December 2010

Advanced Topics on State Complexity of Combined Operations

Yuan Gao
The University of Western Ontario

Supervisor
Dr. Sheng Yu
The University of Western Ontario

Graduate Program in Computer Science

A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

Advanced Topics on State Complexity of Combined Operations

(Thesis Format: Monograph)

by

Yuan Gao

Department of Computer Science

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario
December 2010

THE UNIVERSITY OF WESTERN ONTARIO
SCHOOL OF GRADUATE AND POSTDOCTORAL STUDIES

CERTIFICATE OF EXAMINATION

Advisor	Examining Board
Prof. Sheng Yu	Prof. Lila Kari
	Prof. Stuart Rankin
	Prof. Jeffrey Shallit
	Prof. Robert Webber

The thesis by
Yuan Gao

entitled
Advanced Topics on State Complexity of Combined Operations

is accepted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Date ____________________ ____________________

Chair of Examining Board

ABSTRACT

State complexity is a fundamental topic in formal languages and automata theory. The study of state complexity is also strongly motivated by applications of finite automata in software engineering, programming languages, natural language and speech processing and other practical areas. Since many of these applications use automata of large sizes, it is important to know the number of states of the automata.

In this thesis, we firstly discuss the state complexities of individual operations on regular languages, including union, intersection, star, catenation, reversal and so on. The state complexity of an operation on unary languages is usually different from that of the same operation on languages over a larger alphabet. Both kinds of state complexities are reviewed in the thesis.

Secondly, we study the exact state complexities of twelve *combined operations* on regular languages. The state complexities of most of these combined operations are not equal to the compositions of the state complexities of the individual operations which make up these combined operations. We also explore the reason for this difference.

Finally, we introduce the concept of estimation and approximation of state complexity. We show close estimates and approximations of the state complexities of six combined operations on regular languages which are good enough to use in practice.

Keywords: state complexity, regular languages, combined operations, deterministic finite automata, nondeterministic finite automata, estimation of state complexity, approximation of state complexity.

To my parents

ACKNOWLEDGEMENTS

To begin with and foremost, I thank Dr. Sheng Yu for his constant guidance, invaluable encouragement and thoughtful advice on both technical and non-technical matters.

I also thank the whole faulty and staff of the department, who welcomed me and showed me how hard and seriously people worked in the field of science.

I thank my colleagues and friends for their constant encouragement.

Last, but not least, I would like to thank my parents for their never-ending patience, support, and encouragement, which gave me the strength to finish this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Automata theory is one of the oldest research areas in computer science. It started in the 1930's [102], before electronic computers were invented [52]. Since then, much research has been done in the area. Although it already has a long history, new problems in automata theory arise due to its increasing application. Research on many topics in automata theory is ongoing. For example, statecharts, which are widely used as a modeling tool in software engineering, come from automata theory [71]. The use of finite automata has been shown to be successful in lexical analysis in programming languages [98]. In parallel programming, automata theory has been associated with optimization problems [75]. Automata theory also serves as the basis for pattern recognition in natural language and speech processing [74, 78]. These applications motivate the study of state complexity, a fundamental subarea in automata theory.

1.1 Why State Complexity?

One basic question in research on finite automata and regular languages is how to measure the size of a *deterministic finite automaton* (DFA). There are three ways to do this: the number of states, the number of transitions, or a combination of the two [105]. For a complete DFA, whose transition function is defined for every state and every possible input symbol, the number of transitions is linear in the number of states if the alphabet is considered as a constant [30]. Thus, the number of states becomes the key point when we try to measure the size of a complete DFA. State complexity is the study of the number of states of finite automata.

Generally speaking, the study of complexity issues mainly focuses on the following two kinds of issues: (1) time and space complexity issues, (2) descriptional complexity issues [105]. State complexity is a type of descriptional complexity. It is based on the finite automaton model. The state complexity of an operation on regular languages gives a lower bound for the space complexity and the time complexity of the operation. For many operations, the bounds given by the state complexities are tight.

Compared to other representations of regular languages such as *nondeterministic finite automata* (NFAs) [25, 49] and regular expressions, the DFA model has the following advantages [105]. (1) It takes almost linear time to check two DFAs to determine if they are equal [1], while for NFA and regular expressions it is PSPACE-complete. (2) For a regular language, the minimal DFA that accepts the language is unique up to isomorphism, while other models are not unique in general. (3) There is an $O(n \log n)$-time minimization algorithm for DFAs; however, the same problem for other models is not known to be solvable in polynomial time. Thus, the size of a minimal DFA is a natural and objective measurement for the language it accepts [105].

1.2 Why State Complexity of Combined operations?

During the last 20 years, motivated by new applications of regular languages that require automata of very large sizes, state complexity has received increased attention and many papers have been published on this topic. Examples include [3, 4, 5, 6, 7, 8, 9, 14, 15, 16, 17, 18, 20, 21, 22, 23, 24, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 46, 47, 48, 50, 54, 55, 56, 57, 58, 59, 60, 61, 63, 66, 67, 68, 77, 79, 80, 81, 82, 83, 84, 85, 87, 93, 94, 95, 96, 99, 100, 105, 106, 110, 111].

Most of these papers focused on individual operations, for example, union, intersection, star, catenation, reversal, shuffle, orthogonal catenation, proportional removal, cyclic shift and so on. However, in practice, the operation to be performed is often a combination of several individual operations in a certain order rather than only one individual operation. The state complexity of combined operations is certainly an important research direction in state complexity research. The state complexities of a number of combined operations have been studied in the past five years. It has been shown that the state complexity of a combination of several operations is usually not equal to the composition of the state complexities of individual participating operations [28, 31, 45, 62, 70, 92].

There seems to be no common method to compute the state complexities of combined operations because each combined operation has its own special features. Although the composition of individual state complexities of component operations of a combined operation would give an upper bound to the state complexity of the combined operation, the upper bound is usually too large to be useful. For example, for two regular languages L_1 and L_2 accepted by m-state and n-state deterministic finite automata, respectively, the state complexity of $(L_1 \cup L_2)^*$ is actually $2^{m+n-1} - 2^{m-1} - 2^{n-1} + 1$, while the composition of the individual state complexities is $2^{mn-1} + 2^{mn-2}$. So it appears that the state complexity of each combined operation has to be studied specifically.

1.3 Why Estimation and Approximation of State Complexity of Combined Operations?

There are only a limited number of individual operations on regular languages. However, the number of combined operations on regular languages is unlimited. Thus, it is important to obtain general results that cover not only single combined operations, but also infinite classes of combined operations. A good estimate of the state complexity of a combined operation can be used in many applications.

The method of estimation through nondeterministic state complexity was proposed in [97, 108]. For most combined operations that include the star operation or reversal, it gives good estimates. For example, the estimation of the state complexity of $(L_1 \cup L_2)^*$ through nondeterministic state complexity is 2^{m+n+2}. Note that $m + n + 2$ is the direct mathematical composition of the two individual nondeterministic state complexities and no optimization is made. This estimation is close to the exact state complexity of this combined operation: $2^{m+n-1} - 2^{m-1} - 2^{n-1} + 1$.

However, this method has its limitations. Considering the union of $k > 1$ regular languages accepted by DFAs of $n_1, \ldots, n_k$ states, respectively, the estimate of its state complexity through nondeterministic state complexity would result in $2^{n_1+\cdots+n_k+k-1}$. It can be easily shown that the state complexity of this operation is no more than $n_1 \cdots n_k$.

Although an estimate of the state complexity of a combined operation is simpler and more convenient to use, it does not show how close it is to the exact state complexity. The concept of approximation of state complexity solves this problem by defining the ratio bound which provides the error range of the estimate [32].

Approximation of state complexity can play useful roles in two different cases. In the first case, the exact state complexities have not been obtained. They may be very difficult to obtain. However, approximations with low ratio bounds can be obtained rather easily and they can be used for practical purposes in general. In the second case, the exact state complexities have been proved. The approximations of those results with low ratio bounds can simplify the formulae of the complexities and make them more intuitive and easier to apply. Thus, approximation of state complexity is clearly a useful and important concept in the study of state complexity of combined operations.

1.4 New Contributions of the Thesis

This thesis focuses on two topics on the state complexities of combined operations on regular languages:

(1) exact state complexity of combined operations, and

(2) estimation and approximation of state complexity of combined operations.

In this thesis, we discuss exact state complexity of combined operations on regular languages. It is one of the major topics of the study of state complexity in the past five years. We choose 12 combined operations which are commonly used in practice and investigate their exact state complexities. These combined operations include: combinations of union, intersection and complementation, multiple catenations, combinations of star and catenation, reversal and catenation, Boolean operations and catenation, Boolean operations and star, Boolean operations and reversal. For all these combined operations, we obtain tight bounds on their state complexities.

