# State complexity of star and square of union of $k$ regular languages*

Yuan Gao and Lila Kari

Department of Computer Science
The University of Western Ontario
London, Ontario, Canada N6A 5B7
E-mail: {ygao72,lila}@csd.uwo.ca

**Abstract.** In this paper, we study the state complexities of $(\bigcup_{i=1}^{k} L_i)^*$ and $(\bigcup_{i=1}^{k} L_i)^2$, where $L_i$, $1 \leq i \leq k$, $k \geq 2$ are regular languages. We obtain exact bounds for both of these multiple combined operations and show that they are much lower than the mathematical compositions of the state complexities of their basic individual component operations, but have similar forms with the state complexities of some participating combined operations.

*In Memory of Dr. Sheng Yu*

## 1 Introduction

State complexity is a fundamental topic in automata theory and its study dates back to the 1950's [16]. State complexity is a type of descriptional complexity for regular languages based on the number of states in their minimal finite automata. The state complexity of a language operation gives an upper bound for both the time and space complexity of the operation [20]. The study of state complexity is motivated by the use of automata of very large sizes in multiple areas, e.g. programming languages, natural language and speech processing, and so on.

Many papers on state complexity appeared in the literature, see, e.g., [4–6, 10, 12, 14, 15, 20, 21]. The state complexities of almost all the individual standard regular language operations, e.g., union, intersection, catenation, star, reversal, shuffle, orthogonal catenation, proportional removal, and cyclic shift, etc., have been obtained.

In practice, not only a single operation, but also a sequence of operations can be applied in some specific order. For example, primer extension, which is a basic biological operation, can be formalized as a combination of catenation and antimorphic involution [1]. Therefore, in the mid of 2000s, the study of state

* This research was supported by Natural Science and Engineering Council of Canada Discovery Grant R2824A01, Canada Research Chair Award to L. K.

complexity of combined operations was initiated [18, 22]. Following that, many results on this topic were obtained, e.g., [2, 3, 7–9, 13].

A theoretical reason for studying the state complexity of combined operations is that, given an arbitrary combined operation, we cannot use the mathematical composition of the state complexities of its individual component operations as its state complexity. The state complexity of a combined operation can be much lower than the aforementioned composition, because the resulting languages of one individual operation may not be among the worst case inputs of the next operation [13, 18]. An often used example for this phenomenon is $(L_1 \cup L_2)^*$, where $L_1$ and $L_2$ are regular languages accepted by $n_1$- and $n_2$-state DFAs, respectively. In [18], the state complexity of the combined operation $(L_1 \cup L_2)^*$ was proved to be $2^{n_1+n_2-1} - 2^{n_1-1} - 2^{n_1-1} + 1$, whereas the mathematical composition of the state complexities of union and star is $\frac{3}{4}2^{n_1 n_2}$.

It has been proved 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 this class of languages [19]. It seems that every combined operation must be investigated separately. However, the number of combined operations is obviously unlimited, and it is impossible to investigate all of them. Thus, the combined operations with arbitrarily many individual operations should be the emphasis of theoretical studies because they are more general than the basic combined operations which are composed of only a limited number of individual operations. The latter can indeed be viewed as the special cases of the former.

In this paper, we study such two general combined operations: $(\bigcup_{i=1}^{k} L_i)^*$ and $(\bigcup_{i=1}^{k} L_i)^2$, where $L_i$, $1 \leq i \leq k$, $k \geq 2$ are regular languages. Clearly, the combined operation $(L_1 \cup L_2)^*$ is an instance of $(\bigcup_{i=1}^{k} L_i)^*$. We show that the state complexity of star of union on $k$ regular languages is not only much lower than the mathematical composition of the state complexities of union and star, but also in a similar form with the state complexity of $(L_1 \cup L_2)^*$.

We obtain tight bounds for $(\bigcup_{i=1}^{k} L_i)^2$ as well. One interesting thing is, when we investigated this combined operation, we found that it could be considered as a combination of (1) union and square, or (2) union-catenation ($(L_1 \cup L_2)L_3$) and union, or (3) union and catenation-union ($L_1(L_2 \cup L_3)$). Finally, the tight upper bound was obtained with the last combination which has a similar form with the state complexity of $L_1(L_2 \cup L_3)$. It seems that decomposing a combined operation into its participating combined operations can give better upper bounds than the mathematical composition of the state complexities of its individual component operations.

In the next section, we introduce the basic notation and definitions used in this paper. In Sections 3 and 4, we investigate the state complexities of $(\bigcup_{i=1}^{k} L_i)^*$ and $(\bigcup_{i=1}^{k} L_i)^2$, respectively.

## 2 Preliminaries

A DFA is denoted by a 5-tuple $A = (Q, \Sigma, \delta, s, F)$, where $Q$ is the finite set of states, $\Sigma$ 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 $\delta$ to $Q \times \Sigma^* \to Q$ in the usual way.

In this paper, the state transition function $\delta$ 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 $\delta$ instead of $\hat{\delta}$ if there is no confusion.

A string $w \in \Sigma^*$ is accepted by a DFA if $\delta(s, w) \in F$. Two states in a DFA $A$ 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. A language accepted by a DFA is said to be *regular*. The language accepted by a DFA $A$ is denoted by $L(A)$. The reader may refer to [11] for more details about regular languages and finite automata.

