Theoretical Computer Science 437 (2012) 82–102

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

State complexity of combined operations with two basic operations ☆

Bo Cui, Yuan Gao *, Lila Kari, Sheng Yu

Department of Computer Science, The University of Western Ontario, London, Ontario, Canada N6A 5B7

ARTICLE INFO

Article history:
Received 7 July 2011
Received in revised form 4 November 2011
Accepted 3 February 2012
Communicated by G. Ausiello

Keywords:
State complexity
Combined operations
Regular languages
Finite automata

ABSTRACT

This paper studies the state complexity of $(L_1L_2)^R$, $L_1^RL_2$, $L_1^*L_2$, $(L_1 \cup L_2)L_3$, $(L_1 \cap L_2)L_3$, $L_1L_2 \cap L_3$, and $L_1L_2 \cup L_3$ for regular languages L_1, L_2, and L_3. We first show that the upper bound proposed by Liu et al. (2008) [18] for the state complexity of $(L_1L_2)^R$ coincides with the lower bound and is thus the state complexity of this combined operation by providing some witness DFAs. Also, we show that, unlike most other cases, due to the structural properties of the result of the first operation of the combinations $L_1^RL_2$, $L_1^*L_2$, and $(L_1 \cup L_2)L_3$, the state complexity of each of these combined operations is close to the mathematical composition of the state complexities of the component operations. Moreover, we show that the state complexities of $(L_1 \cap L_2)L_3$, $L_1L_2 \cap L_3$, and $L_1L_2 \cup L_3$ are exactly equal to the mathematical compositions of the state complexities of their component operations in the general cases. We also include a brief survey that summarizes all state complexity results for combined operations with two basic operations.

1. Introduction

State complexity is a type of descriptional complexity based on the *deterministic finite automaton* (DFA) model. The state complexity of an operation on regular languages is the number of states that are necessary and sufficient in the worst case for the minimal, complete DFA to accept the resulting language of the operation. While many results on the state complexity of individual operations, such as union, intersection, catenation, star, reversal, shuffle, power, orthogonal catenation, proportional removal, and cyclic shift [1,2,5–7,12,14–16,19,20,23,25,27], have been obtained in the past 15 years, the research on state complexity of combined operations, which was initiated by Salomaa et al. in 2007 [21], has recently attracted more attention. This is because, in practice, a combination of several individual operations, rather than only one individual operation, is often performed.

In recent publications [3,4,8–11,17,18,21,28], it has been shown that the state complexity of a combined operation is usually not a simple mathematical composition of the state complexities of its component operations. For example, let L_1 be an m-state DFA language and L_2 be an n-state DFA language. Recall that the state complexity of $L_1 \cup L_2$ (considered as $f(m, n)$) is mn and the state complexity of L_2^* (considered as $g(n)$) is $2^{n-1} + 2^{n-2}$. Thus, the composition of these state complexities ($g(f(m, n))$) gives $2^{mn-1} + 2^{mn-2}$ as an upper bound of the state complexity of $(L_1 \cup L_2)^*$. However, this upper bound is too high to be reached and the state complexity of this combined operation has been proven to be $2^{m+n-1} + 2^{m-1} + 2^{n-1} + 1$. This is due to the structural properties of the DFA that results from the first operation of a combined operation.

For example, let us consider reversal combined with catenation ($L_1^RL_2$). We know that, on one hand, if a DFA is obtained for L_1^R, where $m > 1$, and it reaches the upper bound of the state complexity of reversal (2^m), then half of its states are

☆ This work is 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.

* Corresponding author. Tel.: +1 519 661 2111 ext. 86278.
E-mail addresses: bcui2@csd.uwo.ca (B. Cui), ygao72@csd.uwo.ca (Y. Gao), lila@csd.uwo.ca (L. Kari), syu@csd.uwo.ca (S. Yu).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2012.02.030

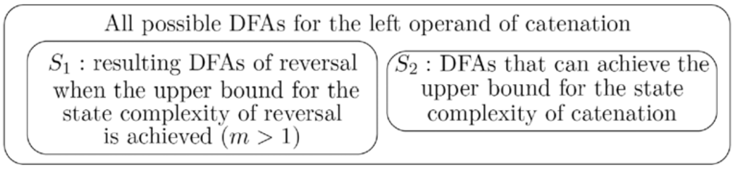

Fig. 1. The set S_1 of DFAs that are outputs of reversal when the upper bound for the state complexity of reversal is achieved *is disjoint* from the set S_2 of DFAs that are the left operand for catenation which can achieve the upper bound for the state complexity of catenation.

final [25]; on the other hand, in order to reach the upper bound of the state complexity of catenation, the DFA of its left operand language has to have only one final state [25]. This situation is depicted in Fig. 1. (In another example, the initial state of a DFA obtained from star is always a final state). In general, the resulting language obtained from the first operation (such as reversal, star, or union) may not be among the worst cases of the subsequent operation (such as catenation).

It has been shown that there does not exist a general algorithm that, for an arbitrarily given combined operation and a class of regular languages, computes the state complexity of the operation on the class of languages [22,24]. Thus, the state complexity of every combined operation must be investigated individually. Although the number of combined operations is unlimited, the study of the state complexity of combinations of two basic operations is clearly necessary since it is the initial step towards the study of combinations of more operations.

There are in total 26 different combinations of two basic operations selected from catenation, star, reversal, intersection, and union. Note that we consider $(L_1^R)^*$ and $(L_1^*)^R$ as the same combined operation because $(L_1^R)^* = (L_1^*)^R$. The combined operations $(L_1^*)^* = L_1^*$ and $(L_1^R)^R = L_1$ are not counted, either. Among the 26 combined operations, the state complexities of the following ones have been studied in the literature: $(L_1 \cup L_2)^*$ in [21], $(L_1 \cap L_2)^*$ in [17], $(L_1L_2)^*$, $(L_1^R)^*$ in [9], $(L_1 \cup L_2)^R$, $(L_1 \cap L_2)^R$, $(L_1L_2)^R$, $L_1L_2^*$, $L_1L_2^R$ in [3], $L_1(L_2 \cup L_3)$, $L_1(L_2 \cap L_3)$ in [4], $L_1^* \cup L_2$, $L_1^* \cap L_2$, $L_1^R \cup L_2$, $L_1^R \cap L_2$ in [11], $L_1L_2L_3$, the combined Boolean operations $L_1 \cup L_2 \cup L_3$, $L_1 \cap L_2 \cap L_3$, $(L_1 \cup L_2) \cap L_3$, and $(L_1 \cap L_2) \cup L_3$ in [8], where L_1, L_2, and L_3 are three regular languages.

In this paper, we study the state complexities of all the other combinations of two basic operations, namely $(L_1L_2)^R$, $L_1^RL_2$, $L_1^*L_2$, $(L_1 \cup L_2)L_3$, $(L_1 \cap L_2)L_3$, $L_1L_2 \cap L_3$, and $L_1L_2 \cup L_3$ for regular languages L_1, L_2, and L_3 accepted by DFAs of m, n, and p states, respectively.

Although the state complexity of $(L_1L_2)^R$ has been considered in [18], only an upper bound has been obtained. In this paper, we prove, by providing some witness DFAs, that the upper bound, $3 \cdot 2^{m+n-2} - 2^n + 1$, proposed in [18] is indeed the state complexity of this combined operation when $m \geq 2$ and $n \geq 1$.

We also show that, unlike some other combined operations, the state complexities of $(L_1 \cap L_2)L_3$, $L_1L_2 \cap L_3$, and $L_1L_2 \cup L_3$ in general cases are equal to the compositions of the state complexities of their component operations, while the state complexities of $L_1^RL_2$, $L_1^*L_2$ and $(L_1 \cup L_2)L_3$ are close to the compositions.

In the next section, we introduce the basic definitions and notations used in the paper. Then we prove our results on the state complexities of $(L_1L_2)^R$ in Section 3, $L_1^RL_2$ in Section 4, $L_1^*L_2$ in Section 5, $(L_1 \cup L_2)L_3$ in Section 6, $(L_1 \cap L_2)L_3$ in Section 7, $L_1L_2 \cap L_3$ in Section 8, and $L_1L_2 \cup L_3$ in Section 9. Section 10 summarizes our results and also provides an overview of the state complexity results of all possible combined operations with two basic operations.

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 [13,26] 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

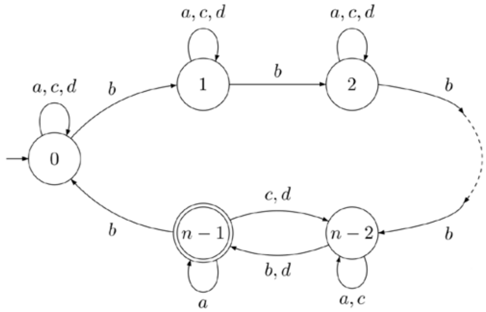

Fig. 2. Witness DFA N which shows that the upper bound of the state complexity of $(L(M)L(N))^R$, $3 \cdot 2^{m+n-2} - 2^n + 1$, is reachable when $m, n \geq 2$.

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 $(L_1L_2)^R$

In this section, we investigate the state complexity of $(L_1L_2)^R$ for an m-state DFA language L_1 and an n-state DFA language L_2, which has been an open problem since 2008. In [18], the following theorem concerning the upper bound of the state complexity of $(L_1L_2)^R$ was proved.

Theorem 3.1 ([18]). *Let L_1 and L_2 be an m-state DFA language and an n-state DFA language, respectively, with $m, n > 1$. Then there exists a DFA with no more than $3 \cdot 2^{m+n-2} - 2^n + 1$ states that accepts $(L_1L_2)^R$.*

In the following, we first show that this upper bound is reachable by some worst-case examples for $m, n \geq 2$ (Theorem 3.2). Then we investigate the state complexity of $(L_1L_2)^R$ when $m = 1$ (Theorem 3.3) or $n = 1$ (Theorem 3.4). Finally, we summarize the state complexity of $(L_1L_2)^R$ (Theorem 3.5).

Let us start with a general lower bound of the state complexity of $(L_1L_2)^R$ when $m, n \geq 2$.

Theorem 3.2. *Given two integers $m, n \geq 2$, there exists a DFA M of m states and a DFA N of n states such that any DFA accepting $(L(M)L(N))^R$ needs at least $3 \cdot 2^{m+n-2} - 2^n + 1$ states.*

Proof. Let $M = (Q_M, \Sigma, \delta_M, 0, \{m-1\})$ be a DFA, where $Q_M = \{0, 1, \ldots, m-1\}$, $\Sigma = \{a, b, c, d\}$, and the transitions are given as:

- $\delta_M(i, a) = i + 1 \mod m, i = 0, \ldots, m-1$,
- $\delta_M(i, h) = i, i = 0, \ldots, m-1, h \in \{b, c, d\}$.

Let $N = (Q_N, \Sigma, \delta_N, 0, \{n-1\})$ be a DFA, shown in Fig. 2, where $Q_N = \{0, 1, \ldots, n-1\}$, $\Sigma = \{a, b, c, d\}$, and the transitions are given as:

- $\delta_N(i, a) = i, i = 0, \ldots, n-1$,
- $\delta_N(i, b) = i + 1 \mod n, i = 0, \ldots, n-1$,
- $\delta_N(i, c) = i, i = 0, \ldots, n-2, \delta_N(n-1, c) = n-2$,
- $\delta_N(i, d) = i, i = 0, \ldots, n-3, \delta_N(n-2, d) = n-1, \delta_N(n-1, d) = n-2$.

Next we construct a DFA $D = (Q_D, \Sigma, \delta_D, s_D, F_D)$ to accept $(L(M)L(N))^R$, where

$$
\begin{aligned}
&Q_D = (R \cup S) - T,\\
&R = \{\langle R_1, R_2\rangle \mid R_1 \subseteq Q_M, R_2 \subseteq Q_N - \{0\}\},\\
&S = \{\langle R_1, R_2\rangle \mid R_1 \subseteq Q_M, m-1 \in R_1, R_2 \subseteq Q_N, 0 \in R_2\}\\
&T = \{\langle Q_M, R_2\rangle \mid R_2 \subseteq Q_N, R_2 \neq \emptyset\},\\
&s_D = \langle \emptyset, \{n-1\}\rangle,\\
&F_D = \{\langle R_1, R_2\rangle \in Q_D \mid 0 \in R_1\}.
\end{aligned}
$$

For any $g = \langle R_1, R_2\rangle \in Q_D, h \in \Sigma$, let $R_1' = \{p \in Q_M \mid \delta_M(p, h) \in R_1\}$, $R_2' = \{q \in Q_N \mid \delta_N(q, h) \in R_2\}$, and then δ_D is defined as follows,

$$
\delta_D(g, h) = \begin{cases} \langle R_1', R_2'\rangle, & \text{if } R_1' \neq Q_M, 0 \notin R_2',\\ \langle R_1' \cup \{m-1\}, R_2'\rangle, & \text{if } R_1' \cup \{m-1\} \neq Q_M, 0 \in R_2',\\ \langle Q_M, \emptyset\rangle, & \text{if } R_1' = Q_M, 0 \notin R_2',\\ \langle Q_M, \emptyset\rangle, & \text{if } R_1' \cup \{m-1\} = Q_M, 0 \in R_2'. \end{cases}
$$

Since M is a complete DFA, each state of M has an outgoing transition with each letter in Σ. It follows that $Q'_M = \{p \in Q_M \mid \delta_M(p, h) \in Q_M\} = Q_M$ for any $h \in \Sigma$. Note that $0 \in Q_M$, so every state $\langle Q_M, R_2 \subseteq Q_N\rangle$ is a final state. This means that all states starting with Q_M are equivalent. Thus, when we construct the DFA D, all such equivalent states are combined into one state, that is, $\langle Q_M, \emptyset\rangle$.

In the following, we will prove D is a minimal DFA.

(I) We first show that every state $\langle R_1, R_2\rangle \in Q_D$, is reachable from s_D. It can be seen that $\langle\emptyset, \emptyset\rangle = \delta_D(s_D, c)$ regardless of whether $n = 2$ or $n > 2$. Then we consider the other 3 cases.

Case 1: $R_1 = \emptyset, R_2 \neq \emptyset$.

