Department of Computer Science
The University of Western Ontario

Technical Report 715

ISBN: 978-0-7714-2687-2

State complexity of multiple Boolean and catenation operations

April 2008

State complexity of multiple Boolean and catenation operations

Yuan Gao[1], Zoltán Ésik[2,3,*], Guangwu Liu[2,4], and Sheng Yu [1]
[1]Department of Computer Science
University of Western Ontario, London, Ontario, Canada N6A 5B7
[2] Research Group on Mathematical Linguistics
Rovira i Virgili University, Tarragona 43005, Spain
[3]Department of Computer Science
University of Szeged, 6720, Hungary
[4]School of Transportation
Wuhan University of Technology, Wuhan 430063, People's Republic of China

April 25, 2008

Abstract

In the last decade, many new results in the area of state complexity have been obtained. Some are about Boolean operations, e.g. intersection, union and catenation and most of them focused on the cases of these operations on two languages. But in the practical applications, these operations are often required to be performed multiple times on three or more languages. The state complexity of these operations on k languages may not necessarily equal to the k-1 times direct combination of their state complexities on two languages. Thus, it is important to study the state complexities of multiple Boolean operations. Several results about the state complexities of intersection and union on k regular languages are given and proved here. The state complexities are also obtained in the cases of the catenations of three and four languages.

1 Introduction

Automata theory is one of the oldest areas of research in computer science. It is the study of abstract computing devices [11]. Automata theory was started in 1930s, when the machine named computer had not been invented. Up to today, many research activities are still going on in automata research, for example, descriptive complexity of automata and grammars, the power of nondeterminism in finite and pushdown automata, automata implementations and applications and so on [11, 19, 24]. State complexity is one of the active subareas.

In the last decade, a large number of new results have been obtained on the topic of state complexity [1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 15, 16, 17, 18, 22, 23, 27, 28]

*Research supported by grant MTM2007-63422.

following the publication of the paper [25] by S. Yu et al. The state complexities of many Boolean operations have been studied, e.g. intersection, union and catenation. However, most of study on these operations focus on the cases of two languages. In some practical applications, operations are often required to be performed multiple times on three or more languages.

Then a problem comes out. Can we simply combine their state complexities on two languages for $k-1$ times as their state complexities on k languages? To answer this question, we will study three operations here: intersection, union and catenation.

For intersection and union, we will study their state complexities on k languages. Assume there are k regular languages $L_1, L_2, \cdots, L_k$, $k \geq 3$, which are accepted by an n_1-state DFA A_1, an n_2-state DFA $A_2, \cdots$, an n_k-state DFA A_k, respectively. It is easy to see that $n_1 \cdot n_2 \cdot \cdots \cdot n_k$ states are sufficient in the worst case for a DFA to accept the intersection (union) of $L_1, L_2, \cdots, L_k$. We will only study the lower bound. Since $L_i \cup L_j = \overline{\overline{L_i} \cap \overline{L_j}}$ and the DFA accepting $\overline{\overline{L_i} \cap \overline{L_j}}$ ($\overline{L_i}$ or $\overline{L_j}$) has the same number of states with the DFA accepting $\overline{L_i} \cap \overline{L_j}$ (L_i or L_j, respectively), so the state complexity of union and intersection are the same. Thus, we only need to study intersection.

For catenation, we will study its state complexity on three and four regular languages, since when there are five or more regular languages to work with, the notation of state complexity will be very complex and our computers are not powerful enough to finish the experiments. Certainly, given enough time, the state complexity of catenation on five and more regular languages can be achieved, however, the cases of three and four regular languages already shows that we can not combine the state complexity on two languages for $k-1$ times as their state complexities on k languages.

2 Preliminaries

In this section, we will introduce several general definitions and notations.

A deterministic finite automaton (DFA) is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is a finite set of all the input symbols, often called the alphabet, δ is a transition function which makes the DFA transfer from the current state to the next state by reading the current input symbol, i.e., δ: $Q \times \Sigma \mapsto Q$, q_0 is an initial state, $q_0 \in Q$ and F is a set of final states, $F \subseteq Q$.

A DFA is said to be complete if it has transitions defined for each state in Q and each input symbol in Σ. In state complexity research, the DFAs used are complete. So without specific mentioning, all DFAs are assumed to be complete DFAs here.

For a string x and a string y over an alphabet Σ, the catenation of x and y is denoted by xy. It is the string obtained by attaching y to the end of x. Catenation is associative. The length of the new string xy is the sum of the length of x and the length of y. We use $\#(x)$ to denote the length of string x and $\#_a(x)$ to denote the number of letter a's in string x, where $a \in \Sigma$.

For a language L_1 and a language L_2 over an alphabet Σ, the catenation of L_1 and L_2 is denoted by L_1L_2. $L_1L_2 = \{xy | x \in L_1, \quad y \in L_2\}$.

Readers may refer to [10, 11, 20, 21, 26] for more detailed background knowledge in automata theory and formal languages.

The state complexity of a regular language L is the number of states of the minimal

DFA that accepts L. The state complexity of a class of regular languages is the worst among the state complexities of all the languages in the class. The state complexity of a collection of classes of regular languages is usually denoted by a function of the state complexities of the classes.

When we speak about the state complexity of an operation on regular languages, we mean that the state complexity of the resulting languages from the operation [27]. For example, when we say that the state complexity of the union on an m-state DFA language and an n-state DFA language is mn, we mean mn is the state complexity of the class of languages each of which is the resulting language of the union of an m-state DFA language and an n-state DFA language. In another word, there exist two regular languages which are respectively accepted by an m-state DFA and an n-state DFA, such that the union of them is accepted by an mn-state DFA in the worst case. We study worst-case state complexity.

3 State complexity of multiple intersections

The state complexity of intersection (union) on two regular languages is $n_1 \cdot n_2$, if n_1 and n_2 are the number of states of DFAs accepting these two languages, respectively. Then the direct $k-1$ times combination of the state complexity of intersection (union) on two regular languages is $n_1 \cdot n_2 \cdot \cdots \cdot n_k$, where n_i is the number of states of the DFA accepting the language L_i, $1 \le i \le k$. We will show it is indeed the state complexity of intersection (union) on k regular languages.

As stated in the first section, we only need to study the lower bound for intersection here.

Theorem 3.1 *Let $R_1, \ldots, R_k$, $k > 1$, be regular languages, over a one-letter alphabet, accepted by minimal DFAs of $n_1, \ldots, n_k$ states, respectively, where $n_1, \ldots, n_k > 0$ and* $\gcd(n_i, n_j) = 1$ *for any $1 \le i < j \le k$. Then the number of states which is both sufficient and necessary in the worst case for a DFA to accept the intersection of $R_1, \ldots, R_k$ is $n_1 \cdots n_k$.*

gcd is short for greatest common divisor.

Proof. Define a DFA $A_i = (Q_i, \{a\}, \delta_i, 0, \{0\})$, where $L_i = L(A_i)$, $1 \le i \le k$ and for $n_i \ge 2$, $Q_i = \{0, \cdots, n_i - 1\}$. $\gcd(n_i, n_j) = 1$, for $1 \le i < j \le k$. The transitions of A_i are given by

$$\delta_i(t, a) \quad = \quad t + 1 \mod n_i, \quad t = 0, 1, \cdots, n_i - 1.$$

$Fig.$ 1 shows the transition diagram of A_i.