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

## 3 State complexity of $(\bigcup_{i=1}^{k} L_i)^*$

We first consider the state complexity of $(\bigcup_{i=1}^{k} L_i)^*$, where $L_i$, $1 \leq i \leq k$, $k \geq 2$ are regular languages accepted by $n_i$-state DFAs. It has been proved that the state complexity of $L_i^*$ is $\frac{3}{4}2^{n_i}$ and the state complexity of $L_i \cup L_j$ is $n_i n_j$ [14, 21]. Their mathematical composition for the combined operation $(\bigcup_{i=1}^{k} L_i)^*$ is $\frac{3}{4}2^{\prod_{i=1}^{k} n_i}$. As we mentioned in Section 1, this upper bound is too high to be reached even when $k = 2$, that is, $(L_1 \cup L_2)^*$ [18]. The combined operation $(L_1 \cup L_2)^*$ can be

viewed as not only a base case of $(\bigcup_{i=1}^{k} L_i)^*$ when $k = 2$, but also its participating combined operation.

In the following, we show that the state complexity of $(\bigcup_{i=1}^{k} L_i)^*$ has a similar form with that of $(L_1 \cup L_2)^*$. Note that although these two state complexities look similar, the proofs for the general case $k \geq 2$ is very different from those for $k = 2$, especially the proof for the highest lower bound. This is because, when $k$ is arbitrarily many, a lot more questions need to be considered which are easy to solve or do not exist for the case with only two operand languages, e.g., how to update the $i$th component of a state of the resulting DFA without interfering with the other $k - 1$ components, and so on.

**Theorem 1.** *Let $L_i$, $1 \leq i \leq k$, $k \geq 2$ be regular languages accepted by $n_i$-state DFAs. Then $(\bigcup_{i=1}^{k} L_i)^*$ is accepted by a DFA of no more than*

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

*states.*

**Proof.** For $1 \leq i \leq k$, let $L_i = L(A_i)$ and $A_i = (Q_i, \Sigma, \delta_i, s_i, F_i)$ be a DFA of $n_i$ states. Without loss of generality, we assume that the state sets of $A_1$, $A_2$, ..., $A_k$ are disjoint. We construct a DFA $A = (Q, \Sigma, \delta, s, F)$ to accept the language $(\bigcup_{i=1}^{k} L_i)^*$ similarly with [18]. We define $Q$ to be $Q = \{s\} \cup P \cup R$ where

$$P = \{\langle P_1, P_2, \ldots, P_k\rangle \mid P_i \subseteq Q_i - F_i, P_i \neq \emptyset, 1 \leq i \leq k\},$$
$$R = \{\langle R_1, R_2, \ldots, R_k\rangle \mid (\bigcup_{j=1}^{k} R_j) \cap (\bigcup_{h=1}^{k} F_h) \neq \emptyset, s_i \in R_i \subseteq Q_i, 1 \leq i \leq k\}.$$

If $s_i \notin F_i$ for every DFA $A_i$, $1 \leq i \leq k$, the initial state $s$ of the DFA $A$ is then a new symbol, because the empty word is not in the language $\bigcup_{i=1}^{k} L_i$. If there exists an $i$ such that $s_i \in F_i$, we choose $s = \langle s_1, s_2, \ldots, s_k\rangle$ to be the initial state of $A$. In this case, $s$ is clearly contained in the set $R$. Note that the sets $P$ and $R$ are always disjoint.

We define the set of final states $F$ to be $R \cup \{s\}$. The transition function $\delta$ of the DFA $A$ is defined as follows.

For each letter $a \in \Sigma$,

$$\delta(s, a) = \begin{cases} \langle\{\delta_1(s_1, a)\}, \ldots, \{\delta_k(s_k, a)\}\rangle, & \text{if } \delta_i(s_i, a) \notin F_i \text{ for all } 1 \leq i \leq k; \\ \langle\{\delta_1(s_1, a)\} \cup \{s_1\}, \ldots, \{\delta_k(s_k, a)\} \cup \{s_k\}\rangle, & \text{otherwise,} \end{cases}$$

and for each state $p = \langle P_1, P_2, \ldots, P_k\rangle \in Q - \{s\}$,

$$\delta(p, a) = \begin{cases} \langle\delta_1(P_1, a), \ldots, \delta_k(P_k, a)\rangle, & \text{if } \delta_i(P_i, a) \cap F_i = \emptyset \text{ for all } 1 \leq i \leq k; \\ \langle\delta_1(P_1, a) \cup \{s_1\}, \ldots, \delta_k(P_k, a) \cup \{s_k\}\rangle, & \text{otherwise.} \end{cases}$$

The DFA $A$ can simulate the computation of the DFAs $A_1$, $A_2$, ..., $A_k$ and when one of them enter a final state, the initial states $s_1$, $s_2$, ..., $s_k$ are added. It is easy to see that $L(A) = (\bigcup_{i=1}^{k} L(A_i))^*$.

Now let us count the number of states of $A$ which is an upper bound of the state complexity of the combined operation $(\bigcup_{i=1}^{k} L_i)^*$.

For the DFAs $A_1$, $A_2$, ..., $A_k$, denote $|F_i|$ by $t_i$. The resulting language

$$(\bigcup_{i=1}^{k} L_i)^* = \begin{cases} \Sigma^*, \text{ if } t_i = n_i; \\ (L_1 \cup L_2 \cup \ldots \cup L_{i-1} \cup L_{i+1} \ldots \cup L_k)^*, \text{ if } t_i = 0. \end{cases}$$

Both of the above cases are trivial. Therefore, we only need to consider the case when $0 < t_i < n_i$. There are $\prod_{i=1}^{k}(2^{n_i - t_i} - 1)$ states in the set $P$. The cardinality of the set $R$ is

$$|R| = \begin{cases} 2^{\sum_{j=1}^{k} n_j - k}, \text{ if } \exists p(s_p \in F_p), 1 \leq p \leq k; \\ 2^{\sum_{j=1}^{k} n_j - k} - 2^{\sum_{j=1}^{k} n_j - \sum_{r=1}^{k} t_r - k}, \text{ otherwise.} \end{cases}$$

There are $2^{\sum_{j=1}^{k} n_j - k}$ states $\langle R_1, R_2, \ldots, R_k \rangle$ in $A$ such that $s_i \in R_i$ for all $1 \leq i \leq k$. When $s_p \notin F_p$ for all $1 \leq p \leq k$, the number of states $\langle R'_1, R'_2, \ldots, R'_k \rangle$ such that $s_p \in R_p$ and $F_p \cap R_p = \emptyset$ is $2^{\sum_{j=1}^{k} n_j - \sum_{r=1}^{k} t_r - k}$. In this case, these states are contained in the set $P$ rather than $R$ according to the definition.