It is trivial when $n = 2$, because $m - 1 \in R_1 \neq \emptyset$ if $0 \in R_2$. Therefore, we only discuss $n > 2$ and use induction on the size of R_2 to prove that the state can be reached from s_D. When $|R_2| = 1$, let R_2 be $\{i\}$, $1 \leq i \leq n - 1$. Then we have $\langle\emptyset, \{i\}\rangle = \delta_D(s_D, b^{n-1-i})$. Now assume that $\langle\emptyset, R_2\rangle \in Q_D$ is reachable from s_D when $|R_2| = k$. We will prove that $\langle\emptyset, R'_2\rangle \in Q_D$ is also reachable when $|R'_2| = k+1 \leq n-1$. We assume $R'_2 = \{q_1, q_2, \ldots, q_{k+1}\}$ such that $1 \leq q_1 < q_2 < \cdots < q_{k+1} \leq n-1$. Then

$$\langle\emptyset, R'_2\rangle = \delta_D(\langle\emptyset, R''_2\rangle, c(bd)^{q_{k+1}-q_k-1}b^{n-1-q_{k+1}}), \text{ where}$$

$$R''_2 = \{q_1 + n - q_k - 2, q_2 + n - q_k - 2, \ldots, q_{k-1} + n - q_k - 2, n - 2\}.$$

Note that $q_{k-1} + n - q_k - 2 < n - 2$ because $q_{k-1} < q_k$.

Case 2: $R_1 \neq \emptyset, R_2 = \emptyset$.

Let R_1 be $\{p_1, p_2, \ldots, p_k\}$ such that $0 \leq p_1 < p_2 < \cdots < p_k \leq m - 1$, $1 \leq k \leq m$. Then $\langle R_1, \emptyset\rangle = \delta_D(s_D, w')$, where

$$w' = b^n a^{p_2-p_1} b^n a^{p_3-p_2} \cdots b^n a^{p_k-p_{k-1}} b^n a^{m-1-p_k} c.$$

When $R_1 = \{p_1\}$, w' is $b^n a^{m-1-p_1} c$.

Case 3: $R_1 \neq \emptyset, R_2 \neq \emptyset$.

Assume $R_1 = \{p_1, p_2, \ldots, p_k\}$ such that $0 \leq p_1 < p_2 < \cdots < p_k \leq m - 1$, $1 \leq k \leq m - 1$. Note that k cannot be m in this case, because all the states starting with Q_M are equivalent and merged into $\langle Q_M, \emptyset\rangle$. We first use w'' to move the DFA D from s_D to $\langle R_1, \{n - 1\}\rangle$, where

$$w'' = b^n a^{p_2-p_1} b^n a^{p_3-p_2} \cdots b^n a^{p_k-p_{k-1}} b^n a^{m-1-p_k}.$$

Then $\langle R_1, R_2\rangle$ can be reached from $\langle R_1, \{n - 1\}\rangle$ by the strings shown in Case 1 because they consist of the letters b, c, d and cannot change R_1. If 0 shows up in R_2, p_k must be $m - 1$ and it has been included in R_1 during the processing of w'' and $R_1 \cup \{m - 1\} = R_1$. If $0 \notin R_2$, then 0 will not appear in the second element of the two-tuples (states) when processing the strings in Case 1 from the state $\langle R_1, \{n - 1\}\rangle$. Thus, the set R_1 will not be changed.

(II) Next, we show that any two different states $\langle R_1, R_2\rangle$, $\langle R'_1, R'_2\rangle \in Q_D$, are distinguishable. It is obvious when one state is final and the other is not. Therefore, we consider only when both the two states are final or non-final. There are three cases in the following.

1. $R_1 \neq R'_1$. Without loss of generality, we may assume that there exists x such that $x \in R_1 - R'_1$. A string a^x can distinguish the two states because

$$\delta_D(\langle R_1, R_2\rangle, a^x) \in F_D,$$
$$\delta_D(\langle R'_1, R'_2\rangle, a^x) \notin F_D.$$

   Note that $x \neq m - 1$ if $0 \in R'_2$.
2. $R_1 = R'_1 = \emptyset, R_2 \neq R'_2$. We may assume without loss of generality that there exists x such that $x \in R_2 - R'_2$. Then there always exists a string $b^x a^m$ such that

$$\delta_D(\langle R_1, R_2\rangle, b^x a^m) \in F_D,$$
$$\delta_D(\langle R'_1, R'_2\rangle, b^x a^m) \notin F_D.$$

3. $R_1 = R'_1 \neq \emptyset, R_2 \neq R'_2$. Let p be an element of R_1 and R'_1. Since $\langle R_1, R_2\rangle$ and $\langle R'_1, R'_2\rangle$ are two different states, according to the definition of D, R_1 and R'_1 cannot be Q_M, otherwise the two states would be the same. Thus, we can find $y \in Q_M - R_1$. We may assume without loss of generality that there exists x such that $x \in R_2 - R'_2$. Then there always exists a string t such that one of $\delta_D(\langle R_1, R_2\rangle, t)$ and $\delta_D(\langle R'_1, R'_2\rangle, t)$ is final and the other is not, where

$$t = \begin{cases} a^{p+1} b^x a^{m-p-1} a^{y+1} a^{m-1}, & \text{if } 0 \notin R'_2, \\ a^m a^y, & \text{if } 0 \notin R_2 \text{ and } 0 \in R'_2, \\ b^x a^{y+1} a^{m-1}, & \text{if } 0 \in R_2 \text{ and } 0 \in R'_2. \end{cases}$$

   Note that when $0 \in R_2$ or $0 \in R'_2$, $m-1$ must be in R_1 and R'_1 according to the definition of D and the condition of $R_1 = R'_1$.

Thus, the states in D are pairwise distinguishable and D is a minimal DFA accepting $(L(M)L(N))^R$ with $3 \cdot 2^{m+n-2} - 2^n + 1$ states. □

The lower bound given in Theorem 3.2 coincides with the upper bound shown in Theorem 3.1 [18]. Thus, the bounds are tight when $m, n \geq 2$.

Next, we consider the state complexity of $(L_1L_2)^R$ when $m = 1$ or $n = 1$. When $m = 1$, L_1 is either Σ^* or $\emptyset$. Clearly,

$$(L_1L_2)^R = \begin{cases} L_2^R\Sigma^*, & \text{if } L_1 = \Sigma^*, \\ \emptyset, & \text{if } L_1 = \emptyset. \end{cases}$$

The state complexity of $L_2^R\Sigma^*$ will be proved later in Theorems 4.5–4.7 and Lemma 4.1 in Section 4. Here we just give the following result on the state complexity of $(L_1L_2)^R$ when $m = 1, n \geq 2$.

Theorem 3.3. *For any integer $n \geq 2$, let L_1 be a 1-state DFA language and L_2 be an n-state DFA language. Then $2^{n-1} + 1$ states are both sufficient and necessary in the worst case for a DFA to accept $(L_1L_2)^R$.*

Note that when $m = 1, n \geq 2$, the general upper bound $3 \cdot 2^{m+n-2} - 2^n + 1 = 2^{n-1} + 1$. Similarly, when $n = 1$, L_2 is either Σ^* or $\emptyset$, and

$$(L_1L_2)^R = \begin{cases} \Sigma^*L_1^R, & \text{if } L_2 = \Sigma^*, \\ \emptyset, & \text{if } L_2 = \emptyset. \end{cases}$$

The state complexity of $\Sigma^*L_1^R$ has been proved in [3]. Thus, we have the following result on the state complexity of $(L_1L_2)^R$ when $m \geq 1, n = 1$.

Theorem 3.4. *For any integer $m \geq 1$, let L_1 be an m-state DFA language and L_2 be a 1-state DFA language. Then 2^{m-1} states are both sufficient and necessary in the worst case for a DFA to accept $(L_1L_2)^R$.*

By summarizing Theorems 3.1–3.3, we can obtain Theorem 3.5.

Theorem 3.5. *For any integers $m \geq 1$, $n \geq 2$, let L_1 be an m-state DFA language and L_2 be an n-state DFA language. Then $3 \cdot 2^{m+n-2} - 2^n + 1$ states are both sufficient and necessary in the worst case for a DFA to accept $(L_1L_2)^R$.*

4. State complexity of $L_1^RL_2$

In this section, we study the state complexity of $L_1^RL_2$ for an m-state DFA language L_1 and an n-state DFA language L_2. We first show that the upper bound of the state complexity of $L_1^RL_2$ is $3 \cdot 2^{m+n-2}$ in general (Theorem 4.1). Then we prove that this upper bound can be reached when $m, n \geq 2$ (Theorem 4.2). Next, we investigate the case when $m = 1$ and $n \geq 1$ and prove the state complexity can be lowered to 2^{n-1} in such a case (Theorem 4.4). Finally, we show that the state complexity of $L_1^RL_2$ is $2^{m-1} + 1$ when $m \geq 2$ and $n = 1$ (Theorem 4.7).

Now, we start with a general upper bound of the state complexity of $L_1^RL_2$ for any integers $m, n \geq 1$.

Theorem 4.1. *Let L_1 and L_2 be two regular languages accepted by an m-state DFA and an n-state DFA, respectively, $m, n \geq 1$. Then there exists a DFA of at most $3 \cdot 2^{m+n-2}$ states that accepts $L_1^RL_2$.*

Proof. Let $M = (Q_M, \Sigma, \delta_M, s_M, F_M)$ be a DFA of m states and $L_1 = L(M)$. Let $N = (Q_N, \Sigma, \delta_N, s_N, F_N)$ be another DFA of n states and $L_2 = L(N)$.

Let $M' = (Q_M, \Sigma, \delta_{M'}, F_M, \{s_M\})$ be an NFA with multiple initial states and $q \in \delta_{M'}(p, a)$ if $\delta_M(q, a) = p$ where $a \in \Sigma$ and $p, q \in Q_M$. Clearly,

$$L(M') = L(M)^R = L_1^R.$$

By performing the subset construction on NFA M', we can get an equivalent, 2^m-state DFA $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ such that $L(A) = L_1^R$. Since M' has only one final state s_M, we know that $F_A = \{I \mid I \subseteq Q_M, s_M \in I\}$. Thus, A has 2^{m-1} final states in total. Now we construct a DFA $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ accepting the language $L_1^RL_2$, where

$$Q_B = \{\langle I, J\rangle \mid I \in Q_A, J \subseteq Q_N\},$$

$$s_B = \begin{cases} \langle s_A, \emptyset\rangle, & \text{if } s_A \notin F_A; \\ \langle s_A, \{s_N\}\rangle, & \text{otherwise}, \end{cases}$$

$$F_B = \{\langle I, J\rangle \in Q_B \mid J \cap F_N \neq \emptyset\},$$

$$\delta_B(\langle I, J\rangle, a) = \begin{cases} \langle I', J'\rangle, \text{ if } \delta_A(I, a) = I', \delta_N(J, a) = J', a \in \Sigma, I' \notin F_A; \\ \langle I', J' \cup \{s_N\}\rangle, \text{ if } \delta_A(I, a) = I', \delta_N(J, a) = J', a \in \Sigma, I' \in F_A. \end{cases}$$

From the above construction, we can see that all the states in B starting with $I \in F_A$ must end with J such that $s_N \in J$. There are in total $2^{m-1} \cdot 2^{n-1}$ states which do not meet this.

Thus, the number of states of the minimal DFA accepting $L_1^RL_2$ is no more than

$$2^{m+n} - 2^{m-1} \cdot 2^{n-1} = 3 \cdot 2^{m+n-2}. \quad \square$$

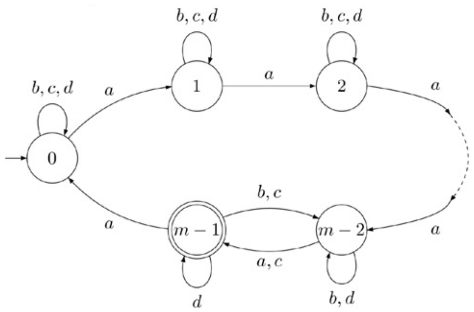

Fig. 3. Witness DFA M which shows that the upper bound of the state complexity of $L(M)^R L(N)$, $\frac{3}{4}2^{m+n}$, is reachable when $m, n \geq 2$.

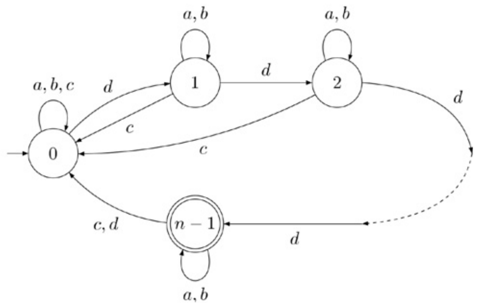

Fig. 4. Witness DFA N which shows that the upper bound of the state complexity of $L(M)^R L(N)$, $\frac{3}{4}2^{m+n}$, is reachable when $m, n \geq 2$.

This result gives an upper bound for the state complexity of $L_1^R L_2$. Next we show that this bound is reachable when $m, n \geq 2$.

Theorem 4.2. *Given two integers $m, n \geq 2$, there exists a DFA M of m states and a DFA N of n states such that any DFA accepting $L(M)^R L(N)$ needs at least $3 \cdot 2^{m+n-2}$ states*

Proof. Let $M = (Q_M, \Sigma, \delta_M, 0, \{m-1\})$ be a DFA, shown in Fig. 3, where $Q_M = \{0, 1, \dots, m-1\}$, $\Sigma = \{a, b, c, d\}$, and the transitions are given as:

- $\delta_M(i, a) = i + 1 \bmod m, i = 0, \dots, m-1$,
- $\delta_M(i, b) = i, i = 0, \dots, m-2, \delta_M(m-1, b) = m-2$,
- $\delta_M(m-2, c) = m-1,\ \delta_M(m-1, c) = m-2$,
  if $m \geq 3, \delta_M(i, c) = i, i = 0, \dots, m-3$,
- $\delta_M(i, d) = i, i = 0, \dots, m-1$,

Note that M is in fact identical with the second witness DFA in the proof of Theorem 3.2 after replacing d by a, a by b, b by c, and c by d.

Let $N = (Q_N, \Sigma, \delta_N, 0, \{n-1\})$ be a DFA, shown in Fig. 4, where $Q_N = \{0, 1, \dots, n-1\}$, $\Sigma = \{a, b, c, d\}$, and the transitions are given as:

- $\delta_N(i, a) = i, i = 0, \dots, n-1$,
- $\delta_N(i, b) = i, i = 0, \dots, n-1$,
- $\delta_N(i, c) = 0, i = 0, \dots, n-1$,
- $\delta_N(i, d) = i + 1 \bmod n, i = 0, \dots, n-1$,

Now we design a DFA $A = (Q_A, \Sigma, \delta_A, \{m-1\}, F_A)$, where $Q_A = \{P \mid P \subseteq Q_M\}$, $\Sigma = \{a, b, c, d\}$, $F_A = \{P \mid 0 \in P, P \in Q_A\}$, and the transitions are defined as:

$$\delta_A(P, e) = \{j \mid \delta_M(j, e) = i, i \in P\}, \quad P \in Q_A, e \in \Sigma.$$

It is easy to see that A is a DFA that accepts $L(M)^R$. Since M is identical with the DFA shown in Fig. 2 by replacing the corresponding letters, and it has been proved in the proof of Theorem 3.2 that any state $\langle \emptyset, R_2 \rangle$ of the resulting DFA is reachable from the initial state, and any two different states $\langle \emptyset, R_2 \rangle$ and $\langle \emptyset, R_2' \rangle$ are distinguishable, then the DFA A constructed in the same manner for $L(M)^R$ in the current proof is minimal.