We also study estimation and approximation of state complexity of combined operations on regular languages. We revisit the method of estimation of state complexity through nondeterministic state complexity and clarify the boundaries of its usage. We introduce the concept of approximation of state complexity and obtain approximations of the state complexities of 6 combined operations on regular languages. All of them are close to the corresponding exact state complexities.

An important aspect of the research of this thesis is that it combines abstract theoretical work with the use of software systems, such as Grail+ [112], to help us find worst-case examples experimentally. Hundreds of DFAs of large sizes are used in the experiments. If we do all these experiments on paper, the researcher can often get no result. State complexity is a theoretical topic. However, experiments play an important role in the study of state complexity. Although the final results always have to be proved theoretically, experiments verify or reject our initial proposal and greatly speed up the research process.

1.5 Outline of the Thesis

The thesis is structured as follows:

Chapter 2 presents basic notation and definitions used in this thesis.

Chapter 3 gives a survey of research results on the state complexities of individual operations on regular languages.

Chapter 4 presents the current results of the state complexities of combined operations, including star of union, star of intersection, star of catenation, star of reversal, reversal of union, reversal of intersection, reversal of catenation and power.

Chapter 5 presents the exact state complexities of 12 combined operations, including catenation combined with star and reversal, catenation combined with union and intersection, combined Boolean operations and multiple catenations.

Chapter 6 introduces the research results on estimation and approximation of state complexity of combined operations on regular languages.

Chapter 7 concludes the thesis with discussions of state complexity of combined operations and future work.

Chapter 2

Basic Definitions and Notation

In this chapter, we review some basic knowledge about formal languages and automata theory [52, 53, 90, 91, 101, 104] that is related to this thesis. This knowledge is the foundation of any study not only in state complexity but also the whole of computer science. We also specify the notation which is used in the thesis.

2.1 Languages

An alphabet is a finite, nonempty set of symbols, denoted by Σ. The symbols in the alphabet are also called letters. The notation Σ^* means the set containing all the finite strings whose symbols are chosen from an alphabet Σ.

Strings, which are finite sequences of letters, are also called words. A special word is the empty word, denoted by ε. It is over any alphabet.

For a word x over an alphabet Σ, its length is the number of occurrences of letters in x. It is denoted by $|x|$. The a-length of the word x is the number of times that the letter a appears in x. It is denoted by $|x|_a$.

A language over Σ is a set of words which are chosen from Σ^*. The languages $\{\varepsilon\}$ and $\emptyset$ are over any alphabet. We use the notation $|L|$ to denote the cardinality of a language L, i.e., the number of words in L. (There should be no confusion with the same notation used for the length of a word.)

2.2 Operations

For a word x over an alphabet Σ, the reversal of x is denoted by x^R. It is x itself if $x = \varepsilon$; it is $y^R a$ if $x = ay$, where a is a letter in Σ and y is a word over

Σ. By the definition, if $x = a_1 \cdots a_n$, where $n \geq 0$ and $a_1, \cdots, a_n$ are letters in Σ, then $x^R = a_n \cdots a_1$.

For a language L over an alphabet Σ, the reversal of L is denoted by L^R, and $L^R = \{x^R \mid x \in L\}$.

For words x and y over an alphabet Σ, the catenation of x and y is denoted by xy. It is the word obtained by attaching y to the end of x. Catenation is associative. The length of the new word xy is the sum of the length of x and the length of y.

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, and $L_1L_2 = \{xy \mid x \in L_1, \quad y \in L_2\}$.

For a language L over an alphabet Σ, the star of L is denoted by L^*. The operation L^* is also called *Kleene closure*. We define $L^0 = \{\varepsilon\}$ and $L^i = L^{i-1}L$, where $i \geq 1$. We define $L^* = \cup_{i=0}^{\infty} L^i$. Similarly, we define L^+ as $\cup_{i=1}^{\infty} L^i$. The operation L^+ is called *positive closure*.

Given a set S, the power set of S is the set of all subsets of S, denoted by $\mathcal{P}(S)$.

Let R and L be two languages over the alphabet Σ. Then the left quotient of R by L, denoted by $L \backslash R$, is the language

$$\{y \mid xy \in R \text{ and } x \in L\},$$

and the right quotient of R by L, denoted by R/L, is the language

$$\{x \mid xy \in R \text{ and } y \in L\}.$$

2.3 Grammars

Definition 2.1 Context-free Grammars

A context-free grammar (CFG) G is specified by a quadruple (N, Σ, P, S) where

N is the set of nonterminals (variables);

Σ is the set of terminals, $\Sigma \cap N = \emptyset$;

$P \subseteq N \times (N \cup \Sigma)^*$ is the set of productions;

$S \in N$ is sentence symbol;

and N, Σ, and P are all finite [109].

A CFG generates a word by rewriting (or derivation) [109]. Let $G = (N, \Sigma, P, S)$ be a CFG and $\beta, \beta' \in (N \cup \Sigma)^*$. If $\beta = \beta_1 A \beta_2$ for $A \in N$, $\beta_1, \beta_2 \in (N \cup \Sigma)^*$, $A \to \alpha \in P$ and $\beta' = \beta_1 \alpha \beta_2$, then we say that β can be rewritten as β', denoted by $\beta \Longrightarrow \beta'$ [109].

$\beta \Longrightarrow^i \beta', i > 0,$ if β' can be obtained from β in i rewriting steps [109].

$\beta \Longrightarrow^+ \beta',$ if β' can be obtained from β in at least one rewriting steps [109].

$\beta \Longrightarrow^* \beta',$ if $\beta = \beta'$ or $\beta \Longrightarrow^+ \beta'$ [109].

The language that consists of the words generated by the CFG G is denoted by $L(G)$ and

$$L(G) = \{w \in \Sigma^* \mid S \Longrightarrow^* w\} \text{ [109]}.$$

Example 2.1 A CFG for $\{a^i b^i \mid i \geq 1\}$ is as follows: $S \to aSb \mid ab$, $N = \{S\}$ and $\Sigma = \{a, b\}$. $\square$

Definition 2.2 Right Linear Grammars

A CFG $G = (N, \Sigma, P, S)$ is right linear if every production in P is of one of the forms:

$$A \to x, \;\; A \to xB, \;\; A, B \in N, \;\; x \in \Sigma^* \text{ [109]}.$$

Definition 2.3 Left Linear Grammars

A CFG $G = (N, \Sigma, P, S)$ is left linear if every production in P is of one of the forms:

$$A \to x, \;\; A \to Bx, \;\; A, B \in N, \;\; x \in \Sigma^* \text{ [109]}.$$

Definition 2.4 Regular Grammars

A CFG G is said to be regular if it is right linear or left linear [109].

Example 2.2 A regular grammar for $\{w \in \{a, b\}^* \mid |w| \geq 1\}$ is as follows: $S \to aS \mid bS \mid a \mid b$, $N = \{S\}$ and $\Sigma = \{a, b\}$. $\square$

2.4 Regular Expressions

Definition 2.5 Regular Expressions

A regular expression over the base alphabet Σ *is a well-formed string over the larger alphabet* $\Sigma \cup A$*, where* $A = \{\varepsilon, \emptyset, (,), +, *\}$*; we assume* $\Sigma \cap A = \emptyset$ *[101]. Valid regular expressions can be defined with a CFG grammar as follows [101]:*

$$
\begin{aligned}
S &\rightarrow E_+ \mid E_\bullet \mid G \\
E_+ &\rightarrow E_+ + F \mid F + F \\
F &\rightarrow E_\bullet \mid G \\
E_\bullet &\rightarrow E_\bullet G \mid GG \\
G &\rightarrow E_* \mid C \mid P \\
C &\rightarrow \emptyset \mid \varepsilon \mid a \quad (a \in \Sigma) \\
E_* &\rightarrow G * \\
P &\rightarrow (S)
\end{aligned}
$$

The meaning of the variables is as follows [101]:

- S *generates all regular expressions.*
- E_+ *generates all unparenthesized expressions where the last operator was* $+$.
- $E_\bullet$ *generates all unparenthesized expressions where the last operator was* $\cdot$ *(implicit concatenation).*
- E_* *generates all unparenthesized expressions where the last operator was* $*$ *(Kleene closure).*
- C *generates all unparenthesized expressions where there was no last operator (i.e., the constants).*
- P *generates all parenthesized expressions.*

Here, by parenthesized we mean there is at least one pair of enclosing parentheses [101]. If the word u is a regular expression, then $L(u)$ represents the language that u is shorthand for [101].

Example 2.3 Consider the regular expression $u = (0 + 1)^*1$. Then $L(u)$ represents all the words over $\{0, 1\}$ that end with 1. $\square$

2.5 Regular Languages

There are four levels of languages according to the Chomsky hierarchy of formal languages, which are the regular languages, the context-free languages, the context-sensitive languages and the recursively enumerable languages. In this thesis, regular languages are discussed. A language L is regular if and only if there is a regular expression E such that $L = L(E)$.