Now we perform intersection operation on L_1 and L_2. $L_1 = \{a^{pn_1} | p \ge 0\}$ and $L_2 = \{a^{qn_2} | q \ge 0\}$. So $L_1 \cap L_2 = \{a^{rn_1n_2} | r \ge 0\}$, since $\gcd(n_1, n_2) = 1$. The DFA B which accepts $L_B = L_1 \cap L_2$ needs at least n_1n_2 states. $B = (Q_B, \{a\}, \delta_B, 0, \{0\})$ where $Q_B = \{0, \cdots, n_1n_2 - 1\}$ and

$$\delta_B(p, a) \quad = \quad p + 1 \mod n_1n_2, \quad p = 0, 1, \cdots, n_1n_2 - 1.$$

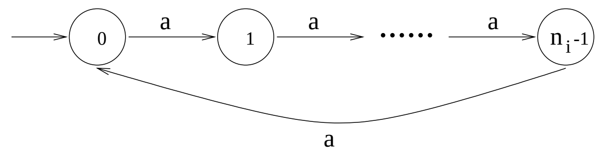

Figure 1: The transition diagram of DFA A_i

It is easy to see that $n_1n_2 \notin \{n_1, n_2, \cdots, n_k\}$ and $\gcd(n_1n_2, u) = 1$, where $u \in \{n_3, \cdots, n_k\}$. Then we perform intersection operation on $L_1 \cap L_2$ and L_3. It is the same with the first step and we get a new DFA C having $n_1n_2n_3$ states which accepts $L_1 \cap L_2 \cap L_3$. Keep doing this until we perform the last intersection on $L_1 \cap \cdots \cap L_{k-1}$ and L_k. We get a new DFA having $n_1 \cdot n_2 \cdot \cdots \cdot n_k$ states which accepts $L_1 \cap L_2 \cap \cdots \cap L_k$. ■

Theorem 3.2 *Let $R_1, \ldots, R_k$, $k > 1$, be regular languages, over a general alphabet, accepted by minimal DFAs of $n_1, \ldots, n_k$ states, respectively, where $n_1, \ldots, n_k > 0$ and $\gcd(n_i, n_j) = 1$ for any $1 \le i < j \le k$. Then the number of states which is both sufficient and necessary in the worst case for a DFA to accept the intersection of $R_1, \ldots, R_k$ is $n_1 \cdots n_k$.*

This theorem stands for languages over an alphabet with any number of letters. We have proved the case of unary languages. Here, we prove the case that the alphabet has more than one letter.

Proof. Define a DFA $A_i = (Q_i, \Sigma, \delta_i, 0, \{0\})$, where $L_i = L(A_i)$, $1 \le i \le k$, $n_i \ge 2$, $Q_i = \{0, \cdots, n_i - 1\}$, $|\Sigma| > 1$, $a \in \Sigma$. $\gcd(n_i, n_j) = 1$, for $1 \le i < j \le k$. The transitions of A_i are given by

$$\begin{aligned} \delta_i(t, a) &= t + 1 \mod n_i, \quad t = 0, 1, \cdots, n_i - 1; \\ \delta_i(t, b) &= t, \quad t = 0, 1, \cdots, n_i - 1, \quad b \in \Sigma - \{a\}. \end{aligned}$$

Fig. 2 shows the transition diagram of A_i.

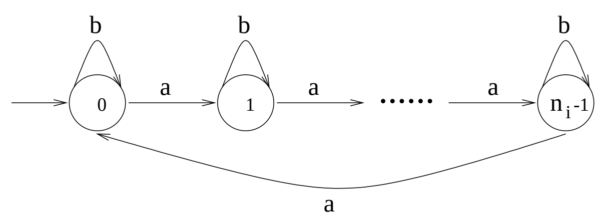

Figure 2: The transition diagram of DFA A_i

We perform intersection operation on L_1 and L_2. $L_1 = \{w | \#_a(w) \mod n_1 = 0\}$ and $L_2 = \{w | \#_a(w) \mod n_2 = 0\}$. So $L_1 \cap L_2 = \{w | \#_a(w) \mod n_1n_2 = 0\}$, since $\gcd(n_1, n_2) = 1$. The DFA B which accepts $L_B = L_1 \cap L_2$ needs at least n_1n_2 states. $B = (Q_B, \Sigma, \delta_B, 0, \{0\})$ where $Q_B = \{0, \cdots, n_1n_2 - 1\}$ and

$$\begin{aligned} \delta_B(p, a) &= p + 1 \mod n_1n_2, \quad p = 0, 1, \cdots, n_1n_2 - 1; \\ \delta_B(p, b) &= p, \quad p = 0, 1, \cdots, n_1n_2 - 1, \quad b \in \Sigma - \{a\}. \end{aligned}$$

It is easy to see that $n_1 n_2 \notin \{n_t | 1 \leq t \leq k\}$ and $\gcd(n_1 n_2, u) = 1$, where $u \in \{n_s | 3 \leq s \leq k\}$. Then we perform intersection operation on $L_1 \cap L_2$ and L_3. It is the same with the first step. Keep doing this until we perform the last intersection on $L_1 \cap \cdots \cap L_{k-1}$ and L_k. We get a new DFA having $n_1 \cdot n_2 \cdot \cdots \cdot n_k$ states which accepts $L_1 \cap L_2 \cap \cdots \cap L_k$. ∎

Theorem 3.3 *Let Σ be a two-letter alphabet and $R_1, \ldots, R_k$, $k \geq 2$, be k regular languages over Σ accepted by minimal DFAs of $n_1, \ldots, n_k$ states, respectively, $n_1, \ldots, n_k > 1$. If the k languages can be partitioned into two sets $\{R_1, \ldots, R_l\}$ and $\{R_{l+1}, \ldots, R_k\}$ for some l, $1 \leq l < k$, such that both $\{n_1, \ldots, n_l\}$ and $\{n_{l+1}, \ldots, n_k\}$ are mutually prime, then the state complexity of $R_1 \cap \cdots \cap R_k$ is $n_1 \cdots n_k$.*

Proof. It is clear that $n_1 \cdots n_k$ is an upper bound. In the following, we show that $n_1 \cdots n_k$ is also a lower bound.

Assume that a set of integers $\{n_1, n_2, \cdots, n_k\}$, $n_i > 1$, $1 \leq i \leq k$ can be divided into two sets M and N such that $\gcd(n_e, n_f) = 1$ for any $n_e, n_f \in M$, $e \neq f$, $\gcd(n_g, n_h) = 1$ for any $n_g, n_h \in N$, $g \neq h$. We construct k DFAs as follows.

For each $n_i \in M$, define a DFA $A_i = (Q_i, \{a, b\}, \delta_i, 0, \{0\})$, where $Q_i = \{0, \cdots, n_i - 1\}$ and δ_i is given by

$$\begin{aligned} \delta_i(t, a) &= t + 1 \mod n_i, \quad t = 0, 1, \cdots, n_i - 1; \\ \delta_i(t, b) &= t, \quad t = 0, 1, \cdots, n_i - 1. \end{aligned}$$

We denote $L(A_i)$ by L_i.

Similarly for each $n_p \in N$, define a DFA $A_p = (Q_p, \{a, b\}, \delta_p, 0, \{0\})$, where $Q_p = \{0, \cdots, n_p - 1\}$ and δ_p is given by

$$\begin{aligned} \delta_p(t, b) &= t + 1 \mod n_p, \quad t = 0, 1, \cdots, n_p - 1; \\ \delta_p(t, a) &= t, \quad t = 0, 1, \cdots, n_p - 1. \end{aligned}$$

We denote $L(A_p)$ by L_p.