Now let $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ be another DFA, where

$$Q_B = \{\langle P, Q\rangle \mid P \in Q_A - F_A, Q \subseteq Q_N\} \cup \{\langle P', Q'\rangle \mid P' \in F_A, Q' \subseteq Q_N, 0 \in Q'\},$$
$$\Sigma = \{a, b, c, d\},$$
$$s_B = \langle\{m-1\}, \emptyset\rangle,$$
$$F_B = \{\langle P, Q\rangle \mid n-1 \in Q, \langle P, Q\rangle \in Q_B\},$$

and for each state $\langle P, Q\rangle \in Q_B$ and each letter $e \in \Sigma$,

$$\delta_B(\langle P, Q\rangle, e) = \begin{cases} \langle P', Q'\rangle & \text{if } \delta_A(P, e) = P' \notin F_A, \delta_N(Q, e) = Q', \\ \langle P', Q'\rangle & \text{if } \delta_A(P, e) = P' \in F_A, \delta_N(Q, e) = R', Q' = R' \cup \{0\}. \end{cases}$$

As we mentioned in the last proof, all the states starting with $P \in F_A$ must end with $Q \subseteq Q_N$ such that $0 \in Q$. Clearly, B accepts the language $L(M)^R L(N)$ and it has

$$2^m \cdot 2^n - 2^{m-1} \cdot 2^{n-1} = 3 \cdot 2^{m+n-2}$$

states. Now we show that B is a minimal DFA.

(I) Every state $\langle P, Q\rangle \in Q_B$ is reachable. We consider the following six cases:

1. $P = \emptyset, Q = \emptyset$. $\langle \emptyset, \emptyset\rangle$ is the sink state of B. $\delta_B(\langle\{m-1\}, \emptyset\rangle, b) = \langle P, Q\rangle$.
2. $P \neq \emptyset, Q = \emptyset$. Let $P = \{p_1, p_2, \ldots, p_k\}$, $1 \le p_1 < p_2 < \cdots < p_k \le m-1$, $1 \le k \le m-1$. Note that $0 \notin P$, because $0 \in P$ guarantees $0 \in Q$. $\delta_B(\langle\{m-1\}, \emptyset\rangle, w) = \langle P, Q\rangle$, where
$$w = ab(ac)^{p_2-p_1-1}ab(ac)^{p_3-p_2-1}\cdots ab(ac)^{p_k-p_{k-1}-1}a^{m-1-p_k}.$$
Please note that $w = a^{m-1-p_1}$ when $k = 1$.
3. $P = \emptyset, Q \neq \emptyset$. In this case, let $Q = \{q_1, q_2, \ldots, q_l\}$, $0 \le q_1 < q_2 < \cdots < q_l \le n-1$, $1 \le l \le n$. $\delta_B(\langle\{m-1\}, \emptyset\rangle, x) = \langle P, Q\rangle$, where
$$x = a^m d^{q_l-q_{l-1}} a^m d^{q_{l-1}-q_{l-2}} \cdots a^m d^{q_2-q_1} a^m d^{q_1} b.$$
4. $P \neq \emptyset, 0 \notin P, Q \neq \emptyset$. Let $P = \{p_1, p_2, \ldots, p_k\}$, $1 \le p_1 < p_2 < \cdots < p_k \le m-1$, $1 \le k \le m-1$ and $Q = \{q_1, q_2, \ldots, q_l\}$, $0 \le q_1 < q_2 < \cdots < q_l \le n-1$, $1 \le l \le n$. We can find a string uv such that $\delta_B(\langle\{m-1\}, \emptyset\rangle, uv) = \langle P, Q\rangle$, where
$$u = ab(ac)^{p_2-p_1-1}ab(ac)^{p_3-p_2-1}\cdots ab(ac)^{p_k-p_{k-1}-1}a^{m-1-p_k},$$
$$v = a^m d^{q_l-q_{l-1}} a^m d^{q_{l-1}-q_{l-2}} \cdots a^m d^{q_2-q_1} a^m d^{q_1}.$$
5. $P \neq \emptyset, 0 \in P, m-1 \notin P, Q \neq \emptyset$. Let $P = \{p_1, p_2, \ldots, p_k\}$, $0 = p_1 < p_2 < \cdots < p_k < m-1$, $1 \le k \le m-1$ and $Q = \{q_1, q_2, \ldots, q_l\}$, $0 = q_1 < q_2 < \cdots < q_l \le n-1$, $1 \le l \le n$. Since 0 is in P, according to the definition of B, 0 has to be in Q as well. There exists a string $u'v'$ such that $\delta_B(\langle\{m-1\}, \emptyset\rangle, u'v') = \langle P, Q\rangle$, where
$$u' = ab(ac)^{p_2-p_1-1}ab(ac)^{p_3-p_2-1}\cdots ab(ac)^{p_k-p_{k-1}-1}a^{m-2-p_k},$$
$$v' = a^m d^{q_l-q_{l-1}} a^m d^{q_{l-1}-q_{l-2}} \cdots a^m d^{q_2-q_1} a^m d^{q_1} a.$$
6. $P \neq \emptyset, \{0, m-1\} \subseteq P, Q \neq \emptyset$. Let $P = \{p_1, p_2, \ldots, p_k\}$, $0 = p_1 < p_2 < \cdots < p_k = m-1$, $2 \le k \le m$ and $Q = \{q_1, q_2, \ldots, q_l\}$, $0 = q_1 < q_2 < \cdots < q_l \le n-1$, $1 \le l \le n$. In this case, we have
$$\langle P, Q\rangle = \begin{cases} \delta_B(\langle\{0, 1, p_2+1, \ldots, p_{k-1}+1\}, Q\rangle, a), & \text{if } m-2 \notin P, \\ \delta_B(\langle P - \{m-1\}, Q\rangle, b), & \text{if } m-2 \in P, \end{cases}$$
where states $\langle\{0, 1, p_2+1, \ldots, p_{k-1}+1\}, Q\rangle$ and $\langle P - \{m-1\}, Q\rangle$ have been proved to be reachable in Case 5.

(II) We then show that any two different states $\langle P_1, Q_1\rangle$ and $\langle P_2, Q_2\rangle$ in Q_B are distinguishable.

1. $Q_1 \neq Q_2$. We may assume without loss of generality that there exists x such that $x \in Q_1 - Q_2$. A string d^{n-1-x} can distinguish them because
$$\delta_B(\langle P_1, Q_1\rangle, d^{n-1-x}) \in F_B,$$
$$\delta_B(\langle P_2, Q_2\rangle, d^{n-1-x}) \notin F_B.$$
2. $P_1 \neq P_2, Q_1 = Q_2$. We may assume without loss of generality that there exists y such that $y \in P_1 - P_2$. Then there always exists a string $a^y c^2 d^n$ such that
$$\delta_B(\langle P_1, Q_1\rangle, a^y c^2 d^n) \in F_B,$$
$$\delta_B(\langle P_2, Q_2\rangle, a^y c^2 d^n) \notin F_B.$$

Since all the states in B are reachable and pairwise distinguishable, DFA B is minimal. Thus, any DFA accepting $L(M)^R L(N)$ needs at least $3 \cdot 2^{m+n-2}$ states. $\square$

Theorem 4.2 gives a lower bound for the state complexity of $L_1^R L_2$ when $m, n \geq 2$. It coincides with the upper bound shown in Theorem 4.1 exactly. Thus, we obtain the state complexity of the combined operation $L_1^R L_2$ for $m \geq 2$ and $n \geq 2$.

Theorem 4.3. *For any integers $m, n \geq 2$, let L_1 be an m-state DFA language and L_2 be an n-state DFA language. Then $3 \cdot 2^{m+n-2}$ states are both necessary and sufficient in the worst case for a DFA to accept $L_1^R L_2$.*

In the rest of this section, we study the remaining cases when either $m = 1$ or $n = 1$.

We first consider the case when $m = 1$ and $n \geq 2$. In this case, $L_1 = \emptyset$ or $L_1 = \Sigma^*$. $L_1^R L_2 = L_1 L_2$ holds regardless of whether L_1 is $\emptyset$ or Σ^*, since $\emptyset^R = \emptyset$ and $(\Sigma^*)^R = \Sigma^*$. It has been shown in [25] that 2^{n-1} states are both sufficient and necessary in the worst case for a DFA to accept the catenation of a 1-state DFA language and an n-state DFA language, $n \geq 2$.

When $m = 1$ and $n = 1$, it is also easy to see that 1 state is sufficient and necessary in the worst case for a DFA to accept $L_1^R L_2$, because $L_1^R L_2$ is either $\emptyset$ or Σ^*. Thus, we have Theorem 4.4 concerning the state complexity of $L_1^R L_2$ for $m = 1$ and $n \geq 1$.

Theorem 4.4. *Let L_1 be a 1-state DFA language and L_2 be an n-state DFA language, $n \geq 1$. Then 2^{n-1} states are both sufficient and necessary in the worst case for a DFA to accept $L_1^R L_2$.*

Now, we study the state complexity of $L_1^R L_2$ for $m \geq 2$ and $n = 1$. Let us start with the following upper bound.

Theorem 4.5. *For any integer $m \geq 2$, let L_1 and L_2 be two regular languages accepted by an m-state DFA and a 1-state DFA, respectively. Then there exists a DFA of at most $2^{m-1} + 1$ states that accepts $L_1^R L_2$.*

Proof. Let $M = (Q_M, \Sigma, \delta_M, s_M, F_M)$ be a DFA of m states, $m \geq 2$, k_1 final states and $L_1 = L(M)$. Let N be another DFA of 1 state and $L_2 = L(N)$. Since N is a complete DFA, as we mentioned before, $L(N)$ is either $\emptyset$ or Σ^*. Clearly, $L_1^R \cdot \emptyset = \emptyset$. Thus, we need to consider only the case $L_2 = L(N) = \Sigma^*$.

We construct an NFA $M' = (Q_M, \Sigma, \delta_{M'}, F_M, \{s_M\})$ with k_1 initial states which is similar to the proof of Theorem 4.1. $q \in \delta_{M'}(p, a)$ if $\delta_M(q, a) = p$ where $a \in \Sigma$ and $p, q \in Q_M$. It is easy to see that

$$L(M') = L(M)^R = L_1^R.$$

By performing subset construction on the NFA M', we get an equivalent, 2^m-state DFA $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ such that $L(A) = L_1^R$. $F_A = \{I \mid I \subseteq Q_M, s_M \in I\}$ because M' has only one final state s_M. Thus, A has 2^{m-1} final states in total.

Define $B = (Q_B, \Sigma, \delta_B, s_B, \{f_B\})$ where $f_B \notin Q_A$, $Q_B = (Q_A - F_A) \cup \{f_B\}$,

$$s_B = \begin{cases} s_A & \text{if } s_A \notin F_A, \\ f_B & \text{otherwise} \end{cases}$$

and for any $a \in \Sigma$ and $P \in Q_B$,

$$\delta_B(P, a) = \begin{cases} \delta_A(P, a) & \text{if } \delta_A(P, a) \notin F_A, \\ f_B & \text{if } \delta_A(P, a) \in F_A, \\ f_B & \text{if } P = f_B. \end{cases}$$

The automaton B is exactly the same as A except that A's 2^{m-1} final states are made to be sink states and these sink, final states are merged into one, since they are equivalent. When the computation reaches the final state f_B, it remains there. Now, it is clear that B has

$$2^m - 2^{m-1} + 1 = 2^{m-1} + 1$$

states and $L(B) = L_1^R \Sigma^*$. □

This theorem shows an upper bound for the state complexity of $L_1^R L_2$ for $m \geq 2$ and $n = 1$. The upper bound can also be proved based on the results in [1]. Next we show the upper bound is reachable.

Lemma 4.1. *Given an integer $m = 2$ or 3, there exists an m-state DFA M and a 1-state DFA N such that any DFA accepting $L(M)^R L(N)$ needs at least $2^{m-1} + 1$ states.*

Proof. When $m = 2$ and $n = 1$, we can construct the following witness DFAs. Let $M = (\{0, 1\}, \Sigma, \delta_M, 0, \{1\})$ be a DFA, where $\Sigma = \{a, b\}$, and the transitions are given as:

- $\delta_M(0, a) = 1$, $\delta_M(1, a) = 0$,
- $\delta_M(0, b) = 0$, $\delta_M(1, b) = 0$.

Let N be the DFA accepting Σ^*. Then the resulting DFA for $L(M)^R \Sigma^*$ is $A = (\{0, 1, 2\}, \Sigma, \delta_A, 0, \{1\})$ where

- $\delta_A(0, a) = 1$, $\delta_A(1, a) = 1$, $\delta_A(2, a) = 2$,
- $\delta_A(0, b) = 2$, $\delta_A(1, b) = 1$, $\delta_A(2, b) = 2$.

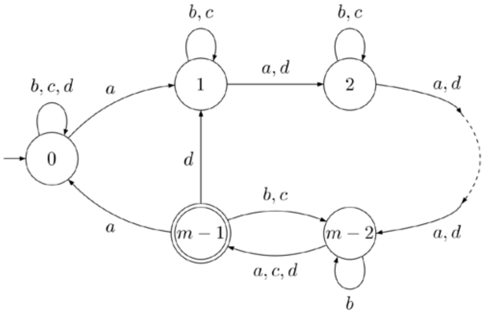

Fig. 5. Witness DFA M which shows that the upper bound of the state complexity of $L(M)^R L(N)$, $2^{m-1}+1$, is reachable when $m \geq 4$ and $n = 1$.

When $m = 3$ and $n = 1$, the witness DFAs are as follows. Let $M' = (\{0, 1, 2\}, \Sigma', \delta_{M'}, 0, \{2\})$ be a DFA, where $\Sigma' = \{a, b, c\}$, and the transitions are:

- $\delta_{M'}(0, a) = 1, \delta_{M'}(1, a) = 2, \delta_{M'}(2, a) = 0,$
- $\delta_{M'}(0, b) = 0, \delta_{M'}(1, b) = 1, \delta_{M'}(2, b) = 1,$
- $\delta_{M'}(0, c) = 0, \delta_{M'}(1, c) = 2, \delta_{M'}(2, c) = 1.$

Let N' be the DFA accepting Σ'^*. The resulting DFA for $L(M')^R \Sigma'^*$ is $A' = (\{0, 1, 2, 3, 4\}, \Sigma', \delta_{A'}, 0, \{3\})$ where

- $\delta_{A'}(0, a) = 1, \delta_{A'}(1, a) = 3, \delta_{A'}(2, a) = 2, \delta_{A'}(3, a) = 3, \delta_{A'}(4, a) = 3,$
- $\delta_{A'}(0, b) = 2, \delta_{A'}(1, b) = 4, \delta_{A'}(2, b) = 2, \delta_{A'}(3, b) = 3, \delta_{A'}(4, b) = 4,$
- $\delta_{A'}(0, c) = 1, \delta_{A'}(1, c) = 0, \delta_{A'}(2, c) = 2, \delta_{A'}(3, c) = 3, \delta_{A'}(4, c) = 4.$

The minimality of A and A' can be easily checked by the reader. □

The above result shows that the bound $2^{m-1} + 1$ is reachable when m is equal to 2 or 3 and $n = 1$. The last case is $m \geq 4$ and $n = 1$.