Regular Languages are generated by regular grammars. A language L is regular if and only if there is a regular grammar G such that $L = L(G)$ [109].

Finite languages make up a specific subset of the class of regular languages. Each finite language contains only a finite number of words. They are regular since a finite language can be described by a regular expression that is the union of every word in the language.

2.6 Deterministic Finite Automata

Definition 2.6 Deterministic Finite Automata

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 that takes a state and an input symbol as arguments and returns a state. If* p *is the current state, and* a *is the input symbol, then* $\delta(p, a) = q$ *means the DFA transfers from* p *to* q *by reading the letter* a*;*

q_0 *is an initial state where* $q_0 \in Q$*;*

F *is a set of final states and* $F \subseteq Q$*.*

An extended transition function $\hat{\delta}$ describes what happens when we start in any state and follow any sequence of inputs [52]. We define $\hat{\delta}$ by induction on the length of the input string, as follows [52]:

Basis: $\hat{\delta}(q, \varepsilon) = q$. That is, if we are in state q and read no inputs, then we are still in state q.

Induction: Suppose w is a string of the form xa; that is, a is the last symbol of w, and x is the string consisting of all but the last symbol. Then

$$\hat{\delta}(q, w) = \delta(\hat{\delta}(q, x), a).$$

The language accepted by a DFA is the set of strings that take the initial state to one of the final states. If a language L is accepted by some DFA, then L is a regular language. Two DFAs are equivalent if they accept the same regular language.

Example 2.4 Let the DFA $A = (Q, \Sigma, \delta, q_0, F)$ be given by $Q = \{0, 1, 2, 3, 4, 5\}$, $\Sigma = \{a\}$, $q_0 = 0$, $F = \{5\}$, and

$$\delta(m, a) = n, \quad m \in Q, \quad n = (m + 1) \bmod 6.$$

The regular language accepted by A can be denoted by

$$L(A) = \{a^i \mid i \equiv 5 \pmod 6\}.$$

The transition diagram of A is shown in Figure 2.1. □

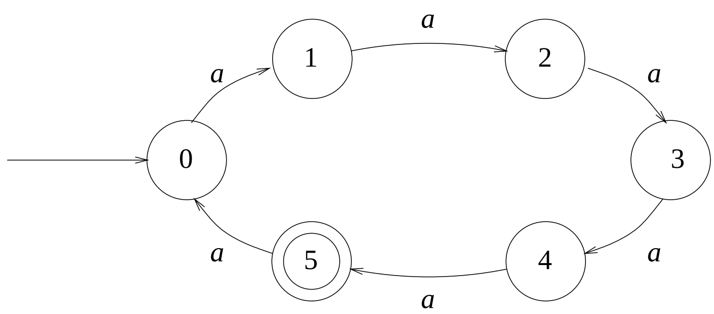

Figure 2.1: The transition diagram of DFA A in Example 2.4

A complete DFA is one that has transitions defined for each state in Q and each input symbol in Σ. A sink state is a state from which there exists no sequence of transitions to a final state.

For every DFA A, there exists an automaton B such that $L(A) = L(B)$ and (1) every state of B is reachable from the initial state and (2) from every state, except at most one sink state, a final state can be reached. The DFA B is called a reduced DFA.

Note that we assume that all the DFAs used are complete in this thesis.

2.7 Minimization of DFAs

There are many DFAs that accept the same regular language. An important way to test the equivalence of DFAs is to minimize them. That is, for each DFA we can find an equivalent DFA that has as few states as any DFA that accepts the same language [52]. Since minimal DFAs are used in the study of state complexity, we will go through DFA minimization algorithms. Firstly, the Myhill-Nerode theorem implies that there is an essentially unique minimum-state DFA for each regular language [53].

Theorem 2.1 *(The Myhill-Nerode theorem). The following three statements are equivalent:*

1) The set $L \subseteq \Sigma^*$ is accepted by some finite automaton.

2) L is the union of some of the equivalence classes of a right invariant equivalence relation of finite index.

3) Let the equivalence relation R_L be defined by xR_Ly if and only if for all z in Σ^*, xz is in L exactly when yz is in L. Then R_L is of finite index.

We say that states p and q in a DFA are equivalent if:

> For all input strings w, $\hat{\delta}(p, w)$ is a final state if and only if $\hat{\delta}(q, w)$ is a final state [52].

If two states are not equivalent, then we say they are distinguishable [52]. State p is distinguishable from state q if there exists at least one string w such that one of $\hat{\delta}(p, w)$ and $\hat{\delta}(q, w)$ is final, and the other is not final [52].

There is a simple method to minimize DFAs. Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA. To minimize A, we firstly eliminate all the states which cannot be reached from the initial state. Secondly, we use the table-filling algorithm to partition the remaining states into blocks such that:

1. All the states in a block are equivalent.

2. No two states chosen from two different blocks are equivalent [52].

The table-filling algorithm shown in [53] is as follows:

begin
for $p \in F$ and $q \in Q - F$ **do** mark (p, q);
for each pair of distinct states (p, q) in $F \times F$ or $(Q - F) \times (Q - F)$
  do
  if for some input symbols a, $(\delta(p, a), \delta(q, a))$ is marked **then**
    begin
    mark (p, q);
    recursively mark all unmarked pairs on the list for (p, q)
    and on the lists of other pairs that are marked at this step;
    end
  else $/*$ no pair $(\delta(p, a), \delta(q, a))$ is marked $*/$
    for all input symbols a **do**
      put (p, q) on the list for $(\delta(p, a), \delta(q, a))$ unless
      $\delta(p, a) = \delta(q, a)$;
end

After partitioning the set of states Q into blocks of mutually equivalent states by the algorithm described above, we can construct the minimum-state equivalent DFA B by using the blocks as its states. The initial state of B is the block containing the initial state of A. The set of final states of B is the set of blocks containing final states of A. Note that if one state of a block is accepting, then all the states of that block must be accepting. The reason is that any final state is distinguishable from any non-final state. Thus, you cannot have both final and non-final states in one block of equivalent states.

A detailed example of DFA minimization can be found in [52]. The time complexity of the above minimization algorithm is $O(n^2)$. Most textbooks on automata theory give the above algorithm to minimize the number of states in a DFA, because it is simple and easy to grasp. However, J. Hopcroft published an $O(n \log n)$-time minimization algorithm [51]. It is more complex but faster because it uses a different approach to partition the states. If $A = \{Q, \Sigma, \delta, q_0, F\}$ is a DFA, the $O(n \log n)$ algorithm is as follows [51]:

begin
for $q \in Q$ and $a \in \Sigma$ **do**
  construct $\delta^{-1}(q,a) = \{t \mid \delta(t,a) = q\}$;
construct $B(1) = F$ and $B(2) = Q - F$;
for $a \in \Sigma$ and $1 \leq i \leq 2$ **do**
  construct $a(i) = \{s \mid s \in B(i)$ and $\delta^{-1}(s,a) \neq \emptyset\}$;
$k = 3$;
for $a \in \Sigma$ **do**

  construct $L(a) = \begin{cases} \{1\} & \text{if } |a(1)| \leq |a(2)|, \\ \{2\} & \text{otherwise;} \end{cases}$

while there exists $a \in \Sigma$ such that $L(a) \neq \emptyset$ **do**
 for $a \in \Sigma$ and $i \in L(a)$ **do**
  begin
  delete i from $L(a)$;
  for $j < k$ **do**
    if there exists $t \in B(j)$ with $\delta(t,a) \in a(i)$ **then**
      begin
      partition $B(j)$ into $B'(j) = \{t \mid \delta(t,a) \in a(i)\}$ and
      $B''(j) = B(j) - B'(j)$;
      $B(j) = B'(j)$;
      $B(k) = B''(j)$;
      for $a \in \Sigma$ **do**
        begin
        construct corresponding $a(j)$ and $a(k)$;

        $L(a) = \begin{cases} L(a) \cup \{j\} & \text{if } j \notin L(a) \text{ and } 0 < |a(j)| \leq |a(k)|, \\ L(a) \cup \{k\} & \text{otherwise;} \end{cases}$

        end
      $k = k + 1$;
      end
  end
end

2.8 Nondeterministic Finite Automata

Definition 2.7 Nondeterministic Finite Automata

A nondeterministic finite automaton (NFA) 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 alphabet;*

δ *is a transition function that takes a state in* Q *and an input symbol in* Σ *as arguments and returns a subset of* Q*. If* p *is the current state, and* a *is the input symbol, then* $\delta(p, a) = \{q, r, t\}$ *means the NFA transfers from* p *to* q, r *or* t *by reading the letter* a*;*

q_0 *is an initial state, where* $q_0 \in Q$*;*

F *is a set of final states, where* $F \subseteq Q$*.*

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 $(Q, \Sigma, \delta, S, F)$, where S is the set of the initial states.