It is easy to show that the following DFA is the minimal DFA that accepts the intersection of all L_i such that $n_i \in M$: $C = (Q_C, \{a, b\}, \delta_C, 0, \{0\})$ where

$$\begin{aligned} Q_C &= \{0, 1, \cdots, \prod_{n_e \in M} n_e - 1\}; \\ \delta_C(t, a) &= t + 1 \mod \prod_{n_e \in M} n_e, \quad t = 0, 1, \cdots, \prod_{n_e \in M} n_e - 1; \\ \delta_C(t, b) &= t, \quad t = 0, 1, \cdots, \prod_{n_e \in M} n_e - 1. \end{aligned}$$

Analogously, we have the following minimal DFA that accepts the intersection of languages L_p such that $n_p \in N$: $D = (Q_D, \{a, b\}, \delta_D, 0, \{0\})$ where

$$\begin{aligned} Q_D &= \{0, 1, \cdots, \prod_{n_g \in N} n_g - 1\}; \\ \delta_D(t, b) &= t + 1 \mod \prod_{n_g \in N} n_g, \quad t = 0, 1, \cdots, \prod_{n_g \in N} n_g - 1; \\ \delta_D(t, a) &= t, \quad t = 0, 1, \cdots, \prod_{n_g \in N} n_g - 1. \end{aligned}$$

Now we have

$$L(C) = \{w | w \in \{a,b\}^*, \ \#_a(w) \mod \prod_{n_e \in M} n_e = 0\};$$
$$L(D) = \{w | w \in \{a,b\}^*, \ \#_b(w) \mod \prod_{n_g \in N} n_g = 0\};$$

Clearly, we have

$$L(C) \cap L(D) = \{w | w \in \{a,b\}^*, \ \#_a(w) \mod \prod_{n_e \in M} n_e = 0, \ \#_b(w) \mod \prod_{n_g \in N} n_g = 0\};$$

Define DFA $E = (Q_E, \{a,b\}, \delta_E, <0,0>, \{<0,0>\})$, where

$$Q_E = \{<X,Y> | X \in Q_C, Y \in Q_D\};$$
$$\delta_E(<X,Y>, a) = <\delta_C(X,a), \delta_D(Y,a)>;$$
$$\delta_E(<X,Y>, b) = <\delta_C(X,b), \delta_D(Y,b)>.$$

It is easy to see that $L(E) = L(C) \cap L(D)$. Now we will show that E is minimal.

1. For each state $<X,Y> \in Q_E$, $\delta_E(<0,0>, a^X b^Y) = <X,Y>$. So every state in Q_E is reachable.

2. For any two different states $<X_1,Y_1>$ and $<X_2,Y_2>$ in Q_E, if $X_1 \neq X_2$ or $Y_1 \neq Y_2$, then

$$\delta_E(<X_1,Y_1>, a^{|Q_C|-X_1} b^{|Q_D|-Y_1}) = <0,0>;$$
$$\delta_E(<X_2,Y_2>, a^{|Q_C|-X_1} b^{|Q_D|-Y_1}) \neq <0,0>.$$

   So any two distinct states of E are not equivalent.

Thus, E is the minimal DFA accepting $L_1 \cap L_2 \cap \cdots \cap L_k$. ■

This result can be easily extended to languages over an arbitrary t-letter alphabet, $t \geq 2$, in the following.

Corollary 3.1 *Let Σ be a t-letter alphabet, $t \geq 2$, and $R_1, \ldots, R_k$, $k \geq 2$, be k regular languages over Σ accepted by DFAs of $n_1, \ldots, n_k$ states, respectively. If the k languages can be partitioned into t sets, $1 \leq t \leq k$, and all the numbers of states of the DFAs that accept the languages in each set are mutually prime, then the state complexity of intersection of all the k languages is $n_1 \cdots n_k$.*

A further improvement of Theorem 3.3 is stated in the following.

Theorem 3.4 *Let Σ be a two-letter alphabet and $R_1, \ldots, R_k, R_{k+1}$, $k \geq 2$, be $k+1$ regular languages over Σ accepted by DFAs of $n_1, \ldots, n_k$ states, respectively. If the first k languages can be partitioned into two sets $\{R_1, \ldots, R_l\}$ and $\{R_{l+1}, \ldots, R_k\}$ for some l, $1 \leq l < k$, such that both $\{n_1, \ldots, n_l\}$ and $\{n_{l+1}, \ldots, n_k\}$ are mutually prime, then the state complexity of $R_1 \cap \cdots \cap R_k \cap R_{k+1}$ is $n_1 \cdots n_k n_{k+1}$.*

Proof. It is clear that $n_1 \cdots n_{k+1}$ is an upper bound. In the following, we show that $n_1 \cdots n_{k+1}$ is also a lower bound.

The first part of the proof of this theorem is the same with that of Corollary 3.1. Assume that a set of integers $\{n_1, n_2, \cdots, n_k\}$, $n_i > 1$, $1 \le i \le k$ can be divided into two sets M and N such that $\gcd(n_e, n_f) = 1$ for any $n_e, n_f \in M$, $e \ne f$, $\gcd(n_g, n_h) = 1$ for any $n_g, n_h \in N$, $g \ne h$. Then construct DFA C accepting the intersection of all R_e, $n_e \in M$ and DFA D accepting the intersection of all R_g, $n_g \in N$. Let $u = |Q_C|$ and $v = |Q_D|$.

Define an n_{k+1}-state DFA $F = \{Q_F, \{a, b\}, \delta_F, 0, \{0\}\}$ where $Q_F = \{0, 1, \cdots, n_{k+1} - 1\}$ and δ_F is given by

$$\begin{aligned} \delta_F(0, b) &= 1; \quad \delta_F(0, a) = 0; \\ \delta_F(1, b) &= 2; \quad \delta_F(1, a) = 1; \\ \delta_F(t, a) &= t + 1 \mod n_{k+1}, \quad t = 2, \cdots, n_{k+1} - 1; \\ \delta_F(t, b) &= t, \quad t = 2, \cdots, n_{k+1} - 1. \end{aligned}$$

Fig. 3 shows the transition diagram of F. We denote $L(F)$ by R_{k+1}.

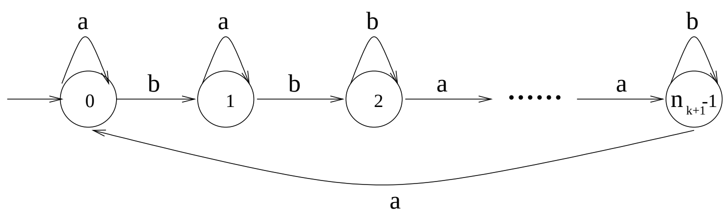

Figure 3: The transition diagram of DFA F

Define DFA $G = \{Q_G, \{a, b\}, \delta_G, <0, 0, 0>, \{<0, 0, 0>\}\}$ where

$$\begin{aligned} Q_G &= \{<X, Y, Z> | X \in Q_C, Y \in Q_D, Z \in Q_F\}; \\ \delta_G(<X, Y, Z>, a) &= <\delta_C(X, a), \delta_D(Y, a), \delta_F(Z, a)>; \\ \delta_G(<X, Y, Z>, b) &= <\delta_C(X, b), \delta_D(Y, b), \delta_F(Z, b)>. \end{aligned}$$

It is easy to see that $L(G) = L(C) \cap L(D) \cap L(F) = R_1 \cap \cdots \cap R_k \cap R_{k+1}$. Now we will check if G is a minimal DFA.

1. For any state $<X, Y, Z> \in Q_G$, $Z \ne 0, 1, 2$, $\delta_G(<0, 0, 0>, a^{n_{k+1}+T} b^{v+Y} a^{Z-2}) = <X, Y, Z>$ where T is a positive integer such that $(n_{k+1} + T + Z - 2) \mod u = X$.

   For $<X, Y, Z> \in Q_G$, $Z = 0$ or 1 or 2, $\delta_G(<0, 0, 0>, a^T b^{n_{k+1}v+Y-Z} a^{n_{k+1}-2} b^Z) = <X, Y, Z>$ where T is a positive integer such that $(n_{k+1} + T - 2) \mod u = X$.

   So every state in Q_G is reachable.

2. $<X_1, Y_1, Z_1>, <X_2, Y_2, Z_2> \in Q_G$ are two different states.

   (1) $X_1 \ne X_2$ or $Y_1 \ne Y_2$

   $$\begin{aligned} \delta_G(<X_1, Y_1, Z_1>, a^{n_{k+1}+T} b^{2v-Y_1} a^{n_{k+1}-2}) &= <0, 0, 0>, \\ \delta_G(<X_2, Y_2, Z_2>, a^{n_{k+1}+T} b^{2v-Y_1} a^{n_{k+1}-2}) &\ne <0, 0, 0>, \end{aligned}$$

   where T is a positive integer such that $(2n_{k+1} + T - 2) \mod u = u - X_1$.