Theorem 4.6. *Given an integer $m \geq 4$, there exists a DFA M of m states and a DFA N of 1 state such that any DFA accepting $L(M)^R L(N)$ needs at least $2^{m-1} + 1$ states.*

Proof. Let $M = (Q_M, \Sigma, \delta_M, 0, \{m-1\})$ be a DFA, shown in Fig. 5, where $Q_M = \{0, 1, \ldots, m-1\}$, $m \geq 4$, $\Sigma = \{a, b, c, d\}$, and the transitions are given as:

- $\delta_M(i, a) = i + 1 \bmod m, i = 0, \ldots, m-1,$
- $\delta_M(i, b) = i, i = 0, \ldots, m-2, \delta_M(m-1, b) = m-2,$
- $\delta_M(i, c) = i, i = 0, \ldots, m-3, \delta_M(m-2, c) = m-1, \delta_M(m-1, c) = m-2,$
- $\delta_M(0, d) = 0, \delta_M(i, d) = i + 1, i = 1, \ldots, m-2, \delta_M(m-1, d) = 1.$

Let N be the DFA accepting Σ^*. Then $L(M)^R L(N) = L(M)^R \Sigma^*$. Now we design a DFA $A = (Q_A, \Sigma, \delta_A, \{m-1\}, F_A)$ similar to the proof of Theorem 4.2, where $Q_A = \{P \mid P \subseteq Q_M\}$, $\Sigma = \{a, b, c, d\}$, $F_A = \{P \mid 0 \in P, P \in Q_A\}$, and the transitions are defined as:

$$\delta_A(P, e) = \{j \mid \delta_M(j, e) = i, i \in P\}, \quad P \in Q_A, e \in \Sigma.$$

It is easy to see that A is a DFA that accepts $L(M)^R$. Since the transitions of M on letters a, b, and c are exactly the same as those of DFA M in the proof of Theorem 4.2, we can say that A is minimal and it has 2^m states, among which 2^{m-1} states are final.

Define $B = (Q_B, \Sigma, \delta_B, s_B, \{f_B\})$ where $f_B \notin Q_A$, $Q_B = (Q_A - F_A) \cup \{f_B\}$, $s_B = \{m-1\}$, and for any $e \in \Sigma$ and $I \in Q_B$,

$$\delta_B(I, e) = \begin{cases} \delta_A(I, e) & \text{if } \delta_A(I, e) \notin F_A, \\ f_B & \text{if } \delta_A(I, e) \in F_A, \\ f_B & \text{if } I = f_B. \end{cases}$$

The DFA B is the same as A except that A's 2^{m-1} final states are changed into sink states and merged to one sink, final state, as we did in the proof of Theorem 4.5. Clearly, B has $2^m - 2^{m-1} + 1 = 2^{m-1} + 1$ states and $L(B) = L(M)^R \Sigma^*$. Next we show that B is a minimal DFA.

(I) Every state $I \in Q_B$ is reachable from $\{m-1\}$. The proof is similar to that of Theorem 4.2. We consider the following four cases:

1. $I = \emptyset$. $\delta_A(\{m-1\}, b) = I = \emptyset$.
2. $I = f_B$. $\delta_A(\{m-1\}, a^{m-1}) = I = f_B$.
3. $|I| = 1$. Assume that $I = \{i\}$, $1 \leq i \leq m-1$. Note that $i \neq 0$ because all the final states in A have been merged into f_B. In this case, $\delta_A(\{m-1\}, a^{m-1-i}) = I$.

4. $2 \le |I| \le m-1$. Assume that $I = \{i_1, i_2, \ldots, i_k\}$, $1 \le i_1 < i_2 < \ldots < i_k \le m-1$, $2 \le k \le m-1$. $\delta_A(\{m-1\}, w) = I$, where

$$w = ab(ac)^{i_2-i_1-1}ab(ac)^{i_3-i_2-1}\cdots ab(ac)^{i_k-i_{k-1}-1}a^{m-1-i_k}.$$

(II) Any two different states I and J in Q_B are distinguishable.

Since f_B is the only final state in Q_B, it is inequivalent to any other state. Thus, we consider the case when neither of I and J is f_B.

We may assume without loss of generality that there exists x such that $x \in I - J$. x is always greater than 0 because all the states which include 0 have been merged into f_B. Then a string $d^{x-1}a$ can distinguish these two states because

$$\delta_B(I, d^{x-1}a) = f_B,$$
$$\delta_B(J, d^{x-1}a) \ne f_B.$$

Since all the states in B are reachable and pairwise distinguishable, B is a minimal DFA. Thus, any DFA accepting $L(M))^R\Sigma^*$ needs at least $2^{m-1}+1$ states. □

After summarizing Theorem 4.5, Theorem 4.6 and Lemma 4.1, we obtain the state complexity of the combined operation $L_1^R L_2$ for $m \ge 2$ and $n = 1$.

Theorem 4.7. *For any integer $m \ge 2$, let L_1 be an m-state DFA language and L_2 be a 1-state DFA language. Then $2^{m-1}+1$ states are both sufficient and necessary in the worst case for a DFA to accept $L_1^R L_2$.*

5. State complexity of $L_1^* L_2$

In this section, we investigate the state complexity of $L(A)^*L(B)$ for two DFAs A and B of sizes $m, n \ge 1$, respectively. We first notice that, when $n = 1$, the state complexity of $L(A)^*L(B)$ is 1 for any $m \ge 1$. This is because B is complete ($L(B)$ is either $\emptyset$ or Σ^*), and we have either $L(A)^*L(B) = \emptyset$ or $\Sigma^* \subseteq L(A)^*L(B) \subseteq \Sigma^*$. Thus, $L(A)^*L(B)$ is always accepted by a 1 state DFA. Next, we consider the case where A has only one final state, which is also the initial state. In such a case, $L(A)^*$ is also accepted by A, and hence the state complexity of $L(A)^*L(B)$ is equal to that of $L(A)L(B)$. We will show that, for any A of size $m \ge 1$ in this form and any B of size $n \ge 2$, the state complexity of $L(A)L(B)$ (also $L(A)^*L(B)$) is $m(2^n-1)-2^{n-1}+1$ (Theorems 5.1 and 5.2), which is lower than the state complexity of catenation in the general case. Lastly, we consider the state complexity of $L(A)^*L(B)$ in the remaining case, that is when A has at least one final state that is not the initial state and $n \ge 2$. We will show that its upper bound (Theorem 5.3) coincides with its lower bound (Theorem 5.4), and the state complexity is $5 \cdot 2^{m+n-3} - 2^{m-1} - 2^n + 1$.

Now, we consider the case where the DFA A has only one final state, which is also the initial state, and first obtain the following upper bound of the state complexity of $L(A)L(B)$ ($L(A)^*L(B)$), for any DFA B of size $n \ge 2$.

Theorem 5.1. *For integers $m \ge 1$ and $n \ge 2$, let A and B be two DFAs with m and n states, respectively, where A has only one final state, which is also the initial state. Then there exists a DFA of at most $m(2^n-1)-2^{n-1}+1$ states that accepts $L(A)L(B)$, which is equal to $L(A)^*L(B)$.*

Proof. Let $A = (Q_1, \Sigma, \delta_1, s_1, \{s_1\})$ and $B = (Q_2, \Sigma, \delta_2, s_2, F_2)$. We construct a DFA $C = (Q, \Sigma, \delta, s, F)$ such that

$$Q = Q_1 \times (2^{Q_2} - \{\emptyset\}) - \{s_1\} \times (2^{Q_2 - \{s_2\}} - \{\emptyset\}),$$
$$s = \langle s_1, \{s_2\}\rangle,$$
$$F = \{\langle q, T\rangle \in Q \mid T \cap F_2 \ne \emptyset\},$$
$$\delta(\langle q, T\rangle, a) = \langle q', T'\rangle, \text{ for } a \in \Sigma, \text{ where } q' = \delta_1(q, a) \text{ and } T' = R \cup \{s_2\}$$
$$\text{if } q' = s_1, T' = R \text{ otherwise, where } R = \delta_2(T, a).$$

Intuitively, Q contains the pairs whose first component is a state of Q_1 and second component is a subset of Q_2. Since s_1 is the final state of A, without reading any letter, we can enter the initial state of B. Thus, states $\langle q, \emptyset\rangle$ such that $q \in Q_1$ can never be reached in C, because B is complete. Moreover, Q does not contain those states whose first component is s_1 and second component does not contain s_2.

Clearly, C has $m(2^n-1)-2^{n-1}+1$ states, and we can verify that $L(C) = L(A)L(B)$. □

Next, we show that this upper bound can be reached by some witness DFAs in this specific form.

Theorem 5.2. *For any integers $m \ge 1$ and $n \ge 2$, there exist a DFA A of m states and a DFA B of n states, where A has only one final state, which is also the initial state, such that any DFA accepting the language $L(A)L(B)$, which is equal to $L(A)^*L(B)$, needs at least $m(2^n-1)-2^{n-1}+1$ states.*

Proof. When $m = 1$, the witness DFAs used in the proof of Theorem 2.1 in [25] can be used to show that the upper bound proposed in Theorem 5.1 can be reached.

Next, we consider the case when $m \ge 2$. We provide witness DFAs A and B, depicted in Figs. 6 and 7, respectively, over the three letter alphabet $\Sigma = \{a, b, c\}$.

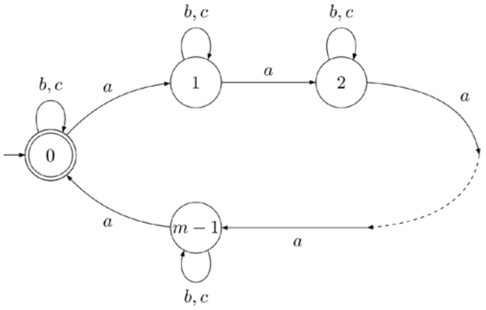

Fig. 6. Witness DFA A which shows that the upper bound of the state complexity of $L(A)^*L(B)$, $m(2^n - 1) - 2^{n-1} + 1$, is reachable when A has only one final state, which is also the initial state, and $m, n \geq 2$.

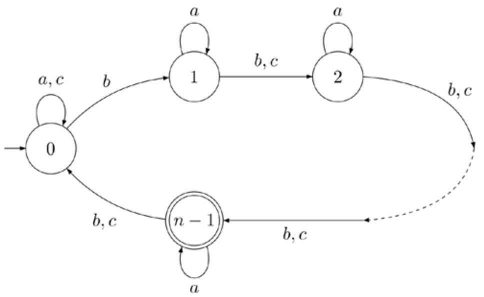

Fig. 7. Witness DFA B which shows that the upper bound of the state complexity of $L(A)^*L(B)$, $m(2^n - 1) - 2^{n-1} + 1$, is reachable, when A has only one final state, which is also the initial state, and $m, n \geq 2$.

A is defined as $A = (Q_1, \Sigma, \delta_1, 0, \{0\})$ where $Q_1 = \{0, 1, \ldots, m-1\}$, and the transitions are given as

- $\delta_1(i, a) = i + 1 \bmod m$, for $i \in Q_1$,
- $\delta_1(i, x) = i$, for $i \in Q_1$, where $x \in \{b, c\}$.

B is defined as $B = (Q_2, \Sigma, \delta_2, 0, \{n-1\})$ where $Q_2 = \{0, 1, \ldots, n-1\}$, where the transitions are given as

- $\delta_2(i, a) = i$, for $i \in Q_2$,
- $\delta_2(i, b) = i + 1 \bmod n$, for $i \in Q_2$,
- $\delta_2(0, c) = 0$, $\delta_2(i, c) = i + 1 \bmod n$, for $i \in \{1, \ldots, n-1\}$.

Following the construction described in the proof of Theorem 5.1, we construct a DFA $C = (Q, \Sigma, \delta, s, F)$ that accepts $L(A)L(B)$ (also $L(A)^*L(B)$). To prove that C is minimal, we show that (I) all the states in Q are reachable from s, and (II) any two different states in Q are not equivalent.

For (I), we show that all the states in $\langle q, T\rangle \in Q$ are reachable by induction on the size of T.

The basis clearly holds, since, for any $i \in Q_1$, the state $\langle i, \{0\}\rangle$ is reachable from $\langle 0, \{0\}\rangle$ by reading string a^i, and the state $\langle i, \{j\}\rangle$ can be reached from the state $\langle i, \{0\}\rangle$ on string b^j, for any $i \in \{1, \ldots, m-1\}$ and $j \in Q_2$.

In the induction steps, we assume that all the states $\langle q, T\rangle$ such that $|T| < k$ are reachable. Then we consider the states $\langle q, T\rangle$ where $|T| = k$. Let $T = \{j_1, j_2, \ldots, j_k\}$ such that $0 \leq j_1 < j_2 << j_k \leq n-1$. We consider the following three cases:

1. $j_1 = 0$ and $j_2 = 1$. For any state $i \in Q_1$, the state $\langle i, T\rangle \in Q$ can be reached as
$$\langle i, \{0, 1, j_3, \ldots, j_k\}\rangle = \delta(\langle 0, \{0, j_3 - 1, \ldots, j_k - 1\}\rangle, ba^i),$$
where $\{0, j_3 - 1, \ldots, j_k - 1\}$ is of size $k - 1$.
2. $j_1 = 0$ and $j_2 > 1$. For any state $i \in Q_1$, the state $\langle i, \{0, j_2, \ldots, j_k\}\rangle$ can be reached from the state $\langle i, \{0, 1, j_3 - j_2 + 1, \ldots, j_k - j_2 + 1\}\rangle$ by reading string $c^{j_2 - 1}$.
3. $j_1 > 0$. In such a case, the first component of the state $\langle q, T\rangle$ cannot be 0. Thus, for any state $i \in \{1, \ldots, m-1\}$, the state $\langle i, \{j_1, j_2, \ldots, j_k\}\rangle$ can be reached from the state $\langle i, \{0, j_2 - j_1, \ldots, j_k - j_1\}\rangle$ by reading string b^{j_1}.