An ε-NFA is a further extension of NFA. Its transition function δ: $Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ allows the ε-transitions between states.

Comparing the definitions of a DFA and an NFA, we find that the definitions of their transition functions are different. The transition function of a DFA maps a pair of a state and an input symbol into one state. The transition function of an NFA maps a pair of a state and an input symbol into a set of states.

Two finite automata are equivalent if they accept the same language. Given an n-state NFA A, we can always find a 2^n-state DFA A' which is equivalent to A by performing the subset construction [52, 89]. A language L is accepted by some DFA if and only if L is accepted by some NFA. Thus, DFAs and NFAs accept exactly the same class of languages as regular expressions describe, which we have termed the "regular languages" [52]. Sometimes, in the study of state complexity, an upper bound on the number of states can be estimated using an NFA and converting it into a DFA at the end.

2.9 State Complexity

Definition 2.8 State Complexity

1. *The state complexity of a regular language L is the number of states of the minimal DFA that accepts L [105].*
2. *The state complexity of a class of regular languages is the largest among the state complexities of all the languages in the class [105].*

There are two kinds of state complexity with respect to different complexity types: average-case state complexity and worst-case state complexity. Average state complexity was first studied by C. Nicaud [76]. In this thesis, we study only worst-case state complexity.

With respect to different automaton models, there is deterministic state complexity and nondeterministic state complexity [26, 35, 36, 42, 43, 44, 47, 48, 49]. As we discussed in Section 1.1, the DFA model is more suitable to represent regular languages in general than the NFA model when we study state complexity. So the state complexity we study here is deterministic state complexity.

With respect to different problem types, we have operational state complexity and representational state complexity. Representational state complexity studies the state complexity of transformations between models. For example, given an n-state NFA, the DFA which is equivalent to it has at most 2^n states. Operational state complexity studies the state complexity of operations on regular languages.

When we speak about the state complexity of an operation on regular languages, we mean the state complexity of the resulting languages from the operation [105]. For example, when we say that the state complexity of the union of 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 other words, there exist two regular languages which are respectively accepted by an m-state DFA and an n-state DFA, such that their union is accepted by an mn-state DFA in the worst case.

In this thesis, when we study the state complexity of an operation, we may assume the operand languages of the operation are over the same alphabet without loss of generality.

Thus, the state complexity we study in this thesis is worst-case, deterministic, operational state complexity.

In the next chapter, we will review the state complexities of many individual operations on regular languages.

Chapter 3

State Complexity of Individual Operations

Many papers on the state complexities of individual operations have been published since the early 1990's, for example, [23, 56, 58, 83, 93, 105, 110, 111]. The state complexities of most individual operations such as union, intersection, catenation, star, etc., have been obtained. In this chapter, we first review the state complexities of these operations on regular languages over a general alphabet. For catenation, star and reversal, both the upper bounds and the worst-case examples of their state complexities are presented. Next, we review the mathematical model for DFAs that accept unary regular languages and the state complexities of individual operations on unary regular languages. Lastly, the state complexities of individual operations on finite languages, both over a general alphabet and over a unary alphabet, are presented.

3.1 State Complexity of Individual Operations on Regular Languages

3.1.1 Regular Languages over a General Alphabet

The following theorems about the state complexities of individual operations on regular languages over a general alphabet have been proved in [72, 73, 110, 111].

State Complexity of Catenation

Theorem 3.1 *For integers $m \geq 1$ and $n \geq 2$, there exists a DFA A of m states and a DFA B of n states, such that any DFA that accepts $L(A)L(B)$*

needs at least $m2^n - 2^{n-1}$ *states.*

Theorem 3.1 is given in [111]. It can be proved in two cases. The first one is when $m = 1$ and $n \geq 2$. Define DFA $A = \{Q, \Sigma, \delta_A, q_0, F_A\}$ where $Q = \{q_0\}$, $\Sigma = \{a, b\}$, $F_A = \{q_0\}$, $\delta_A(q_0, a) = q_0$ and $\delta_A(q_0, b) = q_0$. Define DFA $B = \{P, \Sigma, \delta_B, p_0, F_B\}$ where $P = \{p_0, p_1, \cdots, p_{n-1}\}$, $\Sigma = \{a, b\}$, $F_B = \{p_{n-1}\}$, and

$$\begin{aligned}
\delta_B(p_0, a) &= p_0, \\
\delta_B(p_0, b) &= p_1, \\
\delta_B(p_{n-1}, a) &= p_1, \\
\delta_B(p_i, a) &= p_{i+1}, \quad 1 \leq i \leq n-2, \\
\delta_B(p_i, b) &= p_i, \quad 1 \leq i \leq n-1.
\end{aligned}$$

Figure 3.1 shows the transition diagram of B. It has been proved that any

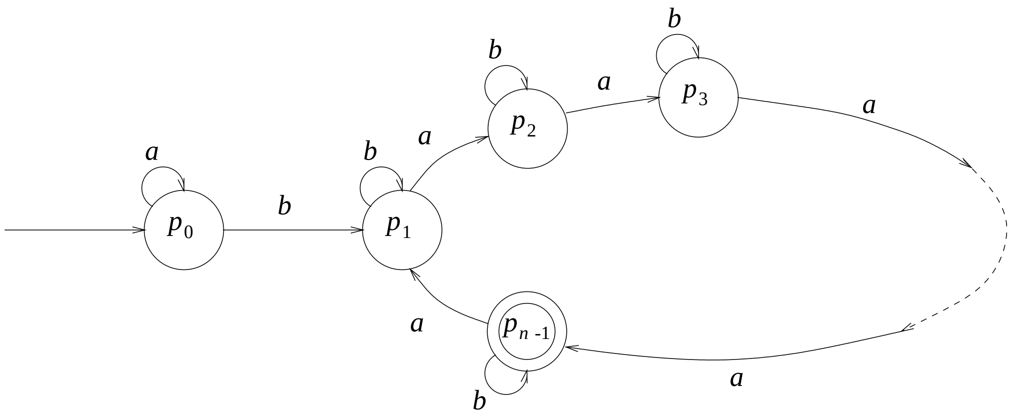

Figure 3.1: Witness DFA B for the first case of Theorem 3.1

DFA that accepts $L(A)L(B)$ needs at least 2^{n-1} states [111].

The second case is when $m \geq 2$ and $n \geq 2$. Define DFA $A = \{Q, \Sigma, \delta_A, q_0, F_A\}$ where $Q = \{q_0, q_1, \cdots, q_{m-1}\}$, $\Sigma = \{a, b, c\}$, $F_A = \{q_{m-1}\}$ and each i, $0 \leq i \leq m-1$, and

$$\delta_A(q_i, X) = \begin{cases} q_j, & j = (i+1) \mod m, \text{ if } X = a; \\ q_0, & \text{if } X = b; \\ q_i, & \text{if } X = c. \end{cases}$$

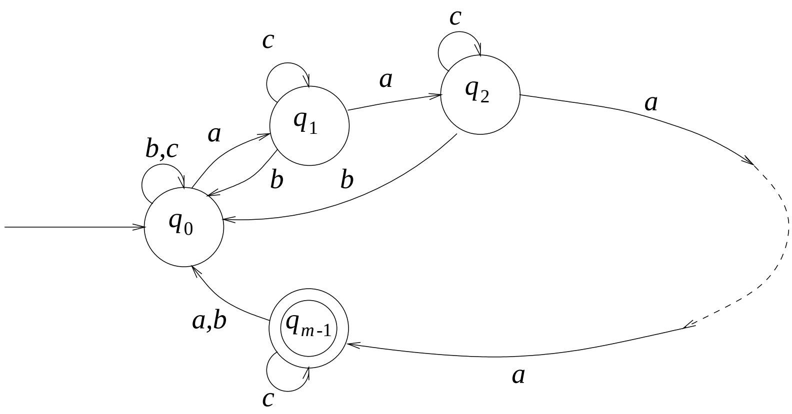

Figure 3.2: Witness DFA A for the second case of Theorem 3.1

Figure 3.2 shows the transition diagram of A. Define DFA $B = \{P, \Sigma, \delta_B, p_0, F_B\}$ where $P = \{p_0, p_1, \cdots, p_{n-1}\}$, $\Sigma = \{a, b, c\}$, $F_B = \{p_{n-1}\}$ and for each i, $0 \leq i \leq n-1$, and

$$\delta_B(p_i, X) = \begin{cases} p_j, & j = (i+1) \mod n, \text{ if } X = b; \\ p_i, & \text{if } X = a; \\ p_1, & \text{if } X = c. \end{cases}$$

Figure 3.3 shows the transition diagram of B.