Since $Q = \{s\} \cup P \cup R$, the size of the state set $Q$ is

$$|Q| = \begin{cases} \prod_{i=1}^{k}(2^{n_i - t_i} - 1) + 2^{\sum_{j=1}^{k} n_j - k}, \text{ if } \exists p(s_p \in F_p), 1 \leq p \leq k; \\ \prod_{i=1}^{k}(2^{n_i - t_i} - 1) + 2^{\sum_{j=1}^{k} n_j - k} - 2^{\sum_{j=1}^{k} n_j - \sum_{r=1}^{k} t_r - k} + 1, \text{ otherwise.} \end{cases}$$

As we mentioned before, a new symbol is needed to be the initial state only when $s_i \notin F_i$ for all $1 \leq i \leq k$. Thus, the upper bound of the number of states in $A$ reaches the worst case when $A_i$ has only one final state ($t_i = 1$) for all $1 \leq i \leq k$ and at least one of the initial states of these DFAs is final. □

Next, we show that this upper bound is reachable.

**Theorem 2.** *For any integer $n_i \geq 3$, $1 \leq i \leq k$, there exist a DFA $A_i$ of $n_i$ states such that any DFA accepting $(\bigcup_{i=1}^{k} L(A_i))^*$ needs at least*

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

*states.*

**Proof.** For $1 \leq i \leq k$, let $A_i = (Q_i, \Sigma, \delta_i, 0, \{0\})$ be a DFA, where $Q_1 = \{0, 1, \ldots, n_i - 1\}$, $\Sigma = \{a_i \mid 1 \leq i \leq k\} \cup \{b_j \mid 1 \leq j \leq k\} \cup \{c\}$ and the transitions of $A_i$ are

$$\begin{aligned}
&\delta_i(q, a_i) = q + 1 \bmod n_i,\ q = 0, 1, \ldots, n_i - 1,\\
&\delta_i(q, a_j) = q,\ j \neq i,\ q = 0, 1, \ldots, n_i - 1,\\
&\delta_i(q, b_i) = 0,\ q = 0, 1, \ldots, n_i - 1,\\
&\delta_i(q, b_j) = q,\ j \neq i,\ q = 0, 1, \ldots, n_i - 1,\\
&\delta_i(0, c) = 1,\ \delta_i(q, c) = q,\ q = 1, \ldots, n_i - 1.
\end{aligned}$$

The transition diagram of $A_i$ is shown in Figure 1.

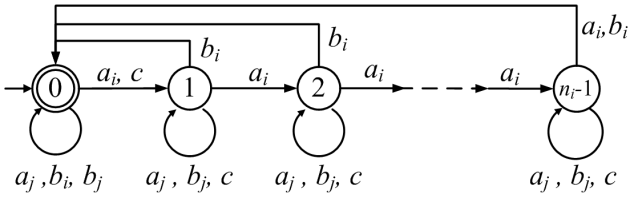


**Fig. 1.** Witness DFA $A_i$ for Theorems 2

Then we construct the DFA $A = (Q, \Sigma, \delta, s, F)$ exactly as described in the proof of Theorem 1, where

$$\begin{aligned}
&Q = P \cup R,\\
&P = \{\langle P_1, P_2, \ldots, P_k\rangle \mid P_i \subseteq Q_i - \{0\}, P_i \neq \emptyset, 1 \leq i \leq k\},\\
&R = \{\langle R_1, R_2, \ldots, R_k\rangle \mid 0 \in R_i \subseteq Q_i, 1 \leq i \leq k\},\\
&s = \langle\{0\}, \{0\}, \ldots, \{0\}\rangle,\\
&F = \{\langle\{0\}, \{0\}, \ldots, \{0\}\rangle\},
\end{aligned}$$

and for each state $p = \langle P_1, P_2, \ldots, P_k\rangle \in Q$,

$$\delta(p, a) = \begin{cases} \langle\delta_1(P_1, a), \delta_2(P_2, a), \ldots, \delta_k(P_k, a)\rangle, \text{ if } 0 \notin \delta_i(P_i, a) \text{ for all } 1 \leq i \leq k;\\ \langle\delta_1(P_1, a) \cup \{0\}, \delta_2(P_2, a) \cup \{0\}, \ldots, \delta_k(P_k, a) \cup \{0\}\rangle, \text{ otherwise.}\end{cases}$$

It is easy to see that $A$ accepts $(\bigcup_{i=1}^{k} L(A_i))^*$ and it has $\prod_{i=1}^{k}(2^{n_i-1}-1)+2^{\sum_{j=1}^{k} n_j - k}$ states. Now we need to show that $A$ is a minimal DFA.

(I) We first show that every state $p = \langle P_1, P_2, \ldots, P_k\rangle \in Q$ is reachable from the initial state $s = \langle\{0\}, \{0\}, \ldots, \{0\}\rangle$.