Next, we show that any two distinct states $\langle q, T\rangle$ and $\langle q', T'\rangle$ in Q are not equivalent. We consider the following two cases:

1. $q \neq q'$. Without loss of generality, we assume $q \neq 0$. Then the string $w = c^{n-1}a^{m-q}b^n$ can distinguish the two states, because $\delta(\langle q, T\rangle, w) \in F$ and $\delta(\langle q', T'\rangle, w) \notin F$.
2. $q = q'$ and $T \neq T'$. We may assume without loss of generality that there exists j such that $j \in T - T'$. It is clear that, when $q \neq 0$, string b^{n-1-j} can distinguish the two states, and when $q = 0$, string c^{n-1-j} can distinguish the two states since j cannot be 0.

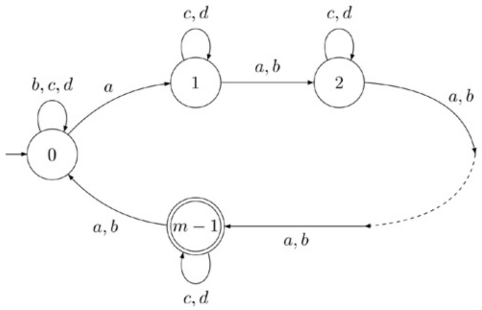

Fig. 8. Witness DFA A which shows that the upper bound of the state complexity of $L(A)^*L(B)$, $5 \cdot 2^{m+n-3} - 2^{m-1} - 2^n + 1$, is reachable when $m, n \geq 2$.

Due to (I) and (II), the DFA C needs at least $m(2^n - 1) - 2^{n-1} + 1$ states and is minimal. □

In the rest of this section, we focus on the case where the DFA A contains at least one final state that is not the initial state. Thus, this DFA is of size at least 2. We first obtain the following upper bound for the state complexity.

Theorem 5.3. *Let $A = (Q_1, \Sigma, \delta_1, s_1, F_1)$ be a DFA such that $|Q_1| = m > 1$ and $|F_1 - \{s_1\}| = k_1 \geq 1$, and $B = (Q_2, \Sigma, \delta_2, s_2, F_2)$ be a DFA such that $|Q_2| = n > 1$. Then there exists a DFA of at most $2^{m+n-2} + 3 \cdot 2^{m+n-k_1-2} - 2^{m-k_1} - 2^n + 1$ states that accepts $L(A)^*L(B)$.*

Proof. We denote $F_1 - \{s_1\}$ by F_0. Then $|F_0| = k_1 \geq 1$.

We construct a DFA $C = (Q, \Sigma, \delta, s, F)$ for the language $L_1^*L_2$, where L_1 and L_2 are the languages accepted by DFAs A and B, respectively.

Let $Q = \{\langle p, t\rangle \mid p \in P \text{ and } t \in T\} - \{\langle p', t'\rangle \mid p' \in P' \text{ and } t' \in T'\}$, where

$$P = \{R \mid R \subseteq (Q_1 - F_0) \text{ and } R \neq \emptyset\} \cup P',$$
$$T = 2^{Q_2} - \{\emptyset\},$$
$$P' = \{R \mid R \subseteq Q_1, s_1 \in R, \text{ and } R \cap F_0 \neq \emptyset\},$$
$$T' = 2^{Q_2 - \{s_2\}} - \{\emptyset\}.$$

The initial state s is $s = \langle \{s_1\}, \{s_2\}\rangle$.

The set of final states is defined to be $F = \{\langle p, t\rangle \in Q \mid t \cap F_2 \neq \emptyset\}$.

The transition relation δ is defined as follows:

$$\delta(\langle p, t\rangle, a) = \begin{cases} \langle p', t'\rangle & \text{if } p' \cap F_1 = \emptyset, \\ \langle p' \cup \{s_1\}, t' \cup \{s_2\}\rangle & \text{otherwise,} \end{cases}$$

where, $a \in \Sigma$, $p' = \delta_1(p, a)$, and $t' = \delta_2(t, a)$.

Intuitively, C is equivalent to the NFA C' obtained by first constructing an NFA A' that accepts L_1^*, then catenating this new NFA with DFA B by λ-transitions. Note that, in the construction of A', we need to add a new initial and final state s_1'. However, this new state does not appear in the first component of any of the states in Q. The reason is as follows. First, note that this new state does not have any incoming transitions. Thus, from the initial state s_1' of A', after reading a nonempty string, we will never return to this state. As a result, states $\langle p, t\rangle$ such that $p \subseteq Q_1 \cup \{s_1'\}$, $s_1' \in p$, and $t \in 2^{Q_2}$ is never reached in DFA C except for the state $\langle \{s_1'\}, \{s_2\}\rangle$. Then we note that in the construction of A', states s_1' and s_1 should reach the same state on any letter in Σ. Thus, we can say that states $\langle \{s_1'\}, \{s_2\}\rangle$ and $\langle \{s_1\}, \{s_2\}\rangle$ are equivalent, because neither of them is final if $s_2 \notin F_2$, and they are both final states otherwise. Hence, we merge this two states and let $\langle \{s_1\}, \{s_2\}\rangle$ be the initial state of C.

Also, we notice that states $\langle p, \emptyset\rangle$ such that $p \in P$ can never be reached in C, because B is complete.

Moreover, C does not contain those states whose first component contains a final state of A and whose second component does not contain the initial state of B.

Therefore, we can verify that DFA C indeed accepts $L_1^*L_2$, and it is clear that the size of the state set of C is

$$\begin{aligned} |Q| &= (2^{m-1} + 2^{m-1-k_1} - 1)(2^n - 1) - (2^{m-1} - 2^{m-k_1-1})(2^{n-1} - 1) \\ &= 2^{m+n-2} + 3 \cdot 2^{m+n-k_1-2} - 2^{m-k_1} - 2^n + 1. \quad \square \end{aligned}$$

Then we show that this upper bound is reachable by some witness DFAs.

Theorem 5.4. *For any integers $m, n \geq 2$, there exist a DFA A of m states and a DFA B of n states such that any DFA accepting $L(A)^*L(B)$ needs at least $5 \cdot 2^{m+n-3} - 2^{m-1} - 2^n + 1$ states.*

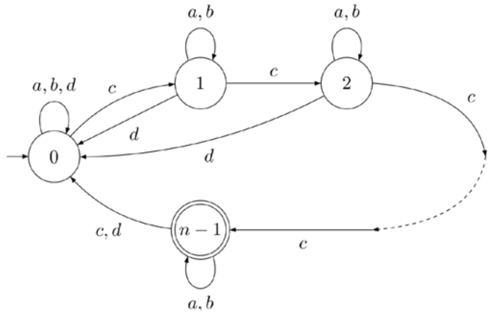

Fig. 9. Witness DFA B which shows that the upper bound of the state complexity of $L(A)^*L(B)$, $5 \cdot 2^{m+n-3} - 2^{m-1} - 2^n + 1$, is reachable when $m, n \geq 2$.

Proof. We define the following two automata over a four letter alphabet $\Sigma = \{a, b, c, d\}$.

Let $A = (Q_1, \Sigma, \delta_1, 0, \{m-1\})$, shown in Fig. 8, where $Q_1 = \{0, 1, \ldots, m-1\}$, and the transitions are defined as

- $\delta_1(i, a) = i + 1 \bmod m$, for $i \in Q_1$,
- $\delta_1(0, b) = 0$, $\delta_1(i, b) = i + 1 \bmod m$, for $i \in \{1, \ldots, m-1\}$,
- $\delta_1(i, x) = i$, for $i \in Q_1$, $x \in \{c, d\}$.

Let $B = (Q_2, \Sigma, \delta_2, 0, \{n-1\})$, shown in Fig. 9, where $Q_2 = \{0, 1, \ldots, n-1\}$, and the transitions are defined as

- $\delta_2(i, x) = i$, for $i \in Q_2$, $x \in \{a, b\}$,
- $\delta_2(i, c) = i + 1 \bmod n$, for $i \in Q_2$,
- $\delta_2(i, d) = 0$, for $i \in Q_2$.

Let $C = \{Q, \Sigma, \delta, \langle\{0\}, \{0\}\rangle, F\}$ be the DFA accepting the language $L(A)^*L(B)$ which is constructed from A and B exactly as described in the proof of Theorem 5.3.

Now, we prove that the size of Q is minimal by showing that (I) any state in Q can be reached from the initial state, and (II) no two different states in Q are equivalent.

We first prove (I) by induction on the size of the second component t of the states in Q.

The basis holds, since, for any $i \in Q_2$, the state $\langle\{0\}, \{i\}\rangle$ can be reached from the initial state $\langle\{0\}, \{0\}\rangle$ on the string c^i. In the proof of Theorem 3.3 in [25], a witness DFA is used to prove the state complexity of star operation on regular languages. The DFA A above is a modification of that witness DFA by adding $c-$ and $d-$ loops to each state. With similar construction of the resulting DFA for star, it has been proved in [25] that any $p \in P$ is reachable from $\{0\}$ on some string over letters a and b. Since $a-$ and $b-$ transitions do not change the second element $\{i\}$ in the state, it is clear that the state $\langle p, \{i\}\rangle$ of Q, where $p \in P$ and $i \in Q_2$, is reachable from the state $\langle\{0\}, \{i\}\rangle$ on the same string.

In the induction steps, assume that all the states $\langle p, t\rangle$ in Q such that $p \in P$ and $|t| < k$ are reachable. Then we consider the states $\langle p, t\rangle$ in Q where $p \in P$ and $|t| = k$. Let $t = \{j_1, j_2, \ldots, j_k\}$ such that $0 \leq j_1 < j_2 < \cdots < j_k \leq n-1$.

Note that states such that $p = \{0\}$ and $j_1 = 0$ are reachable as follows:

$$\langle\{0\}, \{0, j_2, \ldots, j_k\}\rangle = \delta(\langle\{0\}, \{0, j_3 - j_2, \ldots, j_k - j_2\}\rangle, c^{j_2}a^{m-1}b).$$

Then states such that $p = \{0\}$ and $j_1 > 0$ can be reached as follows:

$$\langle\{0\}, \{j_1, j_2, \ldots, j_k\}\rangle = \delta(\langle\{0\}, \{0, j_2 - j_1, \ldots, j_k - j_1\}\rangle, c^{j_1}).$$

Once again, with the same strings over letters a and b in the proof of Theorem 3.3 in [25], states $\langle p, t\rangle$ in Q, where $p \in P$ and $|t| = k$, can be reached from the state $\langle\{0\}, t\rangle$.

Next, we show that any two states in Q are not equivalent. Let $\langle p, t\rangle$ and $\langle p', t'\rangle$ be two different states in Q. We consider the following two cases:

1. $p \neq p'$. We may assume without loss of generality that there exists i such that $i \in p - p'$. It is clear that string $a^{m-1-i}dc^n$ is accepted by C starting from the state $\langle p, t\rangle$, but it is not accepted starting from the state $\langle p', t'\rangle$.
2. $p = p'$ and $t \neq t'$. We may assume without loss of generality that there exists j such that $j \in t - t'$. Then the state $\langle p, t\rangle$ reaches a final state on string c^{n-1-j}, but the state $\langle p', t'\rangle$ does not on the same string. Note that, when $m - 1 \in p$, we can say that $j \neq 0$.

Due to (I) and (II), DFA C has at least $5 \cdot 2^{m+n-3} - 2^{m-1} - 2^n + 1$ reachable states, and any two of them are not equivalent. □

6. State complexity of $(L_1 \cup L_2)L_3$

In this section, we study the state complexity of $(L_1 \cup L_2)L_3$, where L_1, L_2 and L_3 are regular languages accepted by DFAs of m, n, p states, respectively. We first show that the state complexity of $(L_1 \cup L_2)L_3$ is $mn2^p - (m + n - 1)2^{p-1}$ when

$m, n, p \geq 2$ (Theorem 6.1). Next, we investigate the case when $m = 1$ or $n = 1$ and $p \geq 2$ and show that the state complexity is $mn2^p - 2^{p-1}$ in such a case (Theorem 6.2). Then we prove that the state complexity of $(L_1 \cup L_2)L_3$ is mn when $m = 1$ or $n = 1$ and $p = 1$ (Theorem 6.3). Finally, we show that the state complexity of $(L_1 \cup L_2)L_3$ is $mn - m - n + 2$ when $m, n \geq 2$ and $p = 1$ (Theorem 6.4).

Now let us start with the state complexity of $(L_1 \cup L_2)L_3$ for any integers $m, n, p \geq 2$.

Theorem 6.1. *Let L_1, L_2 and L_3 be three regular languages accepted by an m-state DFA, an n-state DFA and a p-state DFA, respectively, $m, n, p \geq 2$. Then $mn2^p - (m + n - 1)2^{p-1}$ states are sufficient and necessary in the worst case for a DFA to accept $(L_1 \cup L_2)L_3$.*

Proof. We first show that $mn2^p - (m+n-1)2^{p-1}$ states are sufficient. It has been proved in [25] that the state complexity of $L(U)L(V)$ is upper bounded by $u2^v - k2^{v-1}$, where U and V are u-state and v-state automata, respectively, and U has k final states. Thus, the state complexity of $(L_1 \cup L_2)L_3$ is no more than $mn2^p - k'2^{p-1}$ by the mathematical composition of the state complexity of union and catenation, where k' is the number of final states in the DFA accepting $L_1 \cup L_2$. We can easily get the upper bound $mn2^p - (m + n - 1)2^{p-1}$ when the DFAs for L_1 and L_2 both have a single final state. Note that in the minimal, complete DFA for arbitrary $L_1 \cup L_2$, the number of final states k' may be less than $(m + n - 1)$. However, it is clear that

$$(mn - (m + n - 1) + k')2^p - k'2^{p-1} \leq mn2^p - (m + n - 1)2^{p-1}.$$

Now let us prove that $mn2^p - (m+n-1)2^{p-1}$ states are necessary in the worst case. Let $A = (Q_A, \Sigma, \delta_A, 0, \{m-1\})$ be a DFA, where $Q_A = \{0, 1, \ldots, m-1\}$, $\Sigma = \{a, b, c, d\}$, and the transitions are given as:

- $\delta_A(i, a) = i + 1 \bmod m, i = 0, \ldots, m-1$,
- $\delta_A(i, e) = i, i = 0, \ldots, m-1, e \in \{b, c, d\}$.

Let $B = (Q_B, \Sigma, \delta_B, 0, \{n-1\})$ be a DFA, where $Q_B = \{0, 1, \ldots, n-1\}$, $\Sigma = \{a, b, c, d\}$, and the transitions are given as:

- $\delta_B(i, e) = i, i = 0, \ldots, n-1, e \in \{a, c, d\}$,
- $\delta_B(i, b) = i + 1 \bmod n, i = 0, \ldots, n-1$.