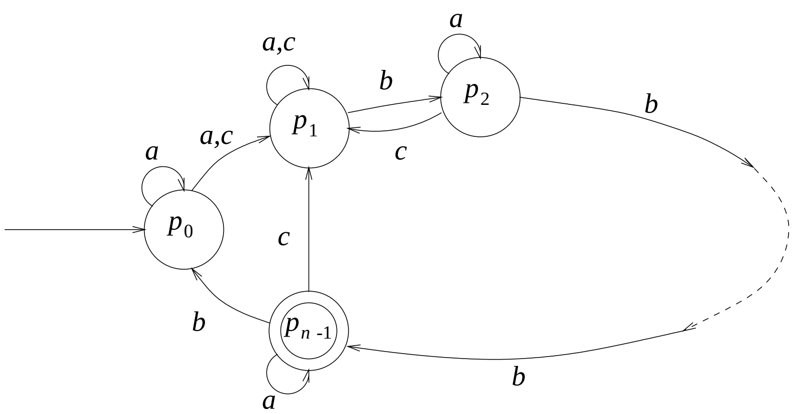

Figure 3.3: Witness DFA B for the second case of Theorem 3.1

It has been proved that any DFA that accepts $L(A)L(B)$ needs at least $m2^n - 2^{n-1}$ states [111]. Theorem 3.1 gives the lower bound on the number of states of the DFA that accepts the catenation of two regular languages.

Theorem 3.2 *Let A and B be two DFAs defined on the same alphabet, where A has m states and B has n states, and A has k final states, $0 < k < m$. Then there exists a $(m2^n - k2^{n-1})$-state DFA that accepts $L(A)L(B)$.*

Theorem 3.2 is shown in [111]. It gives an upper bound on the number of states of the DFA that accepts the catenation of two DFA languages. This upper bound coincides with the lower bound in Theorem 3.1. So the bounds are tight and we get the state complexity of catenation of regular languages shown in following theorem [111].

Theorem 3.3 *The number of states that is sufficient and necessary in the worst case for a DFA to accept the catenation of an m-state DFA language and a one-state DFA language is m.*

State Complexity of Star

Theorem 3.4 *For any n-state DFA $A = (Q, \Sigma, \delta, q_0, F)$ such that $|F\{-\ q_0\}| = k \geq 1$ and $n > 1$, there exists a DFA A' of at most $2^{n-1} + 2^{n-k-1}$ states that accepts $(L(A))^*$.*

Theorem 3.4 is given in [111]. According to this theorem, if $k \geq 1$, A' has at most $2^{n-1} + 2^{n-1-1} = 2^{n-1} + 2^{n-2}$ states. If $k = 0$, then A' needs only n states. So the following corollary can be obtained from this theorem [111].

Corollary 3.1 *For any n-state DFA A, $n > 1$, there exists a DFA A' of at most $2^{n-1} + 2^{n-2}$ states that accepts $L(A') = (L(A))^*$.*

Theorem 3.5 *For any integer $n \geq 2$, there exists a DFA A of n states such that any DFA that accepts $(L(A))^*$ needs at least $2^{n-1} + 2^{n-2}$ states.*

Theorem 3.5 is given in [111]. For $n = 2$, it is clear that $L = \{w \in \{a, b\}^* \mid |w|_a \text{ is odd}\}$ is accepted by a two-state DFA, and $L^* = \{\varepsilon\} \cup \{w \in \{a, b\}^* \mid |w|_a \geq 1\}$ cannot be accepted by a DFA with less than three states [111].

For $n > 2$, DFA $A = \{Q, \Sigma, \delta, 0, F\}$ where $Q = \{0, 1, \cdots, n-1\}$, $\Sigma = \{a, b\}$, $F = \{n - 1\}$, and

$$\begin{aligned}
\delta(i, a) &= (i + 1) \mod n, \quad 0 \leq i < n, \\
\delta(i, b) &= (i + 1) \mod n, \quad 1 \leq i < n, \\
\delta(0, b) &= 0.
\end{aligned}$$

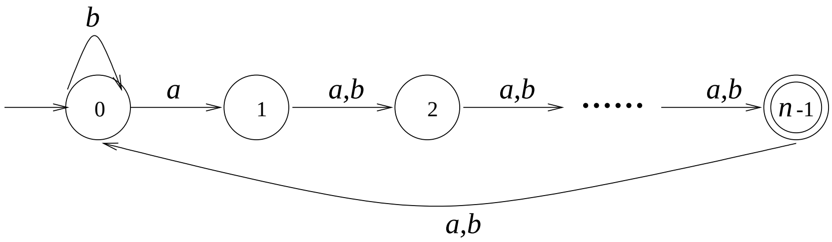

Figure 3.4: Witness DFA for Theorems 3.5 and 6.3

Figure 3.4 shows the transition diagram of A. It has been proved that any DFA that accepts $(L(A))^*$ needs at least $2^{n-1} + 2^{n-2}$ states.

State Complexity of Left Quotient

Theorem 3.6 *For any integer $n > 0$, $2^n - 1$ states are both sufficient and necessary in the worst case for a DFA to accept the left quotient of an n-state DFA language R by an arbitrary language L $(L \setminus R)$.*

Theorem 3.6 shows the state complexity of left quotient of regular languages. It is given in [111].

State Complexity of Right Quotient

Theorem 3.7 *For any integer $n > 0$, n states are both sufficient and necessary in the worst case for a DFA to accept the right quotient of an n-state DFA language R by an arbitrary language L (R/L).*

Theorem 3.7 shows the state complexity of right quotient of regular languages. It is proved in [111].

State Complexity of Reversal

Theorem 3.8 *For any integer $n > 1$, 2^n states are both sufficient and necessary in the worst case for a DFA to accept the reversal of an n-state DFA language.*

Theorem 3.8 is given in [111]. For any $n > 1$, E. Leiss has designed an n-state DFA such that the reversal of the language accepted by this DFA is accepted by a minimal DFA of 2^n states [69]. A modified n-state DFA A was designed by S. Yu, Q. Zhuang and K. Salomaa [111]. $L(A)^R$ is also accepted by a minimal DFA of 2^n states. The following example shows this modified DFA.

Example 3.1 Define DFA $A = (Q, \Sigma, \delta, 0, F)$ where $Q = \{0, 1, \cdots, n-1\}$, $\Sigma = \{a, b, c\}$, $F = \{0\}$, and

$$\begin{aligned}
&\delta(0, a) = n - 1,\ \ \delta(0, b) = 1,\\
&\delta(0, c) = 1,\ \ \delta(1, c) = 0,\\
&\delta(k, a) = k - 1,\ \ 1 \le k \le n - 1,\\
&\delta(k, b) = k,\ \ 1 \le k \le n - 1,\\
&\delta(k, c) = k,\ \ 2 \le k \le n - 1.
\end{aligned}$$

Figure 3.5 shows the transition diagram of A.

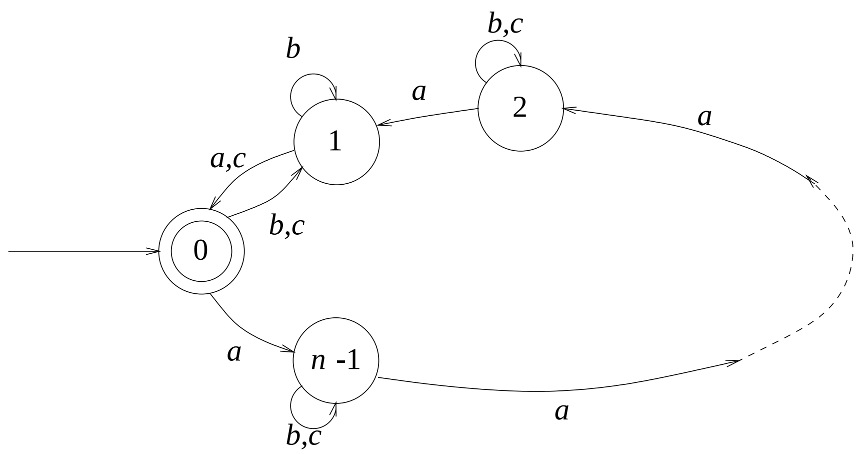

Figure 3.5: Witness DFA for Theorems 3.8, Lemma 4.3 and Theorem 5.15

□

State Complexity of Intersection and Union

Theorem 3.9 *For integers $m, n \ge 2$, $m \cdot n$ states are both sufficient and necessary in the worst case for a DFA to accept the intersection (union) of an m-state DFA language and an n-state DFA language.*

Theorem 3.9 gives the state complexities of intersection and union of regular languages. It is shown in [111].

3.1.2 Unary Regular Languages

The following results on the state complexities of several operations on regular languages with a one-letter alphabet have been proved in [83, 111].