1. $|P_1| \geq 1$, $|P_2| = |P_3| = \ldots = |P_k| = 1$. According to the nature of the combined operation of star of union, the order of $|P_1|$, $|P_2|$, ..., $|P_k|$ does not matter. Thus, in this case, we just let $|P_1| \geq 1$ and $|P_2|$, ..., $|P_k|$ be 1 without loss of generality. Let us use induction on the cardinality of $P_1$ to prove this.
   **Base:** We show that, when $|P_1| = |P_2| = |P_3| = \ldots = |P_k| = 1$, the state $p$ is reachable from the initial state. Assume that $P_i = \{q_i\} \subseteq Q_i$, $1 \leq i \leq k$. Then

$$\langle P_1, P_2, \ldots, P_k \rangle = \begin{cases} s, \text{ if } q_1 = 0; \\ \delta(s, ca_1^{q_1-1}a_2^{q_2-1} \cdots a_k^{q_k-1}), \text{ if } q_1 > 0. \end{cases}$$

   Note that when $q_1 = 0$, $q_2, \ldots, q_k$ must also be 0 according to the construction of the DFA $A$. Similarly, when $q_1 > 0$, all of $q_2, \ldots, q_k$ must be greater than 0.
   **Induction step:** Assume that all states in $A$ such that $|P_1| = m_1 \geq 1$, $|P_2| = |P_3| = \ldots = |P_k| = 1$ are reachable from $s$. Then we prove any state $p$ such that $|P_1| = m_1 + 1$, $|P_2| = |P_3| = \ldots = |P_k| = 1$ is also reachable.
   Assume $P_1 = \{q_{11}, q_{12}, \ldots, q_{1m_1}, q_{1(m_1+1)}\} \subseteq Q_1$, $q_{11} < q_{12} < \ldots < q_{1m_1} < q_{1(m_1+1)}$, $P_j = \{q_{j1}\} \subseteq Q_j$, $2 \leq j \leq k$. Then

$$p = \begin{cases} \delta(p', b_2b_3 \cdots b_k), \text{ if } q_{11} = 0; \\ \delta(p'', ca_1^{q_{11}-1}a_2^{q_2-1}a_3^{q_3-1} \cdots a_k^{q_k-1}), \text{ if } q_{11} > 0, \end{cases}$$

   where

$$p' = \langle \{q_{12}, q_{13}, q_{14}, \ldots, q_{1(m_1+1)}\}, \{1\}, \ldots, \{1\} \rangle,$$
$$p'' = \langle \{0, q_{12} - q_{11} + 1, \ldots, q_{1(m_1+1)} - q_{11} + 1\}, \{0\}, \ldots, \{0\} \rangle.$$

   Since the state $p'$ is reachable according to the induction hypothesis and $p''$ has been proved to be reachable in the case when $q_{11} = 0$, the state $p$ can also be reached.
2. $|P_1| \geq 1$, $|P_2| \geq 1$, ..., $|P_t| \geq 1$, $|P_{t+1}| = |P_{t+2}| \ldots = |P_k| = 1$, $2 \leq t \leq k$. We use induction on $t$ to prove that $p$ is reachable in this case. Case 1 can be used as the base of the induction.
   **Induction step:** Assume all states in $A$ such that $|P_1| = m_1 \geq 1$, $|P_2| = m_2 \geq 1$, ..., $|P_{t-1}| = m_{t-1} \geq 1$, $|P_t| = |P_{t+1}| \ldots = |P_k| = 1$, $2 \leq t \leq k$, can be reached from the initial state $s$. Let us prove any state $p$ such that $|P_1| = m_1 \geq 1$, $|P_2| = m_2 \geq 1$, ..., $|P_t| = m_t \geq 1$, $|P_{t+1}| = |P_{t+2}| \ldots = |P_k| = 1$ can also be reached.
   Assume $P_i = \{q_{i1}, q_{i2}, \ldots, q_{im_i}\} \subseteq Q_i$, $q_{i1} < q_{i2} < \ldots < q_{im_i}$, $2 \leq m_i \leq n_i$ $P_j = \{q_{j1}\} \subseteq Q_j$, $1 \leq i \leq t$, $t + 1 \leq j \leq k$. In the following, let us first consider the case when $q_{11} > 0$ this time.
   (2.1) $q_{11} > 0$. If $q_{11} > 0$, then $q_{21} > 0$, $q_{31} > 0$, ..., $q_{k1} > 0$ and $P_i \neq Q_i$ for all $1 \leq i \leq t$. According to the induction hypothesis, the state

$$p' = \langle P_1, P_2, \ldots, P_{t-1}, \{1\}, \{1\}, \ldots, \{1\} \rangle$$

is reachable from $s$. We begin the computation from $p'$ by reading $q_{tm_t} - q_{t(m_t-1)} - 1$ symbols $a_t$.

$$\delta(p', a_t^{q_{tm_t}-q_{t(m_t-1)}-1}) = \langle P_1, \ldots, P_{t-1}, \{q_{tm_t} - q_{t(m_t-1)}\}, \{1\}, \ldots, \{1\}\rangle.$$

Denote the resulting state by $r$. Next, we apply $n_1 - q_{1m_1}$ symbols $a_1$ and the DFA $A$ reaches the state

$$r' = \langle P_1', P_2 \cup \{0\}, \ldots, P_{t-1} \cup \{0\}, \{0, q_{tm_t} - q_{t(m_t-1)}\}, \{0,1\}, \ldots, \{0,1\}\rangle$$

where

$$P_1' = \{0, q_{11} + n_1 - q_{1m_1}, q_{12} + n_1 - q_{1m_1}, \ldots, q_{1(m_1-1)} + n_1 - q_{1m_1}\}.$$