Let $C = (Q_C, \Sigma, \delta_C, 0, \{p-1\})$ be a DFA, where $Q_C = \{0, 1, \ldots, p-1\}$, $\Sigma = \{a, b, c, d\}$, and the transitions are given as:

- $\delta_C(i, e) = i, i = 0, \ldots, p-1, e \in \{a, b\}$,
- $\delta_C(i, c) = i + 1 \bmod p, i = 0, \ldots, p-1$,
- $\delta_C(i, d) = 1, i = 0, \ldots, p-1$.

Next we construct a DFA $D = (Q_D, \Sigma, \delta_D, s_D, F_D)$, where

$$
\begin{aligned}
&Q_D = M \cup N \cup P,\\
&M = \{\langle i, j, K\rangle \mid i \in Q_A - \{m-1\}, j \in Q_B - \{n-1\}, K \subseteq Q_C\},\\
&N = \{\langle i, j, K\rangle \mid i = m-1, j \in Q_B, K \subseteq Q_C, 0 \in K\},\\
&P = \{\langle i, j, K\rangle \mid i \in Q_A, j = n-1, K \subseteq Q_C, 0 \in K\},\\
&s_D = \langle 0, 0, \emptyset\rangle,\\
&F_D = \{\langle i, j, K\rangle \in Q_D \mid p-1 \in K\},
\end{aligned}
$$

and for any $g = \langle i, j, K\rangle \in Q_D$, $a \in \Sigma$, $\delta_D(g, a) = \langle i', j', K'\rangle$, where

- if $\delta_A(i, a) = i' \neq m-1$ and $\delta_B(j, a) = j' \neq n-1$, then $\delta_C(K, a) = K'$,
- if $\delta_A(i, a) = i' = m-1$ and $\delta_B(j, a) = j'$, then $K' = \delta_C(K, a) \cup \{0\}$,
- if $\delta_A(i, a) = i'$ and $\delta_B(j, a) = j' = n-1$, then $K' = \delta_C(K, a) \cup \{0\}$.

Clearly, D accepts $(L(A) \cup L(B))L(C)$. We will prove D is a minimal DFA in the following.

(I) We first show that every state $\langle i, j, K\rangle \in Q_D$, is reachable from s_D by induction on the size of K.

When $|K| = 0$, we can see $i \neq m-1$ and $j \neq n-1$ according to the definition of D. The state $\langle i, j, \emptyset\rangle$ is reachable from s_D by reading $a^i b^j$. When $|K| = 1$, let K be $\{k_1\}$, $0 \leq k_1 \leq p-1$. We have $\delta_D(s_D, a^m c^{k_1} a^i b^j) = \langle i, j, K\rangle$. Note that if $i = m-1$ or $j = n-1$, then K has to be $\{0\}$ in this case.

Assume that any state $\langle i', j', K'\rangle \in Q_D$ such that $|K'| = q \geq 1$ is reachable from s_D. We will prove that $\langle i, j, K\rangle \in Q_D$ such that $|K| = q + 1$ is reachable in the following. Let $K = \{l_1, l_2, \ldots, l_{q+1}\}$ and $K' = \{l_2 - l_1, \ldots, l_{q+1} - l_1\}$, where $0 \leq l_1 < l_2 < \cdots < l_{q+1} \leq p-1$. Then

$$\delta_D(\langle 0, 0, K'\rangle, a^m c^{l_1} a^i b^j) = \langle i, j, K\rangle.$$

Since $|K'| = q$ and $\langle 0, 0, K'\rangle$ is reachable from s_D according to the induction hypothesis, the state $\langle i, j, K\rangle$ is also reachable. As we mentioned, if $i = m-1$ or $j = n-1$, then l_1 has to be 0. Thus, we have proved every state $\langle i, j, K\rangle \in Q_D$, can be reached from s_D.

(II) Next, we show that any two different states $\langle i_1, j_1, K_1\rangle, \langle i_2, j_2, K_2\rangle \in Q_D$, are distinguishable. We consider the following three cases.

1. $K_1 \neq K_2$. We may assume without loss of generality that there exists x such that $x \in K_1 - K_2$. A string c^{p-1-x} can distinguish the two states because

$$\delta_D(\langle i_1, j_1, K_1\rangle, c^{p-1-x}) \in F_D,$$
$$\delta_D(\langle i_2, j_2, K_2\rangle, c^{p-1-x}) \notin F_D.$$

2. $i_1 \neq i_2, K_1 = K_2$. Without loss of generality, we assume that $i_1 > i_2$. Then there always exists a string $b^{n-j_2}da^{m-1-i_1}c^{p-1}$ such that

$$\delta_D(\langle i_1, j_1, K_1\rangle, b^{n-j_2}da^{m-1-i_1}c^{p-1}) \in F_D,$$
$$\delta_D(\langle i_2, j_2, K_2\rangle, b^{n-j_2}da^{m-1-i_1}c^{p-1}) \notin F_D.$$

3. $i_1 = i_2, j_1 \neq j_2, K_1 = K_2$. Without loss of generality, we assume $j_1 > j_2$ in this case. Then we can distinguish the two states with $a^{m-i_1}db^{n-1-j_1}c^{p-1}$ because

$$\delta_D(\langle i_1, j_1, K_1\rangle, a^{m-i_1}db^{n-1-j_1}c^{p-1}) \in F_D,$$
$$\delta_D(\langle i_2, j_2, K_2\rangle, a^{m-i_1}db^{n-1-j_1}c^{p-1}) \notin F_D.$$

Thus, the states in D are pairwise distinguishable and D is a minimal DFA accepting $(L(A) \cup L(B))L(C)$ with $mn2^p - (m+n-1)2^{p-1}$ states. □

Nest, we consider the case when $m = 1$ or $n = 1$, and $p \geq 2$. When $m = 1, n \geq 2, p \geq 2$, the resulting language $(L_1 \cup L_2)L_3$ is either Σ^*L_3 or L_2L_3 whose state complexities are 2^{p-1} and $n2^p - 2^{p-1}$, respectively [25]. Clearly, the state complexity of $(L_1 \cup L_2)L_3$ should be the latter one. When $m \geq 2, n = 1, p \geq 2$, the case is symmetric and the state complexity is $m2^p - 2^{p-1}$. When $m = n = 1, n \geq 2$, $(L_1 \cup L_2)L_3$ is either Σ^*L_3 or $\emptyset$ and the state complexity is 2^{p-1}. Thus, we can get Theorem 6.2.

Theorem 6.2. *Let L_1, L_2 and L_3 be three regular languages accepted by an m-state DFA, an n-state DFA and a p-state DFA, respectively, with $m = 1$ or $n = 1$, and $p \geq 2$. Then $mn2^p - 2^{p-1}$ states are sufficient and necessary in the worst case for a DFA to accept $(L_1 \cup L_2)L_3$.*

Now let us investigate the case when $p = 1$. In this case, the language L_3 is either Σ^* or $\emptyset$. In [25], it has been proved that the state complexity of $L_1\Sigma^*$ is m. Therefore, the mathematical composition of the state complexities of union and catenation for $(L_1 \cup L_2)L_3$ when $p = 1$ is mn. This upper bound is reachable when $m = 1$ or $n = 1$, and $p = 1$, because

$$(L_1 \cup L_2)\Sigma^* = \begin{cases} L_1\Sigma^*, & \text{if } m \geq 2, n = 1, L_2 = \emptyset, \\ L_2\Sigma^*, & \text{if } m = 1, L_1 = \emptyset, n \geq 2, \\ \Sigma^*, & \text{if } m = n = 1, L_1 = \Sigma^* \text{ or } L_2 = \Sigma^*. \end{cases}$$

Thus, Theorem 6.3 in the following holds.

Theorem 6.3. *Let L_1, L_2 and L_3 be three regular languages accepted by an m-state DFA, an n-state DFA and a 1-state DFA, respectively, $m = 1$ or $n = 1$. Then mn states are sufficient and necessary in the worst case for a DFA to accept $(L_1 \cup L_2)L_3$.*

Now the only case left is $m, n \geq 2$ and $p = 1$. The upper bound can be lowered in this case, because the multiple final states in the resulting DFA for $L_1 \cup L_2$ are merged to one sink, final state to accept $(L_1 \cup L_2)\Sigma^*$. There are $m+n-1$ such final states in the worst case. Thus, the upper bound is $mn-m-n+2$ in this case and it is easy to see that $L_1 = \{w \in \{a, b\}^* \mid |w|_a \equiv m-1 \mod m\}, L_2 = \{w \in \{a, b\}^* \mid |w|_b \equiv n-1 \mod n\}$, and $L_3 = \{a, b\}^*$ are the witness regular languages that reach the upper bound.

Theorem 6.4. *Let L_1, L_2 and L_3 be three regular languages accepted by an m-state DFA, an n-state DFA and a 1-state DFA, respectively, $m, n \geq 2$. Then $mn - m - n + 2$ states are sufficient and necessary in the worst case for a DFA to accept $(L_1 \cup L_2)L_3$.*

7. State complexity of $(L_1 \cap L_2)L_3$

In this section, we investigate the state complexity of $(L_1 \cap L_2)L_3$, where L_1, L_2 and L_3 are regular languages accepted by DFAs of m, n, p states, respectively. We first show that the state complexity of $(L_1 \cap L_2)L_3$ is $mn2^p - 2^{p-1}$ when $m, n \geq 1$, $p \geq 2$ (Theorem 7.1). Next, we prove the case when $m, n \geq 1, p = 1$ and show that the state complexity is mn in this case (Theorem 7.2).

Let us start with the state complexity of $(L_1 \cap L_2)L_3$ for any integers $m, n \geq 1, p \geq 2$.

Theorem 7.1. *Let L_1, L_2 and L_3 be three regular languages accepted by an m-state DFA, an n-state DFA and a p-state DFA, respectively, $m, n \geq 1$, $p \geq 2$. Then $mn2^p - 2^{p-1}$ states are sufficient and necessary in the worst case for a DFA to accept $(L_1 \cap L_2)L_3$.*

Proof. The state complexity of $(L_1 \cap L_2)L_3$ is upper bounded by $mn2^p - 2^{p-1}$ because it is the mathematical composition of the state complexities of intersection and catenation [25]. Thus, we only need to prove that $mn2^p - 2^{p-1}$ states are necessary in the worst case. When $m = 1$ and $p \geq 2$, $(L_1 \cap L_2)L_3$ is either L_2L_3 or $\emptyset$. The state complexity of L_2L_3 is $n2^p - 2^{p-1}$ [25] which coincides with the upper bound we obtained. The case when $n = 1$ and $p \geq 2$ is symmetric.

When $m, n, p \geq 2$, we use the same witness DFAs A, B and C in the proof of Theorem 6.1. Next we construct a DFA $D = (Q_D, \Sigma, \delta_D, s_D, F_D)$, where

$$Q_D = M - N,$$
$$M = \{\langle i, j, K\rangle \mid i \in Q_A, j \in Q_B, K \subseteq Q_C\},$$
$$N = \{\langle i, j, K\rangle \mid i = m-1, j = n-1, K \subseteq Q_C - \{0\}\},$$
$$s_D = \langle 0, 0, \emptyset\rangle,$$
$$F_D = \{\langle i, j, K\rangle \in Q_D \mid p-1 \in K\},$$

and for any $g = \langle i, j, K\rangle \in Q_D$, $a \in \Sigma$, δ_D is defined as follows,

$$\delta_D(g, a) = \begin{cases} \langle \delta_A(i, a), \delta_B(j, a), \delta_C(K, a) \cup \{0\}\rangle, & \text{if } \delta_A(i, a) = m-1 \text{ and } \delta_B(j, a) = n-1, \\ \langle \delta_A(i, a), \delta_B(j, a), \delta_C(K, a)\rangle, & \text{otherwise.} \end{cases}$$

It is easy to see that D accepts $(L(A) \cap L(B))L(C)$. In the following, we will show D is minimal with a similar method as in the proof of Theorem 6.1.

(I) First, we prove that any state $\langle i, j, K\rangle \in Q_D$ can be reached from s_D by induction on the size of K.

When $|K| = 0$, we have $i \neq m-1$ or $j \neq n-1$ according to the definition of D. The state $\langle i, j, \emptyset\rangle$ can be reached from s_D by a^ib^j. When $|K| = 1$, let $K = \{k_1\}$, $0 \leq k_1 \leq p-1$. Then $\delta_D(s_D, a^{m-1}b^{n-1}abc^{k_1}a^ib^j) = \langle i, j, K\rangle$. If $i = m-1$ and $j = n-1$, K must be $\{0\}$ when $|K| = 1$.

Assume any state $\langle i', j', K'\rangle \in Q_D$ such that $|K'| = q \geq 1$ can be reached from s_D. In the following we will prove $\langle i, j, K\rangle \in Q_D$ such that $|K| = q+1$ is also reachable. Let $K = \{l_1, l_2, \ldots, l_{q+1}\}$ and $K' = \{l_2 - l_1, \ldots, l_{q+1} - l_1\}$, where $0 \leq l_1 < l_2 < \cdots < l_{q+1} \leq p-1$. Then

$$\delta_D(\langle 0, 0, K'\rangle, a^{m-1}b^{n-1}abc^{l_1}a^ib^j) = \langle i, j, K\rangle.$$

Since $\langle 0, 0, K'\rangle$ where $|K'| = p$ is reachable as the induction hypothesis, the state $\langle i, j, K\rangle$ is also reachable. Again, if $i = m-1$ and $j = n-1$, l_1 must be 0. Thus, all states in D are reachable from s_D.

(II) Next, we prove that any two different states $\langle i_1, j_1, K_1\rangle$ and $\langle i_2, j_2, K_2\rangle$ in Q_D, are distinguishable. There are three cases to be considered.