(2) $X_1 = X_2,\ Y_1 = Y_2,\ Z_1 \neq Z_2$

(I) $Z_1 \geq 0,\ Z_2 > 2,\ Z_2 > Z_1,\ n_{k+1} > 3$

Let $t_1 = b^{2v-Y_1-1}a^{n_{k+1}-Z_2}ba^{n_{k+1}+T}$, where T is a positive integer such that $(2n_{k+1} - Z_2 + T) \bmod u = u - X_1$. Then

$$\begin{aligned} \delta_G(< X_1, Y_1, 0 >, t_1) &= < 0, 0, 0 >, \\ \delta_G(< X_2, Y_2, Z_2 >, t_1) &\neq < 0, 0, 0 >, \end{aligned}$$

If $Z_1 > 2$, $Z_1 > Z_2$,$Z_2 \geq 0$, $t_1' = b^{2v-Y_2-1}a^{n_{k+1}-Z_1}ba^{n_{k+1}+T}$ can distinguish these two states, where T is a positive integer such that $(2n_{k+1} - Z_1 + T) \bmod u = u - X_1$.

(II) $Z_1 = 0$, $Z_2 = 1$ or 2, $n_{k+1} \geq 3$

Let $t_2 = ba^T(a^{n_{k+1}}b)^{2v-Y_1-1}a^{n_{k+1}}$, where T is a positive integer such that $(T + n_{k+1}(2v - Y_1)) \bmod u = u - X_1$. Then one of $\delta_G(< X_1, Y_1, 0 >, t_2)$ and $\delta_G(< X_2, Y_2, Z_2 >, t_2)$ is $< 0, 0, 0 >$ but the other is not.

If $Z_1 = 1$ or 2 and $Z_2 = 0$, t_2 can also distinguish these two states.

(III) $Z_1 = 1,\ Z_2 = 2,\ n_{k+1} \geq 3$

Let $t_3 = a^{n_{k+1}+T}b(a^{n_{k+1}}b)^{2v-Y_1-1}a^{n_{k+1}}$, where T is a positive integer such that $(T + n_{k+1}(2v - Y_1 + 1)) \bmod u = u - X_1$. Then one of $\delta_G(< X_1, Y_1, 1 >, t_3)$ and $\delta_G(< X_2, Y_2, Z_2 >, t_3)$ is $< 0, 0, 0 >$ but the other is not.

If $Z_1 = 2$ and $Z_2 = 1$, t_3 can also distinguish these two states.

So any two states of G are distinguishable.

Thus, G is the minimal DFA accepting $R_1 \cap R_2 \cap \cdots \cap R_k \cap R_{k+1}$ has $n_1 \cdot n_2 \cdot \cdots \cdot n_{k+1}$ states. ■

4 State complexity of multiple catenations

There are three regular languages $L(A), L(B), L(C)$, which are accepted by an m-state DFA A, an n-state DFA B and a p-state DFA C, respectively. In this section, we will study the number of states of the DFA accepting the $L(A)L(B)L(C)$. The state complexity of catenation on two regular languages is $m2^n - 2^{n-1}$. Then the direct two times combination of the state complexity of catenation on two regular languages is $m2^{n+p} - 2^{n+p-1} - 2^{p-1}$. We will show it is not the state complexity of catenation on three regular languages.

We first consider the lower bound.

Theorem 4.1 *For any integers $m, n, p \geq 2$, there exist DFAs A, B, and C of m, n, and p states, respectively, such that any DFA accepting $L(A)L(B)L(C)$ needs at least $m2^{n+p} - 2^{n+p-1} - (m-1)2^{n+p-2} - 2^{n+p-3} - (m-1)(2^p - 1)$ states.*

Proof. Let $\Sigma = \{a, b, c, d, e\}$ in the following. Define a DFA $A = (Q_A, \Sigma, \delta_A, 0, \{m-1\})$, where for $m > 1$, $Q_A = \{0, \cdots, m-1\}$. The transition function of A is as follows. For $t = 0, 1, \cdots, m-1$, $\delta_A(t, a) = t + 1 \bmod m$, $\delta_A(t, x) = t, x \in \{b, c, e\}$ and $\delta_A(t, d) = 0$. *Fig.* 4 shows the transition diagram of A.

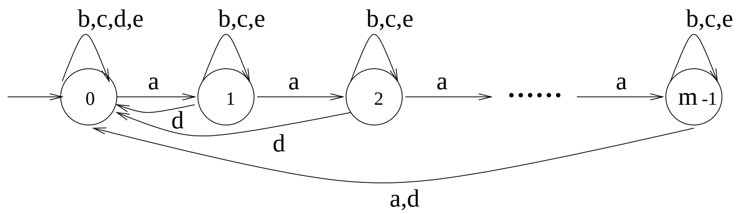

Figure 4: The transition diagram of DFA A

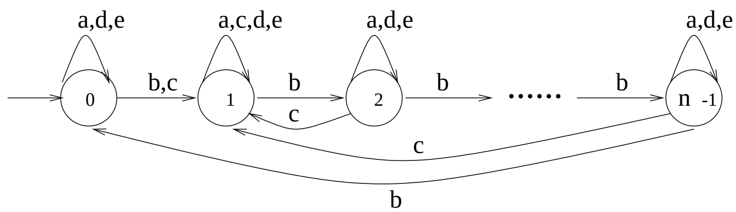

Figure 5: The transition diagram of DFA B

Define a DFA $B = (Q_B, \Sigma, \delta_B, 0, \{n-1\})$, where for $n > 1$, $Q_B = \{0, \cdots, n-1\}$. For $t = 0, 1, \cdots, n-1$, $\delta_B(t, b) = t + 1 \bmod n$, $\delta_B(t, y) = t, y \in \{a, d, e\}$ and $\delta_B(t, c) = 1$. *Fig.* 5 shows the transition diagram of B.

Define a DFA $C = (Q_C, \Sigma, \delta_C, 0, \{p-1\})$, where for $p > 1$, $Q_C = \{0, \cdots, p-1\}$. For $t = 0, 1, \cdots, p-1$, $\delta_C(t, d) = t + 1 \bmod p$, $\delta_C(t, z) = t, z \in \{a, b, c\}$ and $\delta_C(t, e) = 1$. *Fig.* 6 shows the transition diagram of C.

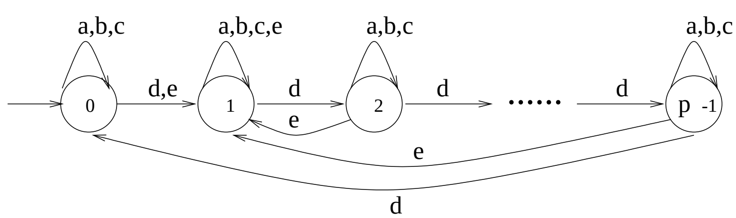

Figure 6: The transition diagram of DFA C

For each $x \in \{a, b, d\}^*$, we define

$$S(x) = \{i | x = uvw \text{ such that } u \in L(A), \; v \in L(B), \text{ and } i = \#_d(w) \bmod p\}.$$

Consider that $x, y \in \{a, b, d\}^*$ such that $S(x) \neq S(y)$. Let $k \in S(x) - S(y)$(or $S(y) - S(x)$). Then it is clear that $xd^{p-1-k} \in L(A)L(B)L(C)$ but $yd^{p-1-k} \notin L(A)L(B)L(C)$. So, x and y are in different equivalence classes of the right-invariant relation induced by $L(A)L(B)L(C)$.

For each $x \in \{a, b, d\}^*$, we define

$$T(x) = \{i | x = uv \text{ such that } u \in L(A), \text{ and } i = \#_b(v) \bmod n\}.$$

Consider that $x, y \in \{a, b, d\}^*$ such that $T(x) \neq T(y)$. Let $k \in T(x) - T(y)$(or $T(y) - T(x)$). Then it is clear that $xb^{n-1-k}ed^{p-1} \in L(A)L(B)L(C)$ but $yb^{n-1-k}ed^{p-1} \notin L(A)L(B)L(C)$.

So, x and y are in different equivalence classes of the right-invariant relation induced by $L(A)L(B)L(C)$.