Basic Lemmas and Models

We denote the greatest common divisor of two integers m and n by $\gcd(m,n)$ and denote the least common multiple of m and n by $\mathrm{lcm}(m,n)$. Lemma 3.1, Fact 3.1 and Lemma 3.2 in the following are given in [111].

Lemma 3.1 *Let $m, n > 0$ be two arbitrary integers such that $\gcd(m,n) = 1$ (m and n are relatively prime).*

1. *The largest integer that cannot be represented as $cm+dn$ for any integers $c, d > 0$ is mn.*
2. *The largest integer that cannot be represented as $cm+dn$ for any integers $c > 0$ and $d \geq 0$ is $(m-1)n$.*
3. *The largest integer that cannot be represented as $cm+dn$ for any integers $c, d \geq 0$ is $mn - (m+n)$.*

Fact 3.1 *Let $R \subseteq \Sigma^*$ be a regular language. If there exists an integer n such that*

$$\max\{|w| \mid w \in \Sigma^*, \ \ w \notin R\} = n$$

then any DFA that accepts R needs at least $n+2$ states. In particular, if Σ is a singleton, the minimal DFA that accepts R uses exactly $n+2$ states.

Lemma 3.2 *Suppose a, b are positive integers. Then each number of the form $ax+by$, with $x, y \geq 0$, is a multiple of $\gcd(a,b)$. Furthermore, the largest multiple of $\gcd(a,b)$ that cannot be represented as $ax+by$, with $x, y \geq 0$, is $\mathrm{lcm}(a,b) - (a+b)$.*

Since the alphabet of a unary regular language has only one letter, the words of the language can be considered as a set of numbers by associating a^n with n (assuming $\Sigma = \{a\}$). In this way, the state complexity can be obtained from the analysis of these numbers. So a mathematical model for the DFA that accepts a unary regular language is designed as two numbers by G. Pighizzini and J. Shallit [83]. It has been shown in [83] that the transition diagram of a unary DFA A, with n states, has a "tail" consisting of $\mu \geq 0$ states and a "circle" of $\lambda \geq 1$ states. Furthermore, if the transition diagram is

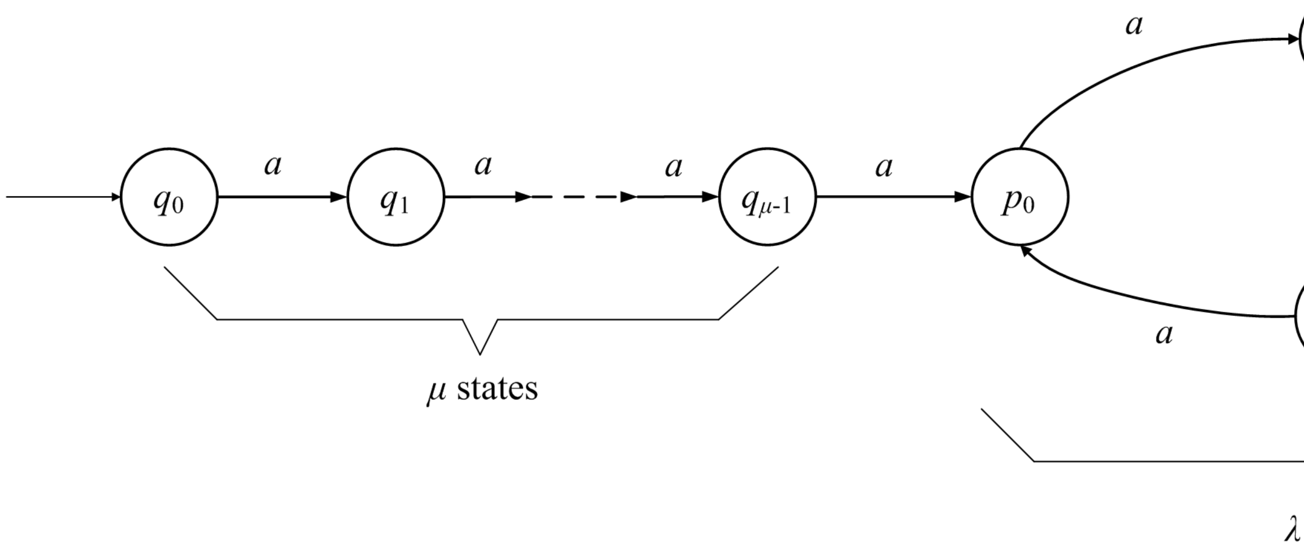

Figure 3.6: The model of DFAs that accept unary regular languages [83]

Theorem 3.10 *A unary language L is regular if and only if there are two integers $\mu \geq 0$, $\lambda \geq 1$, such that for any $n \geq \mu$, the word $a^n \in L$ if and only if $a^{n+\lambda} \in L$.*

Given a unary language L, the pair of integers (λ, μ) in Theorem 3.10 is the size of the DFA that accepts L, and more precisely [83]:

Theorem 3.11 *Given a unary regular language L and two integers $\mu \geq 0$, $\lambda \geq 1$, the following statements are equivalent:*

(i) L is accepted by a DFA of size (λ, μ);

(ii) for any $n \geq \mu$, $a^n \in L$ if and only if $a^{n+\lambda} \in L$.

A condition which characterizes minimal unary DFAs is presented in [76, 83]:

Theorem 3.12 *A unary DFA $A = (Q, \Sigma, \delta, q_0, F)$ of size (λ, μ) is minimal if and only if both the following conditions are satisfied:*

(i) For any maximal proper divisor d of λ (i.e., $\lambda = \alpha \cdot d$ for some prime number $\alpha > 1$) there exists an integer h, with $0 \leq h < \lambda$, such that $p_h \in F$ if and only if $p_{(h+d)\mathrm{mod}\lambda} \notin F$, i.e., $a^{\mu+h} \in L$ if and only if $a^{\mu+h+d} \notin L$;

(ii) $q_{\mu-1} \in F$ if and only if $p_{\lambda-1} \notin F$, i.e., $a^{\mu-1} \in L$ if and only if $a^{\mu+\lambda-1} \notin L$.

Using Theorem 3.12, Corollary 3.2 is shown and proved in [83].

Corollary 3.2 *Given two integers $\mu \geq 0$, $\lambda \geq 1$, let $L = a^{\mu+\lambda-1}(a^{\lambda})^*$. Then the size of the minimal DFA that accepts L is (λ, μ).*

State Complexity of Catenation

Theorems 3.13 and 3.14 in the following are given in [111] concerning the state complexity of catenation of unary regular languages.

Theorem 3.13 *Let m, n be two arbitrary positive integers such that $\gcd(m, n) = 1$. Then there exists an m-state DFA language R_1 and an n-state DFA language R_2, over a one-letter alphabet, such that any DFA that accepts R_1R_2 needs at least mn states.*

Theorem 3.14 *For any integers $m, n \geq 1$, let A and B be an m-state DFA and an n-state DFA, respectively, over a one-letter alphabet. Then there exists a DFA of at most mn states that accepts $L(A)L(B)$.*

Theorems 3.15 and 3.16 in the following are given in [83] concerning the sizes of DFAs for the catenation of unary regular languages.

Theorem 3.15 *Given any $\mu', \mu'' \geq 0$, $\lambda', \lambda'' \geq 1$, let L' and L'' be two unary languages accepted by two automata A' and A'' of size (λ', μ') and (λ'', μ''), respectively. Then, the catenation of L' and L'' is accepted by a DFA of size (λ, μ), where $\lambda = \mathrm{lcm}(\lambda', \lambda'')$ and $\mu = \mu' + \mu'' + \mathrm{lcm}(\lambda', \lambda'') - 1$.*

Theorem 3.16 *For any* $\mu', \mu'' \geq 2$, $\lambda', \lambda'' \geq 2$, *such that* $\gcd(\lambda', \lambda'') > 1$, *there exist two unary languages* L' *and* L'' *which are accepted by two DFAs* A' *and* A'' *of size* (λ', μ') *and* (λ'', μ''), *respectively, such that the catenation of* L' *and* L'' *is accepted by a DFA of size* (λ, μ), *with* $\lambda = lcm(\lambda', \lambda'')$ *and* $\mu = \mu' + \mu'' + \text{lcm}(\lambda', \lambda'') - 1$.

State Complexity of Star

Theorem 3.17 is given and proved in [111]. It shows the state complexity of star of unary regular languages.

Theorem 3.17 *The number of states that is both sufficient and necessary in the worst case for a DFA to accept the star of an* n*-state DFA language,* $n > 1$, *over a one-letter alphabet is* $(n-1)^2 + 1$.

State Complexity of Intersection and Union

Theorem 3.18 in the following is given in [111] concerning the state complexities of union and intersection of unary regular languages.