1. $K_1 \neq K_2$. Without loss of generality, we may assume that there exists x such that $x \in K_1 - K_2$ and a string c^{p-1-x} distinguishes the two states because

$$\delta_D(\langle i_1, j_1, K_1\rangle, c^{p-1-x}) \in F_D,$$
$$\delta_D(\langle i_2, j_2, K_2\rangle, c^{p-1-x}) \notin F_D.$$

2. $i_1 \neq i_2$, $K_1 = K_2$. Without loss of generality, we may assume $i_1 > i_2$. Then there exists a string $b^{n-1-j_1}da^{m-1-i_1}c^{p-1}$ such that

$$\delta_D(\langle i_1, j_1, K_1\rangle, b^{n-1-j_1}da^{m-1-i_1}c^{p-1}) \in F_D,$$
$$\delta_D(\langle i_2, j_2, K_2\rangle, b^{n-1-j_1}da^{m-1-i_1}c^{p-1}) \notin F_D.$$

3. $i_1 = i_2$, $j_1 \neq j_2$, $K_1 = K_2$. Without loss of generality, assume that $j_1 > j_2$. Then the two states can be distinguished by $a^{m-1-i_1}db^{n-1-j_1}c^{p-1}$ because

$$\delta_D(\langle i_1, j_1, K_1\rangle, a^{m-1-i_1}db^{n-1-j_1}c^{p-1}) \in F_D,$$
$$\delta_D(\langle i_2, j_2, K_2\rangle, a^{m-1-i_1}db^{n-1-j_1}c^{p-1}) \notin F_D.$$

Thus, all states in D are distinguishable and D is a minimal DFA for $(L(A) \cap L(B))L(C)$ with $mn2^p - 2^{p-1}$ states. □

Next, we consider the case when $m, n \geq 1$ and $p = 1$. Since L_3 is accepted by a 1-state DFA, it is either $\emptyset$ or Σ^*. When $L_3 = \emptyset$, $(L_1 \cap L_2)L_3$ is also $\emptyset$. When $L_3 = \Sigma^*$, we have $(L_1 \cap L_2)L_3 = (L_1 \cap L_2)\Sigma^*$. As we mentioned in the previous section, the state complexity of $L_1\Sigma^*$ is m [25]. Thus, the state complexity of $(L_1 \cap L_2)\Sigma^*$ is upper bounded by mn and the reader can easily prove that the upper bound is reached by $L_1 = \{w \in \{a,b\}^* \mid |w|_a \equiv m-1 \mod m\}$ and $L_2 = \{w \in \{a,b\}^* \mid |w|_b \equiv n-1 \mod n\}$ when $m, n \geq 2$. For $m = 1$ or $n = 1$, and $p = 1$, we have

$$(L_1 \cap L_2)\Sigma^* = \begin{cases} L_1\Sigma^*, & \text{if } m \geq 2, n = 1, L_2 = \Sigma^*, \\ L_2\Sigma^*, & \text{if } m = 1, L_1 = \Sigma^*, n \geq 2, \\ \Sigma^*, & \text{if } m = n = 1, L_1 = L_2 = \Sigma^*. \end{cases}$$

Thus, we can get Theorem 7.2 after summarizing the subcases above.

Theorem 7.2. *Let L_1, L_2 and L_3 be three regular languages accepted by an m-state DFA, an n-state DFA and a 1-state DFA, respectively, $m, n \geq 1$. Then mn states are sufficient and necessary in the worst case for a DFA to accept $(L_1 \cap L_2)L_3$.*

8. State complexity of $L_1L_2 \cap L_3$

In this section, we investigate the state complexity of $L_1L_2 \cap L_3$ for regular languages L_1, L_2, and L_3 accepted by m-state, n-state, and p-state DFAs, respectively. It is clear that, when $p = 1$, L_3 can only be either Σ^* or $\emptyset$. We do not need to consider the case $L_3 = \emptyset$. Thus, $L_1L_2 \cap L_3 = L_1L_2$. Therefore, when $p = 1$, the state complexity of $L_1L_2 \cap L_3$ is equal to that of L_1L_2. In the following theorem, we show that the state complexity of $L_1L_2 \cap L_3$ is $(m2^n - 2^{n-1})p$ when $m \geq 1$, $n \geq 2$, and $p \geq 2$, and it is mp when $m \geq 1$, $n = 1$, and $p \geq 2$.

Theorem 8.1. *Let L_1, L_2, and L_3 be languages accepted by m-state, n-state, and p-state DFAs, respectively, then, we have:*

(1) *when $m \geq 1$, $n \geq 2$, and $p \geq 2$, the state complexity of $L_1L_2 \cap L_3$ is $(m2^n - 2^{n-1})p$.*
(2) *when $m \geq 1$, $n = 1$, and $p \geq 2$, the state complexity of $L_1L_2 \cap L_3$ is mp.*

Proof. For (1), Denote by A, B, and C the m-state, n-state, and p-state DFAs, respectively. Since the claimed state complexity is exactly the composition of the state complexities of catenation and intersection, the construction of a DFA E that accepts $L_1L_2 \cup L_3$ is as follows. We first construct a DFA D that accepts L_1L_2. Then, the set of the states of E is a Cartesian product of the sets of the states of D and C, the initial state of E is a pair of the initial states of D and C, and each final state of E consists of a final state of D and a final state of C. Moreover, the transitions of E simulate the transitions of D and C on the first element and the second element of each state of E, respectively. Since the state complexity of L_1L_2 is $m2^n - 2^{n-1}$ when $m \geq 1$ and $n \geq 2$, the total number of states in E is upper bounded by $(m2^n - 2^{n-1})p$.

To prove (1), we just need to show that this upper bound can be reached by some witness DFAs.

We first consider the case where $m \geq 2$, $n \geq 2$, and $p \geq 2$. Let us define the following DFAs A, B, and C over the same alphabet $\Sigma = \{a, b, c\}$.

Let $A = (Q_1, \Sigma, \delta_1, 0, F_1)$, where $Q_1 = \{0, 1, \ldots, m-1\}$, $F_1 = \{m-1\}$, and the transitions are given as:

- $\delta_1(i, a) = (i+1) \bmod m$, $i \in Q_1$,
- $\delta_1(i, b) = i + 1$, if $i \leq m - 3$, $\delta_1(m-2, b) = 0$,
- $\delta_1(m-1, b) = (m - n + 1) \bmod (m-1)$,
- $\delta_1(i, c) = i$, $i \in Q_1$.

Let $B = (Q_2, \Sigma, \delta_2, 0, F_2)$, where $Q_2 = \{0, 1, \ldots, n-1\}$, $F_2 = \{n-1\}$, and the transitions are given as:

- $\delta_2(i, a) = i + 1$, $i \leq n - 2$, $\delta_2(n-1, a) = n - 1$,
- $\delta_2(i, b) = (i+1) \bmod n$, $i \in Q_2$,
- $\delta_2(i, c) = i$, $i \in Q_2$.

Let $C = (Q_3, \Sigma, \delta_3, 0, F_3)$, where $Q_3 = \{0, 1, \ldots, p-1\}$, $F_3 = \{p-1\}$, and the transitions are given as:

- $\delta_3(i, x) = i$, $i \in Q_3$ and $x \in \{a, b\}$,
- $\delta_3(i, c) = (i+1) \bmod p$, $i \in Q_3$.

Note that, in DFAs A and B, the transitions on letters a and b are exactly the same as those defined in the DFAs in [15] that prove the lower bound of the state complexity of catenation. Moreover, no state will change after reading a letter c. Let $D = (Q_4, \Sigma, \delta_4, 0, F_4)$ be the DFA accepting $L(A)L(B)$. Thus, D does not move on letter c, it has $|Q_4| = m2^n - 2^{n-1}$ reachable states, and any two states in Q_4 are not equivalent.

Then, as described at the beginning of this proof, we construct the DFA $E = (Q_5, \Sigma, \delta_5, \langle 0, 0\rangle, F_5)$, where Q_5 is a Cartesian product of Q_4 and Q_3. For each state in Q_5, δ_5 simulates the transitions of D on its first element and simulates the transitions of C on its second element. Furthermore, each state in F_5 consists of a final state in F_4 and the final state in F_3. Next we show that (I) all the states in Q_5 are reachable and (II) any two of them are not equivalent. It is clear that (I) is true, because, using the proof of Theorem 1 in [15], any state $\langle s, 0\rangle$, $s \in Q_4$, can be reach from the initial state $\langle 0, 0\rangle$ by reading a string over letters a and b, and then, any state $\langle s, i\rangle$, $s \in Q_4$, can be reached from the state $\langle s, 0\rangle$ by reading c^i. For (II), let $\langle s_1, i_1\rangle$ and $\langle s_2, i_2\rangle$ be two different states in Q_5. If $s_1 = s_2$, then there exists a string w_1 such that, by reading w_1, we can reach a final state in F_4 from the state s_1. Thus, string $w_1c^{p-i_1-1}$ will distinguish the states $\langle s_1, i_1\rangle$ and $\langle s_2, i_2\rangle$. If $s_1 \neq s_2$, then there exists a string w_2 such that w_2 leads s_1 to a final state in F_4 but does not lead s_2 to any final state in F_4. Thus, string $w_2c^{p-i_1-1}$ will distinguish the states $\langle s_1, i_1\rangle$ and $\langle s_2, i_2\rangle$. After verifying (I) and (II), we can say that the size of Q_5 is $(m2^n - 2^{n-1})p$, and therefore this number is the state complexity of $L_1L_2 \cap L_3$ when $m \geq 2$, $n \geq 2$, and $p \geq 2$.

Next we consider the case where $m = 1$, $n \geq 2$, and $p \geq 2$. We use the alphabet $\Sigma = \{a, b, c\}$. L_1 is Σ^*, and we use the same DFA C for L_3. Here we define $F = (Q_6, \Sigma, \delta_6, 0, F_6)$ for L_2, where $Q_6 = \{0, 1, \ldots, n-1\}$, $F_6 = \{n-1\}$, and the transitions are given as follows:

- $\delta_6(0, a) = 0$, $\delta_6(i, a) = i + 1$, $1 \leq i \leq n-2$, $\delta_6(n-1, a) = 1$,
- $\delta_6(0, b) = 1$, $\delta_6(i, b) = i$, $1 \leq i \leq n-1$,
- $\delta_6(i, c) = i$, $i \in Q_6$.

Note that, without the transitions on letter c, F is the second witness DFA in [25] that proves the lower bound of the state complexity of catenation when $m = 1$ and $n \geq 2$. Thus, the proof for this case is very similar to that in the previous case and hence is omitted.

For (2), recall that the state complexity of L_1L_2 is m when $m \geq 1$ and $n = 1$. Thus, mp is the composition of the state complexities of catenation and intersection, and it is an upper bound of the state complexity of $L_1L_2 \cap L_3$ when $m \geq 1$, $n = 1$, and $p \geq 2$. To prove (2), we just need to show the existence of worst case examples that reach this upper bound. Let

$$\begin{aligned} L_1 &= \{w \in \{a, b\}^* \mid |w|_a \equiv m - 1 (\mathrm{mod}\ m)\}, \\ L_2 &= \{a, b\}^*, \text{ and} \\ L_3 &= \{w \in \{a, b\}^* \mid |w|_b \equiv p - 1 (\mathrm{mod}\ p)\}. \end{aligned}$$

It is clear that L_1, L_2, and L_3 are accepted by $m-$, $1-$, and $p-$state DFAs, respectively. The DFA accepting L_1L_2 has m states. Then the proof method is exactly the same as the previous ones, and hence is omitted. □

9. State complexity of $L_1L_2 \cup L_3$

In this section, we investigate the state complexity of $L_1L_2 \cup L_3$ for regular languages L_1, L_2, and L_3 accepted by m-state, n-state, and p-state DFAs, respectively. When $p = 1$, L_3 is either Σ^* or $\emptyset$. Thus, $L_1L_2 \cup L_3$ is either Σ^* or L_1L_2. Therefore, when $p = 1$, the state complexity of $L_1L_2 \cup L_3$ is equal to that of L_1L_2. For the other cases, we will show that the state complexity of $L_1L_2 \cup L_3$ is $mp - p + 1$ when $m \geq 1$, $n = 1$, and $p \geq 2$ (Lemma 9.1), and it is $(m2^n - 2^{n-1})p$ when $m \geq 1$, $n \geq 2$, and $p \geq 2$ (Theorem 9.1).

We first consider the case where $m \geq 1$, $n = 1$, and $p \geq 2$.

Lemma 9.1. *Let L_1, L_2, and L_3 be languages accepted by m-state, n-state, and p-state DFAs, respectively. Then, when $m \geq 1$, $n = 1$, and $p \geq 2$, the state complexity of $L_1L_2 \cup L_3$ is $mp - p + 1$.*

Proof. Let us denote by A, B, and C the m-state, n-state, and p-state DFAs, respectively.

We first show that $mp - p + 1$ is an upper bound of the state complexity of $L_1L_2 \cup L_3$. In the construction of a DFA E that accepts $L_1L_2 \cup L_3$, we first construct a DFA D that accepts L_1L_2. Then, the set of the states of E is a Cartesian product of the state sets of D and C, the initial state of E is a pair of the initial states of D and C, and each final state of E contains a final state of D or the final state of C. Moreover, the transitions of E simulates the transitions of D and C on the first element and the second element of each state of E, respectively. Note that B has only one state and it will go back to this state on any letter in Σ. As a result, the final state f of D will return to itself on any letter in Σ as well.

We know that, when $m \geq 1$ and $n = 1$, the state complexity of L_1L_2 is m. Thus, E has at most mp states. Because f will return to itself on any letter in Σ, all the states $\langle f, i\rangle$, where i is a state of C, are clearly equivalent. Therefore, $mp - p + 1$ is an upper bound of the state complexity of $L_1L_2 \cup L_3$ when $m \geq 1$, $n = 1$, and $p \geq 2$.

To show that this upper bound is reachable, we use the language $L_2 = \{a, b\}^*$, and the DFAs G and H in the proof of Theorem 8.1 for L_1 and L_3, respectively. The proof is straightforward, and hence is omitted. □

For the remaining cases, that is when $m \geq 1$, $n \geq 2$, and $p \geq 2$, we obtain the following result.

Theorem 9.1. *Let L_1, L_2, and L_3 be languages accepted by m-state, n-state, and p-state DFAs, respectively. Then, when $m \geq 1$, $n \geq 2$, and $p \geq 2$, the state complexity of $L_1L_2 \cup L_3$ is $(m2^n - 2^{n-1})p$.*

Proof. Let us denote by A, B, and C the m-state, n-state, and p-state DFAs, respectively.

Since the claimed state complexity is exactly the composition of the state complexities of catenation and union, the construction of a DFA E that accepts $L_1L_2 \cup L_3$ is as follows. We first construct a DFA D that accepts L_1L_2. Then, the set of the states of E is a Cartesian product of the sets of the states of D and C, the initial state of E is a pair of the initial states of D and C, and each final state of E contains a final state of D or the final state of C. Moreover, the transitions of E simulates the transitions of D and C on the first element and the second element of each state of E, respectively. Since the state complexity of L_1L_2 is $m2^n - 2^{n-1}$ when $m \geq 1$ and $n \geq 2$, the total number of states in E is upper bounded by $(m2^n - 2^{n-1})p$. To prove the theorem, we just need to show that there exist witness DFAs that reach this upper bound.

We first consider the case where $m = 1$, $n \geq 2$, and $p \geq 2$. We use the alphabet $\Sigma = \{a, b, c, d\}$, and $L_1 = \Sigma^*$.