For each $x \in \{a, b, d\}^*$, define

$$\begin{aligned} R(x) &= \#_a(z) \text{ where } x = ydz, y \in \{a,b,d\}^*, z \in \{a,b\}^*, \quad \text{if } d \text{ occurs in } x; \\ R(x) &= \#_a(x), \quad \text{otherwise.} \end{aligned}$$

Consider $u, v \in \{a, b, d\}^*$ such that $S(u) = S(v)$, $T(u) = T(v)$ and $R(u) \bmod m > R(v) \bmod m$. Let $i = R(u) \bmod m$ and $w = a^{m-1-i}cb^{n-1}ed^{p-1}$. Then clearly $uw \in L(A)L(B)L(C)$ but $vw \notin L(A)L(B)L(C)$.

Notice that there does not exist a word $w \in \Sigma^*$ such that $0 \notin T(w)$ and $R(w) = m-1$, since $R(w) = m-1$ guarantees that $0 \in T(w)$. Because of the same reason, there does not exist a word $w \in \Sigma^*$ such that $n-1 \in T(w)$ and $0 \notin S(w)$. It is also impossible that $T(w) = \emptyset$ but $S(w) \neq \emptyset$.

For each subset $s = \{i_1, \cdots, i_k\}$ of $\{0, \cdots, p-1\}$ and each subset $t = \{j_1, \cdots, j_l\}$ of $\{0, \cdots, n-1\}$ where $i_1 > \cdots > i_k$ and $j_1 > \cdots > j_l$, and an integer $r \in \{0, \cdots, m-1\}$, except the following three cases (1) $0 \notin t$ and $r = m-1$, (2) $0 \notin s$ and $n-1 \in t$, and (3) $r \neq m-1$, $s \neq \emptyset$ and $t = \emptyset$, there exists a word

$$\begin{aligned} x = \; & a^m b^n d^{i_1 - i_2} a^m b^n d^{i_2 - i_3} \cdots a^m b^n d^{i_{k-1} - i_k} a^m b^n d^{i_k} \\ & a^m b^{j_1 - j_2} a^m b^{j_2 - j_3} \cdots a^m b^{j_{l-1} - j_l} a^m b^{j_l} a^r. \end{aligned}$$

such that $S(x) = s$, $T(x) = t$ and $R(x) = r$. Totally, there are $m2^n2^p$ classes. There are $2^{n-1}2^p$ classes with both $0 \notin t$ and $r = m-1$. Notice that the classes with $r = m-1$, $0 \notin t$, $n-1 \in t$ and $0 \notin s$ have already been included in these $2^{n-1}2^p$ classes. So there are only $(m-1)2^{n-1}2^{p-1} + 2^{n-2}2^{p-1}$ classes with both $0 \notin s$ and $n-1 \in t$. And there are $(m-1)(2^p - 1)$ classes with $r \neq m-1$, $s \neq \emptyset$ and $t = \emptyset$. Thus, there are at least

$$m2^{n+p} - 2^{n+p-1} - (m-1)2^{n+p-2} - 2^{n+p-3} - (m-1)(2^p - 1)$$

distinct equivalence classes. ■

We now show an upper bound of this combined operation.

Theorem 4.2 *Let A, B and C be three DFAs of m, n, and p states, respectively, $m, n, p > 0$, where A has k final states and B has l final states, $0 < k < m$ and $0 < j < n$. Then there exists a DFA of $(2m-k)2^{n+p-2} + (2m-k)2^{n+p-l-2} - (m-k)(2^p - 1)$ states that accepts $L(A)L(B)L(C)$.*

Proof. Let $A = (Q_A, \Sigma, \delta_A, r_0, F_A)$, $B = (Q_B, \Sigma, \delta_B, s_0, F_B)$ and $C = (Q_C, \Sigma, \delta_C, t_0, F_C)$. Construct $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ such that

$$\begin{aligned} Q_E = \; & Q_A \times 2^{Q_B} \times 2^{Q_C} - F_A \times 2^{Q_B - \{s_0\}} \times 2^{Q_C} \\ & -(Q_A - F_A) \times (2^{F_B} - \{\emptyset\}) \times 2^{Q_B - F_B} \times 2^{Q_C - \{t_0\}} \\ & -F_A \times (2^{F_B} - \{\emptyset\}) \times 2^{Q_B - F_B - \{s_0\}} \times 2^{Q_C - \{t_0\}} - (Q_A - F_A) \times \emptyset \times (2^{Q_C} - \{\emptyset\}). \\ q_0 = \; & < r_0, \emptyset, \emptyset >, \quad \text{if } r_0 \notin F_A \text{ and } s_0 \notin F_B; \\ = \; & < r_0, \{s_0\}, \emptyset >, \quad \text{if } r_0 \in F_A \text{ and } s_0 \notin F_B; \end{aligned}$$

$$
\begin{array}{rcl}
& = & < r_0, \{s_0\}, \{t_0\} >, \ \ \text{if} \ \ r_0 \in F_A \ \ \text{and} \ \ s_0 \in F_B; \\
F_E & = & \{< r, S, T > \in Q_E | T \cap F_C \neq \emptyset\}; \\
\delta_E & : & \delta_E(< r, S, T >, a) = < r', S', T' >, \ \ \text{for} \ \ a \in \Sigma, \ \ \text{where} \ \ r' = \delta_A(r, a), \\
& & S' = \delta_B(S, a) \cup \{s_0\} \ \ \text{if} \ \ r' \in F_A, \ \ S' = \delta_B(S, a) \ \ \text{otherwise}, \\
& & T' = \delta_C(T, a) \cup \{t_0\} \ \ \text{if} \ \ S' \cap F_B \neq \emptyset, \ \ T' = \delta_C(T, a) \ \ \text{otherwise}.
\end{array}
$$

Intuitively, Q_E is a set of 3-tuples whose first component is a state in Q_A, the second component is a subset of Q_B, and the last component is a subset of Q_C.

Q_E does not contain those 3-tuples whose first component is a final state of A and whose second component does not contain s_0, the initial state of B.

Q_E does not contain those 3-tuples whose second component contains at least one final state of B and whose third component does not contain t_0, the initial state of C. Notice that the 3-tuples whose first component is a final state of A and whose second component contains at least one final state of B but does not contain s_0 and whose last component does not contain t_0 has been included in the first case.

Q_E also does not contain those 3-tuples whose first component is a non-final state of A and whose second component is $\emptyset$ and whose last component is nonempty.

Clearly, $L(E) = L(A)L(B)L(C)$. Let $|Q_A| = m$, $|Q_B| = n$, $|Q_C| = p$, $|F_A| = k$ and $|F_B| = l$. Then E has $(2m-k)2^{n+p-2} + (2m-k)2^{n+p-l-2} - (m-k)(2^p - 1)$ states. ■

Note that when $k = 1$ and $l = 1$, i.e, A and B each have one final state, this upper bound is exactly the same as the lower bound stated in Theorem 4.1. Thus, this bound is the state complexity of the catenation of three regular languages.

Now we consider the catenation of four regular languages. There are four regular languages $L(A), L(B), L(C), L(D)$, which are accepted by an m-state DFA A, an n-state DFA B, a p-state DFA C and a q-state DFA D, respectively. We will study the number of states of the DFA accepting the $L(A)L(B)L(C)L(D)$. The direct three times combination of the state complexity of catenation on two regular languages is $m2^{n+p+q} - 2^{n+p+q-1} - 2^{p+q-1} - 2^{q-1}$. We will show it is not the state complexity of catenation on four regular languages.

We still consider the lower bound first.

Theorem 4.3 *For any integers $m, n, p, q \geq 2$, there exist DFAs A, B, C and D of m, n, p and q states, respectively, such that any DFA accepting $L(A)L(B)L(C)L(D)$ needs at least $9(2m-1)2^{n+p+q-5} - 3(m-1)2^{p+q-2} - (2m-1)2^{n+q-2} + (m-1)2^q + (2m-1)2^{n-2}$ states.*

Proof. Let $\Sigma = \{a, b, c, d, e, f, g\}$ in the following. Define a DFA $A = (Q_A, \Sigma, \delta_A, 0, \{m-1\})$, where for $m > 1$, $Q_A = \{0, \cdots, m-1\}$. For $t = 0, 1, \cdots, m-1$, $\delta_A(t, a) = t + 1 \bmod m$, $\delta_A(t, x) = t, x \in \{b, c, d, e, g\}$ and $\delta_A(t, f) = 0$. *Fig.* 7 shows the transition diagram of A.