Now we apply an $a_t$-transition and the resulting state $r''$ is

$$\langle P_1', P_2 \cup \{0\}, \ldots, P_{t-1} \cup \{0\}, \{0, 1, q_{tm_t} - q_{t(m_t-1)} + 1\}, \{0,1\}, \ldots, \{0,1\}\rangle.$$

We cycle using $a_1$-transitions as long as elements of $P_1'$ are consecutively passing by 0. The last $a_1$-transition increases the cardinality of $P_1'$ by 1 and after that we apply a $c$-transition which removes the 0 in every component of the state. We continue to apply $a_1$-transitions until a sequence of consecutive elements of $P_1'$ passed by 0 and the cardinality of $P_1'$ is increased by 1. Then a $c$-transition is applied to eliminate 0. Clearly, we can cyclicly shift the set $P_1'$ back into $P_1$ by repeating these two steps. Now the DFA $A$ reaches the state

$$p'' = \langle P_1, P_2, \ldots, P_{t-1}, \{1, q_{tm_t} - q_{t(m_t-1)} + 1\}, \{1\}, \ldots, \{1\}\rangle.$$

The state $p''$ is the same as $p$ except that $q_{tm_t} - q_{t(m_t-1)} + 1$ is added into the $t$th set. Therefore, we can continue in the same way to add more elements to it. After the next loop, the state reached will be

$$\langle P_1, \ldots, P_{t-1}, \{1, q_{t(m_t-1)} - q_{t(m_t-2)} + 1, q_{tm_t} - q_{t(m_t-2)} + 1\}, \{1\}, \ldots, \{1\}\rangle.$$

In this way, we add all the $m_t$ elements of $P_t$ but keep them in a position that is shifted backwards $q_{t1} - 1$ steps so that $q_{t1}$ is in the position 1, $q_{t2}$ is in the position $q_{t2} - q_{t1} + 1$, and so on. Now we use an input word $a_t^{q_{t1}-1}$ to shift all the elements of $P_t$ into correct positions, which does not change the other elements of the state, and the state is

$$p''' = \langle P_1, P_2, \ldots, P_{t-1}, P_t, \{1\}, \ldots, \{1\}\rangle.$$

Finally, by reading a word $a_{t+1}^{q_{(t+1)1}-1} a_{t+2}^{q_{(t+2)1}-1} \cdots a_k^{q_{k1}-1}$, the DFA $A$ reaches the state $p = \langle P_1, P_2, \ldots, P_k\rangle$.
(2.2) $q_{11} = 0$. When $q_{11} = 0$, we know that $q_{21} = q_{31} = \ldots = q_{k1} = 0$. Then the state $p$ is

$$\langle \{0, q_{12}, \ldots, q_{1m_1}\}, \ldots, \{0, q_{t2}, \ldots, q_{tm_t}\}, \{0\}, \ldots, \{0\}\rangle.$$

To prove $p$ is reachable, we start from a state

$$p' = \langle\{q_{12}, \ldots, q_{1m_1}\}, \ldots, \{q_{t2}, \ldots, q_{tm_t}\}, \{1\}, \ldots, \{1\}\rangle.$$

The state $p'$ has been proved to be reachable in the case (2.1). It is easy to see that $\delta(p', b_{t+1}b_{t+2}\cdots b_k) = p$. Thus, the state $p$ can be reached from the initial state $s$ when $q_{11} = 0$.

Now we have proved that all the states in $A$ are reachable.

(II) Any two different states $p_1$ and $p_2$ in $Q$ are distinguishable.
Let $p_1$ and $p_2$ be $\langle P_1, P_2, \ldots, P_k\rangle$ and $\langle P'_1, P'_2, \ldots, P'_k\rangle$, respectively. Since $p_1$ and $p_2$ are different, without loss of generality we can assume that there exists an integer $1 \le t \le k$ such that $P_t \ne P'_t$ and $x \in P_t - P'_t$.

1. $x = 0$. If $x = 0$, then $0 \in P_i$ for all $1 \le i \le k$ and the state $p_1$ is a final state of $A$. Oppositely, since $x \notin P'_t$, none of $P'_i$ contains 0, which makes the state $p_2$ a nonfinal state. Therefore, $p_1$ and $p_2$ are distinguishable.
2. $x > 0$. For this case, we claim that $\delta(p_1, a_t^{m_t-1-x}ca) \in F$. In the DFA $A_t$, the transition function $\delta_t$ on the input word $a_t^{m_t-1-x}$ takes the state $x$ to $m_t - 1$. The input letter $c$ does not change the state $m_t - 1$ and the letter $a$ takes from $m_t - 1$ to 0. The last $a$-transition also adds 0 into the other components in $p_1$ according to the definition of $A$. Thus, the resulting state is final.
Next, we show that $\delta(p_2, a_t^{m_t-1-x}ca) \notin F$. Since $x \notin P'_t$, it is easy to see that $m_t - 1 \notin \delta(P'_t, a_t^{m_t-1-x})$. Note that 0 may be added into the other components in $p_2$ if the state 0 in $A_t$ is passed by when processing the input word $a_t^{m_t-1-x}$. However, since $x > 0$, it is impossible for a computation from the newly added 0's to reach $m_t - 1$ on $a_t^{m_t-1-x}$. Then the input letter $c$ removes the 0 in $P'_i$ for all $1 \le i \le k$. The last input letter $a$ shifts the states in $\delta(P'_t, a_t^{m_t-1-x}c)$ by 1 but none of its elements can reach 0 because it does not contain $m_t - 1$. The $a$-transition does not change the other elements in $P_2$. Clearly, the resulting state is nonfinal. Thus, the states $p_1$ and $p_2$ are distinguishable.