Theorem 3.18 *The number of states which is both sufficient and necessary in the worst case for a DFA to accept the intersection (union) of an* m*-state DFA language and an* n*-state DFA language,* $m, n > 1$, $\gcd(m, n) = 1$, *over a one-letter alphabet is* mn.

Theorems 3.19 and 3.20 in the following are given in [83] concerning the sizes of the DFAs for the union and intersection of unary regular languages.

Theorem 3.19 *Let* L' *and* L'' *be two languages accepted by unary automata* A' *and* A'' *of the size* (λ', μ') *and* (λ'', μ''), *respectively. The intersection (the union, respectively) of* L' *and* L'' *is accepted by a DFA of the size* $(\text{lcm}(\lambda', \lambda''), \max(\mu', \mu''))$.

Theorem 3.20 *For any* $\mu', \mu'' \geq 0$, $\lambda', \lambda'' \geq 1$, *there exist two languages* L' *and* L'' *which are accepted by DFAs of size* (λ', μ') *and* (λ'', μ''), *respectively, such that the minimal DFAs that accept* $L' \cup L''$ *and* $L' \cap L''$ *have both size* $(\text{lcm}(\lambda', \lambda''), \max(\mu', \mu''))$.

3.2 State Complexity of Individual Operations on Finite Languages

In this section, we assume that all the DFAs mentioned are reduced DFAs. The following theorems and corollaries about the state complexity of operations on finite languages have been proved in [13].

3.2.1 Finite Languages over a General Alphabet

State Complexity of Star

Theorems 3.21, 3.22 and Corollary 3.3 in the following are given in [13] concerning the state complexity of star of finite languages.

Theorem 3.21 *Let $A = (Q, \Sigma, \delta, 0, F)$ be a DFA that accepts a finite language L, where $0 \notin F$, $|F| = t \geq 2$, $|Q| = n \geq 4$. Then there exists a DFA of at most $2^{n-3} + 2^{n-t-2}$ states that accepts L^*.*

Corollary 3.3 *Let $A = (Q, \Sigma, \delta, 0, F)$ be a DFA that accepts a finite language L, where $|Q| = n > 4$. Then there exists a DFA of at most $2^{n-3} + 2^{n-4}$ states that accepts L^*.*

Theorem 3.22 *There exists a DFA $A = (Q, \Sigma, \delta, 0, F)$ with $|Q| = n \geq 4$ such that any DFA recognizing $(L(A))^*$ has at least $2^{n-3} + 2^{n-4}$ states.*

State Complexity of Catenation

We use the following notation:

$$\binom{n}{\leq i} = \sum_{j=0}^{i} \binom{n}{j}$$

Theorems 3.23, 3.24, Corollaries 3.4 and 3.5 in the following are given and proved in [13] concerning the state complexity of catenation of finite languages.

Theorem 3.23 *Let $A_i = (Q_i, \Sigma_i, \delta_i, 0, F_i)$, $i = 1, 2$, be two DFAs that accept finite languages L_i, respectively, where $|Q_1| = m$, $|Q_2| = n$, $|\Sigma| = k$ and $|F_1| =$*

t. *There exists a DFA* $A = (Q, \Sigma, \delta, 0, F)$ *that accepts* $L(A) = L(A_1)L(A_2)$ *and*

$$|Q| \leq \sum_{i=0}^{m-2} \min\left\{k^i, \binom{n-2}{\leq i}, \binom{n-2}{\leq t-1}\right\} + \min\left\{k^{m-1}, \binom{n-2}{\leq t}\right\}.$$

Corollary 3.4 *Let* $A_i = (Q_i, \Sigma_i, \delta_i, 0, F_i)$, $i = 1, 2$, *be two DFAs that accept finite languages* L_i, *respectively, where* $|Q_1| = m$, $|Q_2| = n$, $|F_1| = t$ *and* $t > 0$. *Then there exists a DFA* $A = (Q, \Sigma, \delta, 0, F)$ *of* $O(mn^{t-1} + n^t)$ *states that accepts* $L(A) = L(A_1)L(A_2)$.

Corollary 3.5 *Let* $A_i = (Q_i, \Sigma_i, \delta_i, 0, F_i)$, $i = 1, 2$, *be two DFAs that accept finite languages* L_i, *respectively, where* $|Q_1| = m$, $|Q_2| = n$, $|\Sigma| = k$, $k = 2$ *and* $m + 1 \geq n > 2$. *Then there exists a DFA* $A = (Q, \Sigma, \delta, 0, F)$ *of* $(m - n + 3)2^{n-2} - 1$ *states that accepts* $L(A) = L(A_1)L(A_2)$.

Theorem 3.24 *The bound given by Corollary 3.5 for* $k = 2$ *is attainable.*

State Complexity of Reversal

Theorems 3.25, 3.26 and Corollary 3.6 in the following are shown and proved in [13] concerning the state complexity of reversal of finite languages.

Theorem 3.25 *Let* $A = (Q, \Sigma, \delta, 0, F)$ *be a DFA that accepts a finite language* L, *where* $|Q| = n \geq 3$, $|\Sigma| = k \geq 2$. *Let* t *be the smallest integer such that* $2^{n-1-t} \leq k^t$. *Then there exists a DFA, with*

$$|Q_B| \leq \sum_{i=0}^{t-1} k^i + 2^{n-1-t},$$

that accepts the reversal of L.

Corollary 3.6 *Let* $|\Sigma| = 2$ *and* A *be a DFA of* $n \geq 3$ *states that accepts a finite language* $L \subseteq \Sigma^*$. *Then there exists a DFA* B *that accepts the reversal of* L *such that* B *has at most* $3 \cdot 2^{p-1} - 1$ *states if* $n = 2p$ *or* $2^p - 1$ *states if* $n = 2p - 1$.

Theorem 3.26 *The bounds given by Corollary 3.6 are attainable.*

3.2.2 Finite Languages over a One-letter Alphabet

Note that if DFA $A = (Q, \{a\}, \delta, 0, F)$ is a minimal DFA that accepts words whose largest length is l, then $|Q| = l + 1$.

Theorem 3.27 is given in [13]. It shows the state complexities of operations on finite languages.

Theorem 3.27 *Let $A_i = (Q_i, \{a\}, \delta_i, 0, F_i)$, $i = 1, 2$, be two minimal DFAs with $|L(A_i)| < \infty$, $|Q_1| = m$, $|Q_2| = n$. Let $A = (Q, \{a\}, \delta, 0, F)$, $|Q| = k$, be a minimal DFA. Then we have the following:*

a) If $L(A) = L(A_1) \cup L(A_2)$, then $k = \max\{m, n\}$.

b) If $L(A) = L(A_1) \cap L(A_2)$, then $k \leq \min\{m, n\}$.

c) If $L(A) = L(A_1) - L(A_2)$, then $k \leq m$.

d) If $L(A) = \{a\}^* - L(A_1)$, then $k = m$.

e) If $L(A) = L(A_1)L(A_2)$, then $k = m + n - 1$.

f) If $L(A) = L(A_1)^*$, then $k \leq m^2 - 7m + 13$ for $m > 4$ and $m = 3$, $k \leq 2$ otherwise.

g) If $L(A) = a \setminus L(A_1)$, then $k = m - 1$.

h) If $L(A) = L(A_1)^R$, then $k = m$.

3.3 Conclusion

After reviewing the above results on the state complexities of operations on regular languages, we find that most basic individual operations have been studied. Tight bounds have been found and proved.

We assume that languages L_1 and L_2 are accepted by an m-state DFA A_1 and an n-state DFA A_2, respectively, $m, n > 1$. L_1 and L_2 are over the same alphabet Σ. The state complexities of individual operations on regular languages are listed in Table 3.1.

From Table 3.1, we know that the state complexity of an individual operation on regular languages over a general alphabet may not be the same as its state complexity on unary regular languages. Note that the state complexities

Table 3.1: The state complexities of individual operations on regular languages [105]

	$\|\Sigma\| = 1$	$\|\Sigma\| > 1$
$L_1 \cup L_2$	mn, for $\gcd(m, n) = 1$	mn
$L_1 \cap L_2$	mn, for $\gcd(m, n) = 1$	mn
$\Sigma^* - L_1$	m	m
L_1L_2	mn, for $\gcd(m, n) = 1$	$m2^n - 2^{n-1}$
${L_1}^R$	m	2^m
${L_1}^*$	$(m-1)^2 + 1$	$2^{m-1} + 2^{m-2}$

of some individual operations on regular languages over a two-letter alphabet remain open.

Using these results as the foundation, we can start the study of the state complexity of combined operations on regular languages. The direct compositions of the state complexities of these individual operations can give the upper bounds for combined operations. In the next chapter, we will study whether these bounds are tight or not.