Define a DFA $B = (Q_B, \Sigma, \delta_B, 0, \{n-1\})$, where for $n > 1$, $Q_B = \{0, \cdots, n-1\}$. For $t = 0, 1, \cdots, n-1$, $\delta_B(t, b) = t + 1 \bmod n$, $\delta_B(t, y) = t, y \in \{a, d, e, f, g\}$ and $\delta_B(t, c) = 1$. *Fig.* 8 shows the transition diagram of B.

Define a DFA $C = (Q_C, \Sigma, \delta_C, 0, \{p-1\})$, where for $p > 1$, $Q_C = \{0, \cdots, p-1\}$. For $t = 0, 1, \cdots, p-1$, $\delta_C(t, d) = t + 1 \bmod p$, $\delta_C(t, z) = t, z \in \{a, b, c, f, g\}$ and $\delta_C(t, e) = 1$. *Fig.* 9 shows the transition diagram of C.

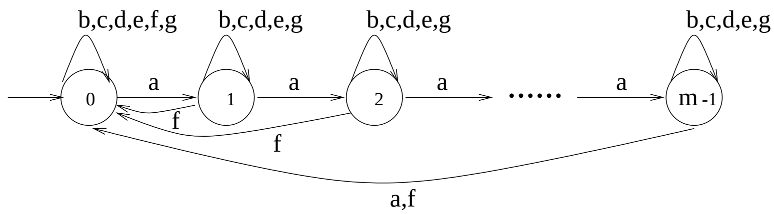

Figure 7: The transition diagram of DFA A

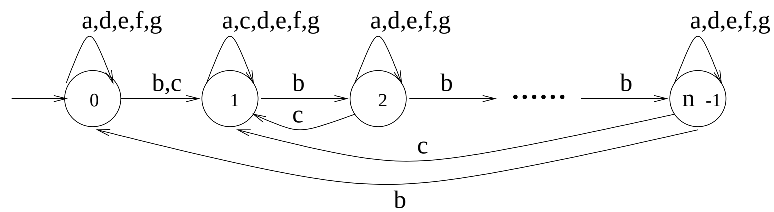

Figure 8: The transition diagram of DFA B

Define a DFA $D = (Q_D, \Sigma, \delta_D, 0, \{q-1\})$, where for $q > 1$, $Q_D = \{0, \cdots, q-1\}$. For $t = 0, 1, \cdots, q-1$, $\delta_D(t, f) = t + 1 \bmod q$, $\delta_D(t, w) = t, w \in \{a, b, c, d, e\}$ and $\delta_D(t, g) = 1$. $Fig.$ 10 shows the transition diagram of D.

For each $x \in \{a, b, d, f\}^*$, we define

$$H(x) = \{i | x = uvwz \text{ such that } u \in L(A), \ v \in L(B), \ w \in L(C) \text{ and } i = \#_f(z) \bmod q\}.$$

Consider that $x, y \in \{a, b, d, f\}^*$ such that $H(x) \neq H(y)$. Let $k \in H(x) - H(y)$(or $H(y) - H(x)$). Then it is clear that $xf^{q-1-k} \in L(A)L(B)L(C)L(D)$ but $yf^{q-1-k} \notin L(A)L(B)L(C)L(D)$. So, x and y are in different equivalence classes of the right-invariant relation induced by $L(A)L(B)L(C)L(D)$.

For each $x \in \{a, b, d, f\}^*$, we define

$$S(x) = \{i | x = uvw \text{ such that } u \in L(A), \ v \in L(B), \text{ and } i = \#_d(w) \bmod p\}.$$

Consider that $x, y \in \{a, b, d, f\}^*$ such that $S(x) \neq S(y)$. Let $k \in S(x) - S(y)$(or $S(y) - S(x)$). Then it is clear that $xd^{p-1-k}gf^{q-1} \in L(A)L(B)L(C)L(D)$ but $yd^{p-1-k}gf^{q-1} \notin L(A)L(B)L(C)L(D)$. So, x and y are in different equivalence classes of the right-invariant relation induced by $L(A)L(B)L(C)L(D)$.

For each $x \in \{a, b, d, f\}^*$, we define

$$T(x) = \{i | x = uv \text{ such that } u \in L(A), \text{ and } i = \#_b(v) \bmod n\}.$$

Consider that $x, y \in \{a, b, d, f\}^*$ such that $T(x) \neq T(y)$. Let $k \in T(x) - T(y)$(or $T(y) - T(x)$). Then it is clear that $xb^{n-1-k}ed^{p-1}gf^{q-1} \in L(A)L(B)L(C)L(D)$ but $yb^{n-1-k}ed^{p-1}gf^{q-1} \notin L(A)L(B)L(C)L(D)$. So, x and y are in different equivalence classes of the right-invariant relation induced by $L(A)L(B)L(C)L(D)$.

For each $x \in \{a, b, d, f\}^*$, define

$$R(x) = \#_a(z), \quad x = yfz, y \in \{a, b, d, f\}^*, z \in \{a, b, d\}^*, \text{ if } f \text{ occurs in } x;$$
$$R(x) = \#_a(x), \quad \text{otherwise.}$$

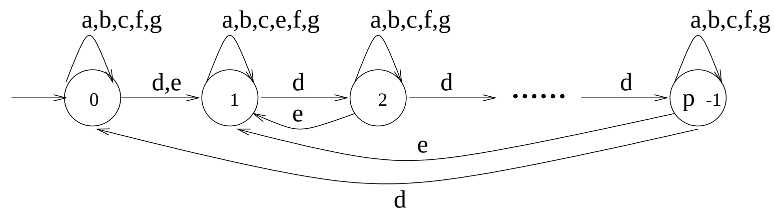

Figure 9: The transition diagram of DFA C

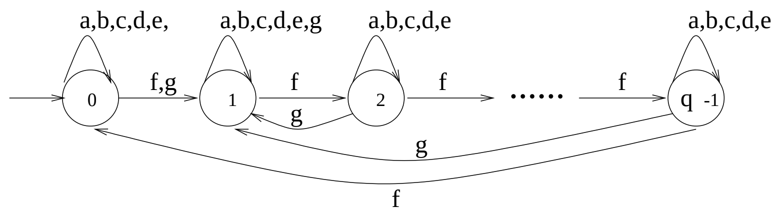

Figure 10: The transition diagram of DFA D

Consider $u, v \in \{a, b, d, f\}^*$ such that $S(u) = S(v)$, $T(u) = T(v)$, $H(u) = H(v)$ and $R(u) \bmod m > R(v) \bmod m$. Let $i = R(u) \bmod m$ and $w = a^{m-1-i}cb^{n-1}ed^{p-1}gf^{q-1}$. Then clearly $uw \in L(A)L(B)L(C)L(D)$ but $vw \notin L(A)L(B)L(C)L(D)$.

Notice that there does not exist a word $w \in \Sigma^*$ such that $0 \notin T(w)$ and $R(w) = m-1$, since the fact that $R(w) = m-1$ guarantees that $0 \in T(w)$. Because of the same reason, there does not exist a word $w \in \Sigma^*$ such that $0 \notin S(w)$ and $n-1 \in T(w)$. There does not exist a word $w \in \Sigma^*$ such that $0 \notin H(w)$ and $n-1 \in S(w)$, neither. It is impossible that $T(w) = \emptyset$ but $S(w)$ or $H(w) \neq \emptyset$. It is also impossible that $S(w) = \emptyset$ and $H(w) \neq \emptyset$.