Define $B = (Q_2, \Sigma, \delta_2, 0, F_2)$ that accepts L_2, where $Q_2 = \{0, 1, \ldots, n-1\}$, $F_2 = \{n-1\}$, the transitions on letters a, b, and c are exactly the same as those defined in the DFA F used in the proof of Theorem 8.1, and the transitions on letter d are given as $\delta_2(i, d) = 0, i \in Q_2$.

Define $C = (Q_3, \Sigma, \delta_3, 0, F_3)$ that accepts L_3, where $Q_3 = \{0, 1, \ldots, p-1\}$, $F_3 = \{p-1\}$, the transitions on letters a, b, and c are exactly the same as those defined in the DFA C used in the proof of Theorem 8.1, and the transitions on letter d are given as $\delta_3(i, d) = i, i \in Q_3$.

As described at the beginning of this proof, we first construct the DFA D. Note that, without the transitions on letters c and d, B is the second witness DFA in [25] that proves the lower bound of the state complexity of catenation when $m = 1$ and $n \geq 2$. Thus, D has 2^{n-1} states, all these states are reachable, and any two of the states are not equivalent. After constructing $E = (Q_5, \Sigma, \delta_5, \langle 0, 0\rangle, F_5)$ we just need to show that (I) all the states in Q_5 are reachable, and (II) any two states in Q_5 are not equivalent. The reachability of all the states in Q_5 is immediate since all the transitions on letters a, b, and c of B and C are exactly the same as those defined in the DFAs F and C used in the proof of Theorem 8.1, respectively.

For (II), let $\langle s_1, i_1\rangle$ and $\langle s_2, i_2\rangle$ be two different states in Q_5. We consider the following two cases:

1. $i_1 \neq i_2$. The string dc^{p-1-i_1} will distinguish these two states.
2. $i_1 = i_2$. We have $s_1 \neq s_2$, and there exists a string $w \in \{a, b\}^*$ such that, after reading w, we can reach a final state of D from s_1, but we cannot reach any final state of D from s_2. As a result, if i_1 is not a final state of C, then w will distinguish $\langle s_1, i_1\rangle$ from $\langle s_2, i_2\rangle$, otherwise, the string cw will distinguish these two states.

Since E has $2^{n-1}p$ reachable states and any two of them are not equivalent, we have showed the existence of witness DFAs that prove the state complexity of $L_1L_2 \cup L_3$ to be $(m2^n - 2^{n-1})p$ when $m = 1$, $n \geq 2$, and $p \geq 2$.

In the following, we consider the case where $m \geq 2$, $n \geq 2$, and $p \geq 2$. We use the same DFAs A, B, and C used in the proof of Theorem 8.1 for L_1, L_2, and L_3, respectively, and denote them by A', B', and C'. As described at the beginning of this proof, we construct D' and E' for L_1L_2 and $L_1L_2 \cup L_3$, respectively. Note that the only difference between E' and the DFA E used in the proof of Theorem 8.1 is the definitions of their final state sets. Here, each final state of E' contains a final state of B' or the final state of C'. Thus, we can say that, E' has $(m2^n - 2^{n-1})p$ states, and all these states are reachable from its initial state. The proof for the reachability of the states of E' is exactly the same as the proof for the reachability of the states of the DFA E used in the proof of Theorem 8.1.

In order to prove the theorem, we need to show that any two states in E' are not equivalent in the next step. Before proving this, we need some details about the construction of D'. The DFAs A' and B' are obtained by adding the transitions on letter c to the DFAs in [15] that prove the lower bound of the state complexity of catenation. Thus, the set of the states of D' can be written in the same form as used in [15]:

$$Q_4 = \{\{i\} \cup S \mid i \in Q_1 - \{m-1\} \text{ and } S \subseteq Q_2\} \cup \{\{m-1\} \cup S \mid S \subseteq Q_2 - \{0\}\},$$

i.e., any state in Q_4 consists of exactly one state of Q_1 and some states of Q_2, and if a set in Q_4 contains the state $m-1$, then it does not contain the state 0 of Q_2. We know that there are $m2^n - 2^{n-1}$ reachable states in Q_4 and any two of them are not equivalent.

Now, we show that any two states in E' are not equivalent. Let $\langle t_1, j_1\rangle$ and $\langle t_2, j_2\rangle$ be two different states in E'. We consider the following two cases:

1. $j_1 = j_2$. Then, $t_1 \neq t_2$, and there exists a string w that will distinguish t_1 from t_2 in D'. Therefore, if j_1 is the final state of C', then string cw will distinguish $\langle t_1, j_1\rangle$ from $\langle t_2, j_2\rangle$, otherwise, w will distinguish these two states.
2. $j_1 \neq j_2$. We have three sub-cases. (1) $t_1 = t_2$ and t_1 is not a final state of D'. The string c^{p-j_1-1} will distinguish $\langle t_1, j_1\rangle$ from $\langle t_2, j_2\rangle$. (2) $t_1 = t_2$ and t_1 is a final state of D'. Let us rewrite t_1 as $t_1 = \{i\} \cup T$, where $i \in Q_1$ and $T \subseteq Q_2$. The string $a^{m-i}b^{n-1}c^{p-j_1-1}$ will distinguish $\langle t_1, j_1\rangle$ from $\langle t_2, j_2\rangle$, since after reading $a^{m-i}b^{n-1}$ t_1 will not reach any final state of D'. (3) $t_1 \neq t_2$. Then, there exists a string $w' \in \{a, b\}^*$ that leads the state t_1 to a final state of D' but does not lead the state t_2 to any final state of D'. Thus, string $w'c^{p-j_1-1}$ will distinguish the two states.

We have showed that E', which is constructed from A', B', and C', has $(m2^n - 2^{n-1})p$ reachable states, and any two of its states are not equivalent. Therefore, the state complexity of $L_1L_2 \cup L_3$ is equal to the composition of the state complexities of catenation and union, which is $(m2^n - 2^{n-1})p$. □

10. Conclusion

In this paper, we completed the investigation of the state complexity of combined operations with two basic operations, by studying the state complexities of $(L_1L_2)^R$, $L_1^RL_2$, $L_1^*L_2$, $(L_1 \cup L_2)L_3$, $(L_1 \cap L_2)L_3$, $L_1L_2 \cap L_3$, and $L_1L_2 \cup L_3$ for regular languages L_1, L_2, and L_3. In particular, we solved an open problem posed in [18] by showing that the upper bound proposed in [18] for the state complexity of $(L_1L_2)^R$ coincides with the lower bound and is thus indeed the state complexity of this combined operation when $m \geq 2$ and $n \geq 1$. Also, we showed that, due to the structural properties of DFAs obtained from reversal, star, and union, the state complexities of $L_1^RL_2$, $L_1^*L_2$, and $(L_1 \cup L_2)L_3$ are close to the mathematical compositions of the state complexities of their individual participating operations, although they are not exactly the same. Furthermore, we showed that, in the general cases, the state complexities of $(L_1 \cap L_2)L_3$, $L_1L_2 \cap L_3$, and $L_1L_2 \cup L_3$ are exactly equal to the mathematical compositions of the state complexities of their component operations.

Table 1
The state complexities of all the combinations of two basic operations, where L_1, L_2, and L_3 are accepted by DFAs of m, n, and p states, respectively. Note that we only list the most general case for each combined operation in this table.

Operation	State complexity	Most General Case
$(L_1 \cup L_2)^*$	$2^{m+n-1} - 2^{m-1} - 2^{n-1} + 1$ ([21])	$m, n \geq 2$
$(L_1 \cap L_2)^*$	$2^{mn-1} + 2^{mn-2}$ ([17])	$m, n \geq 2$
$(L_1L_2)^*$	$2^{m+n-1} + 2^{m+n-4} - 2^{m-1} - 2^{n-1} + m + 1$ ([9])	$m, n \geq 2$
$(L_1^R)^* = (L_1^*)^R$	2^m ([9])	$m \geq 1$
$(L_1 \cup L_2)^R$	$2^{m+n} - 2^m - 2^n + 2$ ([18])	$m, n \geq 3$
$(L_1 \cap L_2)^R$	$2^{m+n} - 2^m - 2^n + 2$ ([18])	$m, n \geq 3$
$(L_1L_2)^R$	$3 \cdot 2^{m+n-2} - 2^n + 1$ ([18] and Section 3)	$m \geq 2, n \geq 1$
$L_1^*L_2$	$5 \cdot 2^{m+n-3} - 2^{m-1} - 2^n + 1$, the DFA for L_1 has at least one final state that is not the initial state (Section 5)	$m, n \geq 2$
$L_1L_2^*$	$(3m-1)2^{n-2}$, the DFA for L_2 has at least one final state that is not the initial state ([3])	$m, n \geq 2$
$L_1^RL_2$	$3 \cdot 2^{m+n-2}$ (Section 4)	$m, n \geq 2$
$L_1L_2^R$	$m2^n - 2^{n-1} - m + 1$ ([3])	$m, n \geq 1$
$L_1(L_2 \cup L_3)$	$(m-1)(2^{n+p} - 2^n - 2^p + 2) + 2^{n+p-2}$ ([4])	$m, n, p \geq 1$
$L_1(L_2 \cap L_3)$	$m2^{np} - 2^{np-1}$ ([4])	$m, n, p \geq 1$
$L_1^* \cup L_2$	$3 \cdot 2^{m-2} \cdot n - n + 1$ ([11])	$m, n \geq 2$
$L_1^* \cap L_2$	$3 \cdot 2^{m-2} \cdot n - n + 1$ ([11])	$m, n \geq 2$
$L_1^R \cup L_2$	$2^m \cdot n - n + 1$ ([11])	$m, n \geq 2$
$L_1^R \cap L_2$	$2^m \cdot n - n + 1$ ([11])	$m, n \geq 2$
$(L_1 \cup L_2)L_3$	$mn2^p - (m+n-1)2^{p-1}$ (Section 6)	$m, n, p \geq 2$
$(L_1 \cap L_2)L_3$	$mn2^p - 2^{p-1}$ (Section 7)	$m, n \geq 1, p \geq 2$
$L_1L_2 \cap L_3$	$(m2^n - 2^{n-1})p$ (Section 8)	$m \geq 1, n, p \geq 2$
$L_1L_2 \cup L_3$	$(m2^n - 2^{n-1})p$ (Section 9)	$m \geq 1, n, p \geq 2$
$L_1L_2L_3$	$m2^{n+p} - 2^{n+p-1} - (m-1)2^{n+p-2}$ $-2^{n+p-3} - (m-1)(2^p - 1)$ ([8])	$m, n, p \geq 2$
$L_1 \cup L_2 \cup L_3$	mnp ([8])	$m, n, p \geq 1$
$L_1 \cap L_2 \cap L_3$	mnp ([8])	$m, n, p \geq 1$
$(L_1 \cup L_2) \cap L_3$	mnp ([8])	$m, n, p \geq 1$
$(L_1 \cap L_2) \cup L_3$	mnp ([8])	$m, n, p \geq 1$

A summary of the state complexity for *all* combinations of two basic operations on regular languages is presented in Table 1.

The results obtained and summarized in this paper are on regular languages. Therefore, future work might address the state complexity of the same operations for sub-families of the family of regular languages, such as finite languages and codes. Another interesting research direction is to investigate the state complexity of combined operations composed of language operations other than the basic ones, e.g. shuffle [2], proportional removal [6,19], cyclic shift [16,19], etc.

References

[1] H. Bordihn, M. Holzer, M. Kutrib, Determinization of finite automata accepting subregular languages, Theoretical Computer Science 410 (2009) 3209–3222.
[2] 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.
[3] 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 for publication).
[4] 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 for publication).
[5] M. Daley, M. Domaratzki, K. Salomaa, State complexity of orthogonal catenation, in: Proceedings of DCFS 2008, Charlottetown, PE, Canada, July 16–18, 2008, pp. 134–144.
[6] M. Domaratzki, State complexity and proportional removals, Journal of Automata, Languages and Combinatorics 7 (2002) 455–468.
[7] M. Domaratzki, A. Okhotin, State complexity of power, Theoretical Computer Science 410 (24–25) (2009) 2377–2392.
[8] Z. Ésik, Y. Gao, G. Liu, S. Yu, Estimation of state complexity of combined operations, Theoretical Computer Science 410 (35) (2009) 3272–3280.

[9] 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.
[10] Y. Gao, S. Yu, State complexity approximation, in: Proceedings of Descriptional Complexity of Formal Systems, 2009, pp. 163–174.
[11] Y. Gao, S. Yu, State complexity of union and intersection combined with star and reversal, Computing Research Repository (2010) arXiv:1006.3755v1.
[12] M. Holzer, M. Kutrib, State complexity of basic operations on nondeterministic finite automata, in: Proceedings of CIAA 2002, Tours, France, July 3–5, in: LNCS, vol. 2608, 2002, pp. 148–157.
[13] J.E. Hopcroft, R. Motwani, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, 2nd edition, Addison Wesley, 2001.
[14] 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.
[15] G. Jirásková, State complexity of some operations on binary regular languages, Theoretical Computer Science 330 (2005) 287–298.
[16] G. Jirásková, A. Okhotin, State complexity of cyclic shift, in: Proceedings of DCFS 2005, Como, Italy, June 30–July 2, 2005, pp. 182–193.
[17] 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.
[18] 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.
[19] A. Maslov, Estimates of the number of states of finite automata, Soviet Mathematics Doklady 11 (1970) 1373–1375.
[20] 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.
[21] A. Salomaa, K. Salomaa, S. Yu, State complexity of combined operations, Theoretical Computer Science 383 (2007) 140–152.
[22] A. Salomaa, K. Salomaa, S. Yu, Undecidability of the state complexity of composed regular operations, in: Proceedings of LATA 2011, Tarragona, Spain, May 26–31, in: LNCS, vol. 6638, 2011, pp. 489–498.
[23] A. Salomaa, D. Wood, S. Yu, On the state complexity of reversals of regular languages, Theoretical Computer Science 320 (2004) 293–313.
[24] S. Yu, Y. Gao, State complexity research and approximation, in: Proceedings of DLT 2011, invited talk, Milan, Italy, in: LNCS, vol. 6795, 2011, pp. 46–57.
[25] S. Yu, Q. Zhuang, K. Salomaa, The state complexity of some basic operations on regular languages, Theoretical Computer Science 125 (1994) 315–328.
[26] S. Yu, Regular languages, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. 1, Springer-Verlag, 1997, pp. 41–110.
[27] S. Yu, State complexity of regular languages, Journal of Automata, Languages and Combinatorics 6 (2) (2001) 221–234.
[28] S. Yu, On the state complexity of combined operations, in: Proceedings of CIAA 2006, Taipei, Taiwan, August 21–23, in: LNCS, vol. 4094, 2006, pp. 11–22.