Since all states in $A$ are reachable and distinguishable, $A$ is a minimal DFA. □

This lower bound coincides with the upper bound in Theorem 1. Thus, it is the state complexity of $(\bigcup_{i=1}^{k} L(A_i))^*$.

## 4 State complexity of $(\bigcup_{i=1}^{k} L_i)^2$

In this section, we consider the state complexity of $(\bigcup_{i=1}^{k} L_i)^2$, where $L_i$, $1 \le i \le k$, $k \ge 2$ are regular languages accepted by $n_i$-state DFAs. As we mentioned in Section 1, this combined operation can be viewed as a combination of (1) union and square, or (2) union-catenation ($(L_1 \cup L_2)L_3$) and union, or (3) union and

catenation-union ($L_1(L_2 \cup L_3)$). It was shown that the state complexity of $L_1^2$ is $n_1 2^{n_1} - 2^{n_1-1}$ [17] and the state complexity of $L_1 \cup L_2$ is $n_1 n_2$ [14, 21]. Thus, for combination (1), we can get an upper bound through mathematical composition

$$\prod_{h=1}^{k} n_h \cdot 2^{\prod_{i=1}^{k} n_i} - 2^{\prod_{j=1}^{k} n_j - 1}$$

Next, we consider $(\bigcup_{i=1}^{k} L_i)^2$ as the second combination. The state complexity of $(L_1 \cup L_2)L_3$ was proved to be $n_1 n_2 2^{n_3} - (n_1 + n_2 - 1)2^{n_3-1}$ in [2]. Then its naive mathematical composition with the state complexity of union is

$$\prod_{h=1}^{k} n_h \cdot 2^{\prod_{i=1}^{k} n_i} - (n_1 + \prod_{j=2}^{k} n_j - 1)2^{\prod_{l=1}^{k} n_l - 1}$$

which is better than the first upper bound.

Now, let us consider the last combination. In [3], the state complexity of $L_1(L_2 \cup L_3)$ is shown to be

$$(n_1 - 1)[(2^{n_2} - 1)(2^{n_3} - 1) + 1] + 2^{n_2+n_3-2}$$

and its naive mathematical composition with the state complexity of union is

$$\prod_{h=1}^{k} (n_h - 1)[(2^{n_2} - 1)(2^{n_1 \prod_{i=3}^{k} n_i} - 1) + 1] + 2^{\sum_{j=1}^{k} n_j - 2}$$

which is the best among the three upper bounds.

In the following, we will show that the state complexity of $(\bigcup_{i=1}^{k} L_i)^2$ has a similar form with the third bound. Again, although the two state complexities look similar, the proofs vary a lot because one is a general combined operation for $k \geq 2$ and the other is a specific combined operation. Besides, the base case of the combined operation when $k = 2$, that is, $(L_1 \cup L_2)^2$, has never been studied. Its state complexity is obtained in this paper as a case of the general operation.

**Theorem 3.** *Let $L_i$, $1 \leq i \leq k$, $k \geq 2$ be regular languages accepted by DFAs of $n_i$ states and $f_i$ final states. Then $(\bigcup_{i=1}^{k} L_i)^2$ is accepted by a DFA of no more than*

$$\prod_{h=1}^{k} (n_h - f_h)[\prod_{i=1}^{k} (2^{n_i} - 1) + 1] + [\prod_{j=1}^{k} n_j - \prod_{l=1}^{k} (n_l - f_l)]2^{\sum_{m=1}^{k} n_m - k}.$$

*states.*

**Proof.** For $1 \leq i \leq k$, let $L_i = L(A_i)$ and $A_i = (Q_i, \Sigma, \delta_i, s_i, F_i)$ be a DFA of $n_i$ states and $f_i$ final states. We construct a DFA $A = (Q, \Sigma, \delta, s, F)$ to accept the language $(\bigcup_{i=1}^{k} L_i)^2$. We define the state set $Q$ to be $Q = P \cup R \cup T$, where

$$P = \{\langle p_1, p_2, \ldots, p_k, P_1, P_2, \ldots, P_k\rangle \mid p_i \in Q_i - F_i, P_i \in 2^{Q_i} - \{\emptyset\}, 1 \leq i \leq k\},$$
$$R = \{\langle p_1, p_2, \ldots, p_k, \emptyset, \ldots, \emptyset\rangle \mid p_i \in Q_i - F_i, 1 \leq i \leq k\},$$
$$T = \{\langle p_1, p_2, \ldots, p_k, \{s_1\} \cup P_1, \ldots, \{s_k\} \cup P_k\rangle \mid p_i \in F_i, P_i \in 2^{Q_i - \{s_i\}}, 1 \leq i \leq k\}.$$

The initial state $s$ is

$$s = \begin{cases} \langle s_1, s_2, \ldots, s_k, \emptyset, \emptyset, \ldots, \emptyset\rangle, \text{ if } s_i \notin F_i, 1 \leq i \leq k; \\ \langle s_1, s_2, \ldots, s_k, \{s_1\}, \{s_2\}, \ldots, \{s_k\}\rangle, \text{ otherwise.} \end{cases}$$

We define the set of final states $F$ to be

$$F = \{\langle p_1, p_2, \ldots, p_k, P_1, P_2, \ldots, P_k\rangle \in Q \mid \exists i(P_i \cap F_i \neq \emptyset), 1 \leq i \leq k\}.$$

For any $p \in Q$ and $a \in \Sigma$, the transition function $\delta$ is defined as:

$$\delta(p, a) = \begin{cases} \langle p'_1, p'_2, \ldots, p'_k, P'_1, P'_2 \ldots, P'_k\rangle, \text{ if } p'_i \cap F_i = \emptyset \text{ for all } 1 \leq i \leq k; \\ \langle p'_1, p'_2, \ldots, p'_k, P'_1 \cup \{s_1\}, P'_2 \cup \{s_2\}, \ldots, P'_k \cup \{s_k\}\rangle, \text{ otherwise,} \end{cases}$$

where $p'_i = \delta_i(p_i, a)$ and $P'_i = \delta_i(P_i, a)$, $1 \leq i \leq k$.