For each subset $h = \{u_1, \cdots, u_v\}$ of $\{0, \cdots, q-1\}$, each subset $s = \{i_1, \cdots, i_k\}$ of $\{0, \cdots, p-1\}$ and each subset $t = \{j_1, \cdots, j_l\}$ of $\{0, \cdots, n-1\}$ where $u_1 > \cdots > u_v$, $i_1 > \cdots > i_k$ and $j_1 > \cdots > j_l$, and an integer $r \in \{0, \cdots, m-1\}$ except the case when $0 \notin s$ and $n-1 \in t$, the case when $0 \notin t$ and $r = m-1$, the case when $0 \notin h$ and $n-1 \in s$, the case when s or $h \neq \emptyset$ and $t = \emptyset$ and the case when $h \neq \emptyset$ and $s = \emptyset$, there exists a word

$$\begin{aligned} x &= a^m b^n d^p f^{u_1-u_2} a^m b^n d^p f^{u_2-u_3} \cdots a^m b^n d^p f^{u_{v-1}-u_v} a^m b^n d^p f^{u_v} \\ &\quad a^m b^n d^{i_1-i_2} a^m b^n d^{i_2-i_3} \cdots a^m b^n d^{i_{k-1}-i_k} a^m b^n d^{i_k} \\ &\quad a^m b^{j_1-j_2} a^m b^{j_2-j_3} \cdots a^m b^{j_{l-1}-j_l} a^m b^{j_l} a^r. \end{aligned}$$

such that $x = k \bmod l$, $y = v \bmod l$, $h = v \bmod k$, $H(x) = h$, $S(x) = s$, $T(x) = t$ and $R(x) = r$.

Totally, there are $m2^n2^p2^q$ classes. There are $2^{n-1}2^p2^q$ classes with both $0 \notin t$ and $r = m-1$. Notice that the classes with $r = m-1$, $0 \notin t$, $n-1 \in t$ and $0 \notin s$ have already been included in these $2^{n-1}2^p2^q$ classes. So there are only $(m-1)2^{n-1}2^{p-1}2^q + 2^{n-2}2^{p-1}2^q$ classes with both $0 \notin s$ and $n-1 \in t$. There are $(m-1)(2^{n-1}-1)2^{p-1}2^{q-1} + 2^{n-2}2^{p-1}2^{q-1}$ classes with $n-1 \notin t$, $p-1 \in s$ and $0 \notin h$. There are $(m-1)2^{n-1}2^{p-2}2^{q-1} + 2^{n-2}2^{p-2}2^{q-1}$ classes with $n-1 \in t$, $p-1 \in s$ and $0 \notin h$. And there are $(m-1)(2^p2^q-1)$ classes with $r \neq m-1$, s or $h \neq \emptyset$ and $t = \emptyset$. There are $(m-1)(2^{n-1}-1)(2^q-1)$ classes with $r \neq m-1$, $n-1 \notin t$, $t \neq \emptyset$, $s = \emptyset$ and $h \neq \emptyset$. There are $2^{n-2}(2^q-1)$ classes with $r = m-1$, $n-1 \notin t$,

$0 \in t$, $s = \emptyset$ and $h \neq \emptyset$. Thus, there are at least

$$9(2m-1)2^{n+p+q-5} - 3(m-1)2^{p+q-2} - (2m-1)2^{n+q-2} + (m-1)2^q + (2m-1)2^{n-2}$$

distinct equivalence classes. ■

We now show an upper bound of this combined operation.

Theorem 4.4 *Let A, B, C and D be four DFAs of m, n, p and q states, respectively, $m, n, p, q > 0$, where A has k final states B has l final states and C has u final states , $0 < k < m$, $0 < j < n$ and $0 < u < p$. Then there exists a DFA of $(2m-k)2^{n+p+q-3} + (2m-k)2^{n+p+q-l-u-3} + (2m-k)2^{n+p+q-l-3} + (2m-k)2^{n+p+q-u-3} - (m-k)2^{p+q-1} - (m-k)2^{p+q-u-1} - (2m-k)2^{n+q-l-1} + (m-k)2^q + (2m-k)2^{n-l-1}$ states that accepts L(A)L(B)L(C)L(D).*

Proof. Let $A = (Q_A, \Sigma, \delta_A, r_0, F_A)$, $B = (Q_B, \Sigma, \delta_B, s_0, F_B)$, $C = (Q_C, \Sigma, \delta_C, t_0, F_C)$ and $D = (Q_D, \Sigma, \delta_D, h_0, F_D)$. Construct $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ such that

$$\begin{aligned}
Q_E \;=\; & Q_A \times 2^{Q_B} \times 2^{Q_C} \times 2^{Q_D} - F_A \times 2^{Q_B - \{s_0\}} \times 2^{Q_C} \times 2^{Q_D} \\
& -(Q_A - F_A) \times (2^{F_B} - \{\emptyset\}) \times 2^{Q_B - F_B} \times 2^{Q_C - \{t_0\}} \times 2^{Q_D} \\
& -F_A \times (2^{F_B} - \{\emptyset\}) \times 2^{Q_B - F_B - \{s_0\}} \times 2^{Q_C - \{t_0\}} \times 2^{Q_D} \\
& -(Q_A - F_A) \times (2^{Q_B - F_B} - \{\emptyset\}) \times (2^{F_C} - \{\emptyset\}) \times 2^{Q_C - F_C} \times 2^{Q_D - \{h_0\}} \\
& -F_A \times 2^{Q_B - F_B - \{s_0\}} \times (2^{F_C} - \{\emptyset\}) \times 2^{Q_C - F_C} \times 2^{Q_D - \{h_0\}} \\
& -(Q_A - F_A) \times (2^{F_B} - \{\emptyset\}) \times 2^{Q_B - F_B} \times (2^{F_C} - \{\emptyset\}) \times 2^{Q_C - F_C - \{t_0\}} \times 2^{Q_D - \{h_0\}} \\
& -F_A \times (2^{F_B} - \{\emptyset\}) \times 2^{Q_B - F_B - \{s_0\}} \times (2^{F_C} - \{\emptyset\}) \times 2^{Q_C - F_C - \{t_0\}} \times 2^{Q_D - \{h_0\}} \\
& -(Q_A - F_A) \times \emptyset \times (2^{Q_C} \times 2^{Q_D} - \{<\emptyset, \emptyset>\}) \\
& -(Q_A - F_A) \times (2^{Q_B - F_B} - \{\emptyset\}) \times \emptyset \times (2^{Q_D} - \{\emptyset\}) \\
& -F_A \times 2^{Q_B - F_B - \{s_0\}} \times \emptyset \times (2^{Q_D} - \{\emptyset\});
\end{aligned}$$

$$\begin{aligned}
q_0 &= <r_0, \emptyset, \emptyset, \emptyset>, \quad \text{if } \; r_0 \notin F_A, \;\; s_0 \notin F_B \;\text{ and }\; t_0 \notin F_C; \\
q_0 &= <r_0, \{s_0\}, \emptyset, \emptyset>, \quad \text{if } \; r_0 \in F_A, \;\; s_0 \notin F_B \;\text{ and }\; t_0 \notin F_C; \\
q_0 &= <r_0, \{s_0\}, \{t_0\}, \emptyset>, \quad \text{if } \; r_0 \in F_A, \;\; s_0 \in F_B \;\text{ and }\; t_0 \notin F_C; \\
q_0 &= <r_0, \{s_0\}, \{t_0\}, \{h_0\}>, \quad \text{if } \; r_0 \in F_A, \;\; s_0 \in F_B \;\text{ and }\; t_0 \in F_C; \\
F_E &= \{<r, S, T, H> \in Q_E \mid H \cap F_D \neq \emptyset\}; \\
\delta_E(<r, S, T, H>, a) &= <r', S', T', H'>, \quad \text{for } \; a \in \Sigma, \quad \text{where } \; r' = \delta_A(r, a), \\
S' &= \delta_B(S, a) \cup \{s_0\} \;\text{ if }\; r' \in F_A, \quad S' = \delta_B(S, a) \;\text{ otherwise}, \\
T' &= \delta_C(T, a) \cup \{t_0\} \;\text{ if }\; S' \cap F_B \neq \emptyset, \quad T' = \delta_C(T, a) \;\text{ otherwise}, \\
H' &= \delta_D(H, a) \cup \{h_0\} \;\text{ if }\; T' \cap F_C \neq \emptyset, \quad H' = \delta_D(H, a) \;\text{ otherwise}.
\end{aligned}$$

Intuitively, Q_E is a set of 4-tuples such that the first component is in Q_A, the second component is a subset of Q_B, the third component is a subset of Q_C and the last component is a subset of Q_D.

