $\bigcap_{i=1}^{k} L(N_i)^R$ in Theorem 5.2 coincides with the upper bound in Theorem 5.1. Therefore, we get the following theorem.

Theorem 5.3. *For any integers $n_i \geq 3$ and $k \geq 2$, the number of states that are both sufficient and necessary in the worst case for a DFA to accept $\bigcap_{i=1}^{k} L(N_i)^R$, where N_i is an n_i-state DFA, is*

$$\prod_{i=1}^{k}(2^{n_i} - 1) + 1.$$

The state complexity of $\bigcup_{i=1}^{k} L_i^R$ is the same as that of $\bigcap_{i=1}^{k} L_i^R$, since

$$\bigcup_{i=1}^{k} L_i^R = \overline{\bigcap_{i=1}^{k} \overline{L_i^R}} = \overline{\bigcap_{i=1}^{k} \overline{L_i}^R}$$

according to De Morgan's laws and $\overline{L_i^R} = \overline{L_i}^R$, where $\overline{L_i}$ denotes the complement of L_i, and the state complexity of the complementation of an n-state DFA language is n. Thus, we have the following theorem.

Theorem 5.4. *For any integers $n_i \geq 3$ and $k \geq 2$, the number of states that are both sufficient and necessary in the worst case for a DFA to accept $\bigcup_{i=1}^{k} L(N_i)^R$, where N_i is an n_i-state DFA, is*

$$\prod_{i=1}^{k}(2^{n_i} - 1) + 1.$$

6. Conclusion

In this paper, we studied the state complexities of union and intersection of squares of k regular languages, and union and intersection of reversals of k regular languages. We obtained the state complexities of the four particular combined operations $\bigcup_{i=1}^{k} L_i^2$, $\bigcap_{i=1}^{k} L_i^2$, $\bigcup_{i=1}^{k} L_i^R$, and $\bigcap_{i=1}^{k} L_i^R$, where L_i is a regular language accepted by an n_i-state DFA, $n_i \geq 3$, $1 \leq i \leq k$, and $k \geq 2$. The state complexities of the first two combined operations are equal. They are also exactly the same as the mathematical compositions of the state complexities of their component individual operations. The state complexities of the latter two combined operations are also equal, but lower than the corresponding mathematical compositions.

In this paper, all the results are proved with increasing alphabets. In the worst-case example for $\bigcup_{i=1}^{k} L_i^2$ and $\bigcap_{i=1}^{k} L_i^2$, an alphabet of the size $2k$ was used. The witness DFA for $\bigcap_{i=1}^{k} L_i^R$ is over a $3k$-letter alphabet. It is interesting to study whether the sizes of these alphabets can be reduced. However, it is impossible to design a worst-case example for arbitrary $k \geq 2$ and $n_i \geq 3$ with a fixed alphabet. Note that there are a limited number of different DFAs with a fixed number of states if the alphabet is fixed. Thus, when k is large enough, some of the operand DFAs with the same number of states may be indeed the same according to pigeonhole principle. Therefore, the study of state complexity of operations on k operand languages uses increasing alphabets in general.

Another possible future topic could be the state complexities of these combined operations on a smaller, fixed alphabet when k is also fixed. We expect more results on the state complexities of combined operations with arbitrarily many individual operations and operand languages.

References

[1] 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.

[2] B. Cui, Y. Gao, L. Kari, S. Yu: State complexity of two combined operations: catenation-star and catenation-reversal, *International Journal of Foundations of Computer Science*, accepted.

[3] B. Cui, Y. Gao, L. Kari, S. Yu: State complexity of two combined operations: catenation-union and catenation-intersection, *International Journal of Foundations of Computer Science*, accepted.

[4] M. Daley, M. Domaratzki, K. Salomaa: State complexity of orthogonal catenation, in: *Proceedings of Descriptional Complexity of Formal Systems*, Charlottetown, PE, Canada, July 16-18, 2008, 134-144.

[5] M. Domaratzki: State complexity and proportional removals, *Journal of Automata, Languages and Combinatorics* 7 (2002) 455-468.

[6] M. Domaratzki, A. Okhotin: State complexity of power, *Theoretical Computer Science* 410(24-25) (2009) 2377-2392.

[7] Z. Ésik, Y. Gao, G. Liu, S. Yu: Estimation of state complexity of combined operations, *Theoretical Computer Science* 410 (35) (2009) 3272-3280.

[8] Y. Gao, K. Salomaa, S. Yu: The state complexity of two combined operations: star of catenation and star of Reversal, *Fundamenta Informaticae* 83 (1-2) (2008) 75-89.

[9] Y. Gao and S. Yu: State complexity approximation, in:*Proceedings of Descriptional Complexity of Formal Systems* Magdeburg, Germany, 2009, 163-174.

[10] Y. Gao and S. Yu: State complexity of four combined operations composed of union, intersection, star and reversal, *Proceedings of Descriptional Complexity of Formal Systems*, Limburg, Gemany, July 25-27, 2011, LNCS 6808, 158-171.

[11] 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, LNCS 2608, 2002, 148-157.

[12] J. E. Hopcroft, R. Motwani, J. D. Ullman: *Introduction to Automata Theory, Languages, and Computation (2nd Edition)*, Addison Wesley, 2001.

[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: State complexity of some operations on binary regular languages, *Theoretical Computer Science* 330 (2005) 287-298.

[15] G. Jirásková, A. Okhotin: State complexity of cyclic shift, in: *Proceedings of DCFS 2005*, Como, Italy, June 30-July 2, 2005, 182-193.

[16] G. Jirásková, A. Okhotin: On the state complexity of star of union and star of intersection, *Turku Center for Computer Science TUCS Technical Report* No. 825, 2007.

[17] G. Liu, C. Martin-Vide, A. Salomaa, S. Yu: State complexity of basic language operations combined with reversal, *Information and Computation* 206 (2008) 1178-1186.

[18] A. Maslov: Estimates of the number of states of finite automata, *Soviet Mathematics Doklady*, 11 (1970) 1373-1375.

[19] 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.

[20] M. Rabin, D. Scott: Finite automata and their decision problems, *IBM Journal of Research and Development* 2 (3) (1959) 114-125.

[21] N. Rampersad: The state complexity of L^2 and L^k, *Information Processing Letters* 98 (2006) 231-234.

[22] G. Rozenberg, A. Salomaa: *Handbook of Formal Languages*, Springer-Verlag, 1997.

[23] A. Salomaa, K. Salomaa, S. Yu: State complexity of combined operations, *Theoretical Computer Science* 383 (2007) 140-152.

[24] A. Salomaa, K. Salomaa, S. Yu: Undecidability of the State Complexity of Composed Regular Operations, *Proceedings of Language and Automata Theory and Applications* Tarragona, Spain, LNCS 6638 , 2011, 489-498.

[25] A. Salomaa, D. Wood, S. Yu: On the state complexity of reversals of regular languages, *Theoretical Computer Science* 320 (2004) 293-313.

[26] S. Yu: State complexity of regular languages, *Journal of Automata, Languages and Combinatorics* 6 (2) (2001) 221-234.

[27] S. Yu, Q. Zhuang, K. Salomaa: The state complexity of some basic operations on regular languages, *Theoretical Computer Science* 125 (1994) 315-328.

[28] S. Yu: Regular languages, in: G. Rozenberg, A. Salomaa (Eds.), *Handbook of Formal Languages*, Vol. 1, Springer-Verlag, 1997, 41-110.

[29] Grail+, developed by Department of Computer Science, The University of Western Ontario, http://www.csd.uwo.ca/Research/grail