An arbitrary state in $A$ is a $2k$-tuple whose first $k$ components can be viewed as a state in the DFA accepting $\bigcup_{i=1}^{k} L_i$ constructed through cross-product and last $k$ components components are subsets of $Q_1$, $Q_2$, …, $Q_k$, respectively.

If the first $k$ components of a state are non-final states in $A_1$, $A_2$, …, $A_k$, respectively, then the last $k$ components are either all empty sets or all nonempty sets, because the last $k$ components always change from the empty set to a non-empty set at the same time. This is why $P$ and $R$ are subsets of $Q$.

Also, we notice that if at least one of the first $k$ components of a state in $A$ is final in the corresponding DFA, then the last $k$ components of the state must contain the initial states of $A_1$, $A_2$, …, $A_k$, respectively. Such states are contained in the set $T$.

It is easy to see that $A$ accepts $(\bigcup_{i=1}^{k} L_i)^2$. Now let us count the number of states in $A$. The cardinalities of $P$, $R$ and $T$ are respectively

$$|P| = \prod_{h=1}^{k}(n_h - f_h)[\prod_{i=1}^{k}(2^{n_i} - 1)], \qquad |R| = \prod_{h=1}^{k}(n_h - f_h),$$
$$|T| = [\prod_{j=1}^{k} n_j - \prod_{l=1}^{k}(n_l - f_l)]2^{\sum_{m=1}^{k} n_m - k}.$$

Thus, the total number of states in $A$ is $|P| + |R| + |T|$ which is the same as the upper bound shown in Theorem 3. □

Next, we show this upper bound can be reached.

**Theorem 4.** *For any integer $n_i \geq 3$, $1 \leq i \leq k$, there exist a DFA $A_i$ of $n_i$ states such that any DFA accepting $(\bigcup_{i=1}^{k} L(A_i))^2$ needs at least*

$$\prod_{h=1}^{k}(n_h-1)[\prod_{i=1}^{k}(2^{n_i}-1)+1]+[\prod_{j=1}^{k}n_j-\prod_{l=1}^{k}(n_l-1)]2^{\sum_{m=1}^{k}n_m-k}$$

*states.*

**Proof.** For $1 \leq i \leq k$, let $A_i = (Q_i, \Sigma, \delta_i, 0, \{n_i - 1\})$ be a DFA, where $Q_1 = \{0, 1, \ldots, n_i - 1\}$, $\Sigma = \{a_i \mid 1 \leq i \leq k\} \cup \{b_j \mid 1 \leq j \leq k\} \cup \{c\}$ and the transitions of $A_i$ are

$$\begin{aligned}
&\delta_i(q, a_i) = q + 1 \bmod n_i,\ q = 0, 1, \ldots, n_i - 1,\\
&\delta_i(q, a_j) = q,\ j \neq i,\ q = 0, 1, \ldots, n_i - 1,\\
&\delta_i(1, b_i) = 0,\ \delta_i(q, b_i) = q,\ q = 0, 2, 3 \ldots, n_i - 1,\\
&\delta_i(q, b_j) = q,\ j \neq i,\ q = 0, 1, \ldots, n_i - 1,\\
&\delta_i(q, c) = q + 1 \bmod n_i,\ q = 0, 1, \ldots, n_i - 1.
\end{aligned}$$

The transition diagram of $A_i$ is shown in Figure 2.

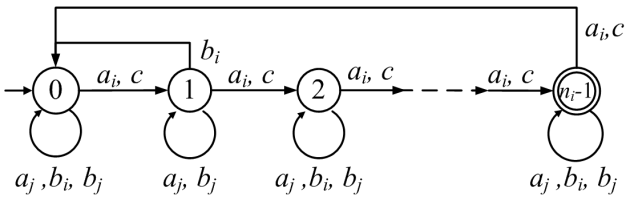


**Fig. 2.** Witness DFA $A_i$ for Theorems 4

Now we construct the DFA $A = (Q, \Sigma, \delta, s, F)$ accepting $(\bigcup_{i=1}^{k} L(A_i))^2$ exactly as described in the proof of Theorem 3. The number of states in $A$ is clearly

$$\prod_{h=1}^{k}(n_h-1)[\prod_{i=1}^{k}(2^{n_i}-1)+1]+[\prod_{j=1}^{k}n_j-\prod_{l=1}^{k}(n_l-1)]2^{\sum_{m=1}^{k}n_m-k}.$$

Next, we will prove that $A$ is a minimal DFA.

(I) We first need to show that every state

$$p = \langle p_1, p_2, \ldots, p_k, P_1, P_2, \ldots, P_k \rangle \in Q$$

is reachable from the initial state $s = \langle 0, 0, \ldots, 0, \emptyset, \emptyset, \ldots, \emptyset \rangle$. The reachability of $p$ can be proved by considering the following three cases.

1. $p_i \notin F_i, P_i = \emptyset, 1 \leq i \leq k$.
2. $|P_1| \geq 1, |P_2| = |P_3| = \ldots = |P_k| = 1$.
3. $|P_1| \geq 1, |P_2| \geq 1, \ldots, |P_t| \geq 1, |P_{t+1}| = \ldots = |P_k| = 1, 2 \leq t \leq k$.