Q_E does not contain those 4-tuples whose first component is a final state of A and whose second component does not contain s_0, the initial state of B.

Q_E does not contain those 4-tuples whose second component contains at least one final state of B and whose third component does not contain t_0, the initial state of C, neither. Notice that the 4-tuples whose first component is a final state of A and whose second component contains at least one final state of B but does not contain s_0 and whose last component does not contain t_0 has been included in the first case.

Q_E does not contain those 4-tuples whose second component does not contain any final state of B, and whose third component contains at least one final state of C and whose last component does not contain h_0, the initial state of D.

Q_E does not contain those 4-tuples whose second component contains a final state of B, and whose third component contains t_0 and at least one final state of C and whose last component does not contain h_0, the initial state of D.

Q_E also does not contain those 4-tuples whose first component is a non-final state of A and whose second component is $\emptyset$ and whose third component or last component is nonempty.

Q_E does not contain those 4-tuples whose first component is a non-final state of A and whose second component is nonempty and contains no final state of B and whose third component is $\emptyset$ and last component is nonempty.

Q_E does not contain those 4-tuples whose first component is a final state of A and second component contains no final state of B but contains s_0 and third component is $\emptyset$ and last component is nonempty.

Clearly, $L(E) = L(A)L(B)L(C)L(D)$. Let $|Q_A| = m$, $|Q_B| = n$, $|Q_C| = p$, $|Q_D| = q$, $|F_A| = k$, $|F_B| = l$ and $|F_C| = u$. Then E has

$$(2m-k)2^{n+p+q-3} + (2m-k)2^{n+p+q-l-u-3} + (2m-k)2^{n+p+q-l-3} + (2m-k)2^{n+p+q-u-3}$$
$$-(m-k)2^{p+q-1} - (m-k)2^{p+q-u-1} - (2m-k)2^{n+q-l-1} + (m-k)2^{q} + (2m-k)2^{n-l-1}$$

states. ■

Note that when $k = l = u = 1$, this upper bound is exactly the same as the lower bound given in Theorem 4.3. Thus, the bound is indeed the state complexity of catenation of four regular languages.

5 Conclusions and future work

We have studied the state complexities of intersection and union on k regular languages and they do equal to the $k-1$ times direct combinations of their state complexities on two regular languages.

We have also studied the state complexities of catenation on three and four regular languages. Neither of them equals to the direct three or four times combination of its state complexity on two regular languages. The reason of this difference appears to be that the result of the catenation is not among the worst cases of catenation.

Note that in the first new result, we have some limitations on the set of integers $n_1, n_2, \cdots, n_k$. It remains open what the state complexities of intersection and union on k regular languages are without those limitations.

And for catenation, we used a 5-letter alphabet and a 7-letter alphabet to achieve the state complexities of catenation on three and four regular languages, respectively. It is also open whether we can use smaller alphabets to reach the lower bounds.

Thus, many further results in this direction may be obtained in the near future.

References

[1] C. Campeanu, K. Culik, K. Salomaa and S. Yu, "State complexity of basic operations on finite languages", *Proceedings of the Fouth International Workshop on Implementing Automata*, Vol. 8, 1-11, 1999, LNCS, 2214, pp. 60-70.

[2] C. Campeanu, K. Salomaa and S. Yu, Chapter 5: "State complexity of regular languages: finite versus infinite", *Finite VS Infinite - Contributions to an Eternal Dilemma*, edited by C. Calude and G. Paun, Springer 2000, pp. 53-73.

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

[4] M. Domaratzki, "State complexity and propotional removals", *Journal of Automata, Languages and Combinatorics*, 7, 2002, pp. 455-468.

[5] Y. Gao, K. Salomaa and S. Yu, "State complexity of star of catenation and reversal", *Proceedings of 2006 Descriptional Complexity of Formal Systems (DCFS 06)*, 2006, pp. 153-164.

[6] Y. Gao, K. Salomaa and S. Yu, "State complexity of two combined operations: star of catenation and star of reversal", *Fundamenta Informaticae*, vol. 83, no. 1-2 (2008) 75-89.

[7] M. Holzer, M. Kutrib, "State complexity of basic operations on nondeterministic finite automata", *CIAA*, 2002, pp. 148-157.

[8] M. Holzer, K. Salomaa and S. Yu, "On the state complexity of K-entry deterministic finite automata", *Journal of Automata, Languages and Combinatorics*, 6, 2001, pp. 453-466.

[9] M.Holzer, M. Kutrib, "Unary language operations and their nondeterministic state complexity", *Developments in Language Theory*, 2002, Springer LNCS 2450, pp. 162-172.

[10] John E. Hopcroft, Jeffrey D. Ullman, "Introduction to automata theory, languages, and computation", 1st ed., Addison-Wesley, 1979.

[11] John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, "Introduction to automata theory, languages, and computation", 2nd ed., Addison-Wesley, 2001.

[12] K. Iwama, Y. Kambayashi and K. Takaki, "Tight bounds on the number of states of DFAs that are equivalent to an n-state NFAs", *Theoretical Computer Science*, 237, 2002, pp. 485-494.

[13] G. Jiraskova, A. Okhotin, "State complexity of cyclic shift", *Proceedings of DCFS 2005*, 2005, pp. 182-193.

[14] J. Jirasek, G. Jiraskova and A. Szabari, “State complexity of concatenation and complementation of regular languages”, *International Journal of Foundations of Computer Science*, Vol. 16, 2005, pp. 511-529.

[15] G. Jiraskova, “State complexity of some operations on regular languages”, *Proceedings of the Fifth Workshop on Discriptional Complexity of Formal Systems*, 2003, pp. 114-125.

[16] G. Jiraskova, “State complexity of some operations on binary regular languages,”, *Theoretical Computer Science*, 2003, accepted for publication.

[17] C. Nicaud, “Average state complexity of operations on unary automata” In M. Kutylowski, L. Pacholski, and T. Wierzbicki, editors, *Symposium on Mathematical Foun dations of Computer Science*, LNCS, 1672, 1999, pp. 231-240.

[18] G. Pighizzini, J. Shallit, “Unary language operations, state complexity and Jacobsthal’s function”, *International Journal of Foundations of Computer Science*, Vol. 13, 1, 2002, pp. 145-159.

[19] D. Raymond, D. Wood and S. Yu edited, “Automata implementation”, First Intern. Workshop on Implementing Automata, WIA’96, LNCS 1260.

[20] A. Salomaa, “Theory of automata”, Pergamon, New York, 1969.

[21] A. Salomaa, “Formal languages”, Academic Press, New York, 1973.

[22] A. Salomaa, D. Wood and S. Yu, “On the state complexity of reversals of regular languages”, *Theoretical Computer Science*, 320, 2004, pp. 293-313.

[23] A. Salomaa, K. Salomaa and S. Yu, “State complexity of combined operations”, *Theoretical Computer Science*, accepted for publication.

[24] D. Wood, S. Yu edited, “Automata implementation”, Second Intern. Workshop on Implementing Automata, WIA’96, LNCS 1260.

[25] S. Yu, Q. Zhuang and K. Salomaa, “The state complexity of some basic operations on regular languages”, *Theoretical Computer Science*, 125, 1994, pp. 315-328.

[26] S. Yu, “Regular languages. ” In: *Handbook of Formal Languages*, Vol. 1,(G. Rozenberg and A. Salomaa, Eds.), Springer-Verlag, 1997, pp. 41-110.

[27] S. Yu, “State complexity of regular languages”, *Proceedings of the International Workshop on Descriptional Complexity of Automata, Grammars and Related Structures*, 2000.

[28] S. Yu, “State complexity: recent results and open problems”, *Fundamenta Informaticae*,Vol. 64, 1-4, 2005, pp. 471-480.