Due to the page limitation, we omit the proof for the three cases above.

(II) Any two different states $p$ and $p'$ in $Q$ are distinguishable.
Assume that

$$p = \langle p_1, p_2, \ldots, p_k, P_1, P_2, \ldots, P_k \rangle,$$
$$p' = \langle p'_1, p'_2, \ldots, p'_k, P'_1, P'_2, \ldots, P'_k \rangle.$$

1. $\exists t(P_t \neq P'_t), 1 \leq t \leq k$.
   Let $x \in P_t - P'_t$ without loss of generality. Then there exists a word $w$ such that

   $$\delta(p, w) = \langle 0, \ldots, 0, r_t, 0, \ldots, 0, \{0\}, \ldots, \{0\}, R_t, \{0\}, \ldots, \{0\} \rangle \in F,$$
   $$\delta(p', w) = \langle 0, \ldots, 0, r'_t, 0, \ldots, 0, \{0\}, \ldots, \{0\}, R'_t, \{0\}, \ldots, \{0\} \rangle \notin F,$$

   where

   $$w = a^{n_t - 1 - x} w_1 w_2 \cdots w_{t-1} w_{t+1} w_{t+2} \cdots w_k,$$
   $$w_j = (a_j b_j)^{n_j}, 1 \leq j \leq k, j \neq t.$$

   It is easy see that $R_t \cap F_t \neq \emptyset$ whereas $R'_t \cap F_t = \emptyset$.
2. $\exists t(p_t \neq p'_t), 1 \leq t \leq k$ and $P_i = P'_i$ for all $1 \leq i \leq k$.

   For this case, there exists a word $w'$ such that

   $$\delta(p, w') = \langle 0, \ldots, 0, R_1, \{0\}, \ldots, \{0\} \rangle \in F,$$
   $$\delta(p', w') = \langle 0, \ldots, 0, R'_1, \{0\}, \ldots, \{0\} \rangle \notin F,$$

   where

   $$w' = w_1 w_2 \cdots w_{t-1} w_{t+1} w_{t+2} \cdots w_k w_t,$$
   $$w_j = (a_j b_j)^{n_j}, 1 \leq j \leq k, j \neq t,$$
   $$w_t = a_t^{n_t + 1 - p_t} (a_t b_t)^{n_t - 2} a_1^{n_1} a_t b_t.$$

   We can see that $R_1 \cap F_1 \neq \emptyset$ whereas $R'_1 \cap F_1 = \emptyset$.

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

$$\prod_{h=1}^{k} (n_h - 1)[\prod_{i=1}^{k} (2^{n_i} - 1) + 1] + [\prod_{j=1}^{k} n_j - \prod_{l=1}^{k} (n_l - 1)] 2^{\sum_{m=1}^{k} n_m - k}$$

states. □

Since this lower bound coincides with the upper bound in Theorem 3, it is the state complexity of the combined operation $(\bigcup_{i=1}^{k} L_i)^2$.

## References


1. 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, 23 (1) (2012) 51-56
2. B. Cui, Y. Gao, L. Kari, S. Yu: State complexity of combined operations with two basic operations, Theoretical Computer Science, accepted, 2011
3. B. Cui, Y. Gao, L. Kari, S. Yu: State complexity of two combined operations: catenation-union and catenation-intersection, International Journal of Foundations of Computer Science, 22 (8) (2011) 1797-1812
4. C. Campeanu, K. Culik, K. Salomaa, S. Yu: State complexity of basic operations on finite language, in: Proceedings of WIA 99, VIII 1-11, LNCS 2214, 1999, 60–70
5. 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
6. M. Daley, M. Domaratzki, K. Salomaa: State complexity of orthogonal catenation, in: Proceedings of DCFS 08, Charlottetown, 2008, 134–144
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) (2008) 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. M. Holzer, M. Kutrib: State complexity of basic operations on nondeterministic finite automata, in: Proceedings of CIAA 02, LNCS 2608, 2002, 148–157
11. J. E. Hopcroft, R. Motwani, J. D. Ullman: Introduction to Automata Theory, Languages, and Computation (2nd Edition), Addison Wesley, 2001
12. 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
13. G. Jirásková, A. Okhotin: On the state complexity of star of union and star of intersection, TUCS Technical Report No. 825, 2007
14. A. N. Maslov: Estimates of the number of states of finite automata, Soviet Mathematics Doklady 11 (1970) 1373–1375
15. G. Pighizzini, J. O. Shallit: Unary language operations, state complexity and Jacobsthal's function, IJFCS 13 (2002) 145–159
16. M. Rabin, D. Scott: Finite automata and their decision problems, IBM Journal of Research and Development, 3 (2) (1959) 114–125
17. N. Rampersad: The state complexity of $L^2$ and $L^k$, *Information Processing Letters* 98 (2006) 231-234.
18. A. Salomaa, K. Salomaa, S. Yu: State complexity of combined operations, Theoretical Computer Science 383 (2007) 140–152
19. A. Salomaa, K. Salomaa, S. Yu: Undecidability of the state complexity of composed regular operations, in: Proceedings of LATA 2011, LNCS 6638 (2011) 489-498
20. S. Yu: State complexity of regular languages, Journal of Automata, Languages and Combinatorics 6 (2) (2001) 221–234
21. S. Yu, Q. Zhuang, K. Salomaa: The state complexity of some basic operations on regular languages, Theoretical Computer Science 125 (1994) 315–328
22. S. Yu: On the state complexity of combined operations, in: Proceedings of 11th International Conference on Implementation and Application of Automata, Springer LNCS 4094, 2006, 11–22