Chapter 4

Recent Results on State Complexity of Combined Operations

The research on state complexity of combined operations started in 2005. Up to now, the state complexities of some combined operations have been studied, e.g., star of union and intersection, star of catenation and reversal, reversal of union and intersection, reversal of catenation and star, etc. [23, 62, 70, 86, 92]. These results are reviewed in this chapter. We will start with the state complexities of star of union and star of intersection in the following.

4.1 State Complexity of Star of Union and Star of Intersection

It is known that the state complexity of the union operation on a DFA of m_1 states and a DFA of m_2 states is $m_1 \cdot m_2$. The state complexity of the star operation on a DFA of m states is $2^{m-1} + 2^{m-2}$. The direct composition of state complexities of the union and star operations on regular languages is

$$2^{m_1 m_2 - 1} + 2^{m_1 m_2 - 2}.$$

However, the actual state complexity of the star of a union is very different from the composition of the state complexities of the individual operations [62, 92].

Theorem 4.1 *Let $L_i = L(A_i)$ and A_i be a complete DFA of m_i states, $i = 1, 2$. Then $(L_1 \cup L_2)^*$ is accepted by a complete DFA of no more than $2^{m_1+m_2-1} - 2^{m_1-1} - 2^{m_2-1} + 1$ states.*

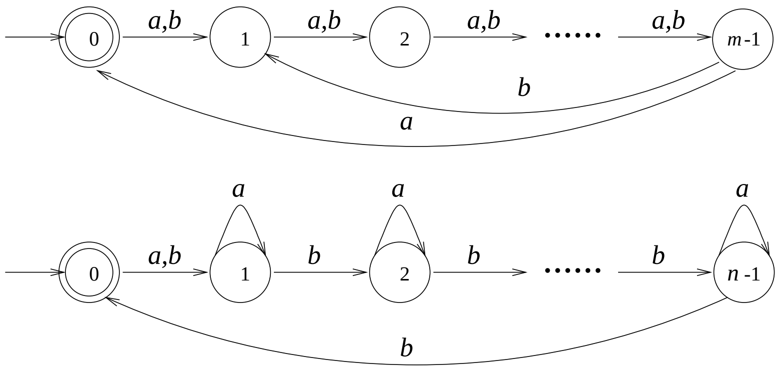

Figure 4.1: Witness DFAs A_1 and A_2 for the star of a union

Note that this upper bound

$$2^{m_1+m_2-1} - 2^{m_1-1} - 2^{m_2-1} + 1$$

is much smaller than

$$2^{m_1 m_2-1} + 2^{m_1 m_2-2}.$$

This upper bound has been proved to be attainable with the witness DFAs shown in Figure 4.1 [62, 92].

Theorem 4.2 *For all integers $m_1 > 2$ and $m_2 > 2$, there exist binary DFAs A and B of m_1 and m_2 states, respectively, such that the state complexity of the language $(L(A) \cup L(B))^*$ is $2^{m_1+m_2-1} - 2^{m_1-1} - 2^{m_2-1} + 1$.*

After considering other cases, the following corollary has been obtained in [62].

Corollary 4.1 *For every alphabet Σ, such that $|\Sigma| \geq 2$, the state complexity of the star of a union over Σ is:*

$$f(m_1, m_2) = \begin{cases} 2^{m_1+m_2-1} - 2^{m_1-1} - 2^{m_2-1} + 1, & \text{if } m_1,\ m_2 \geq 2, \\ 3 \cdot 2^{m_1-2}, & \text{if } m_1 \geq 2,\ m_2 = 1, \\ 3 \cdot 2^{m_2-2}, & \text{if } m_1 = 1,\ m_2 \geq 2, \\ 2, & \text{if } m_1 = m_2 = 1. \end{cases}$$

The state complexity of intersection operation on a DFA of m_1 states and a DFA of m_2 states is also $m_1 \cdot m_2$. Thus, the direct composition of state complexities of the intersection and star operations on regular languages is also

$$2^{m_1 m_2-1} + 2^{m_1 m_2-2}.$$

This is an upper bound on the state complexity of the star of an intersection [92]. It has been proved to be attainable by some witness DFAs shown in Figure 4.2 [62].

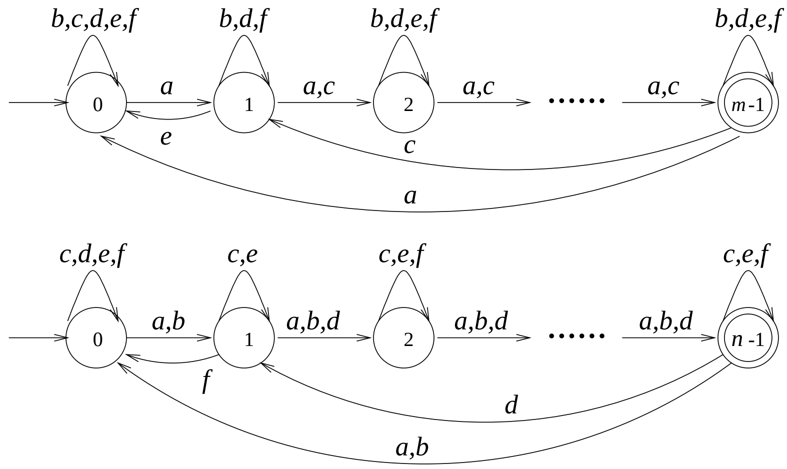

Figure 4.2: Witness DFAs A and B for the star of an intersection

Theorem 4.3 *For all integers $m_1 > 2$ and $m_2 > 2$, there exist DFAs A and B of m and n states, respectively, defined over a six-letter alphabet, such that the state complexity of the language $(L(A) \cap L(B))^*$ is $2^{m_1 m_2 - 1} + 2^{m_1 m_2 - 2}$.*

4.2 State Complexity of Star of Catenation and Star of Reversal

The state complexity of the combined operation the star of a catenation has been proved to be much smaller than the direct composition of state complexities of catenation and star [28, 31].

Theorem 4.4 *Let L_1 and L_2 be two regular languages accepted by an m-state and n-state DFA, respectively, $m, n \geq 2$. Then there exists a DFA of at most $2^{m+n-1} + 2^{m+n-4} - 2^{m-1} - 2^{n-1} + m + 1$ states that accepts $(L_1 L_2)^*$.*

The state complexity of the star of the catenation of an m-state DFA language and a one-state DFA language is at most $m + 1$. The state complexity of the star of a catenation of a one-state DFA language and an n-state DFA language is upper bounded by 2^n.

It has also been shown in [28, 31] that the upper bound in Theorem 4.4 is tight, when $m \geq 2$ and $n \geq 2$. Figure 4.3 shows the transition diagrams of the witness DFAs A and B.

Theorem 4.5 *For all integers $m \geq 2$ and $n \geq 2$, there exist DFAs A and B of m states and n states, respectively, defined over a four-letter input alphabet and such that any DFA that accepts $(L(A)L(B))^*$ needs at least $2^{m+n-1} + 2^{m+n-4} - 2^{m-1} - 2^{n-1} + m + 1$ states.*

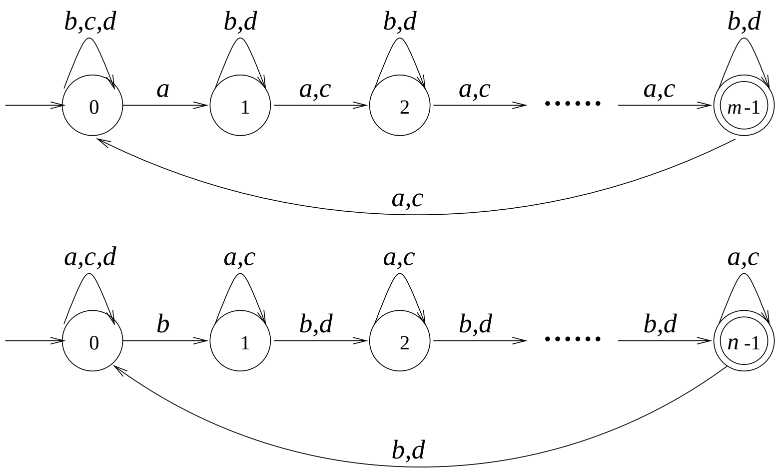

Figure 4.3: Witness DFAs A and B for the star of a catenation

The state complexity of the star of a reversal on regular languages cannot attain the direct composition of the state complexities of its component operations either [28, 31].

Lemma 4.1 *Let $A = (Q, \Sigma, \delta, s, F)$ be a DFA, where $|Q| = n \geq 2$ and $|(F - \{s\})| = k$ with $1 \leq k \leq n-1$, and $L = L(A)$. Then there exists another DFA $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ of no more than $2^n - (2^k - 1)2^{n-k-1} + 1$ states that accepts $(L^R)^*$.*

After considering the case when $n = 1$, the following lemma concerning an upper bound on the state complexity of $(L^R)^*$ has been obtained [28, 31].

Lemma 4.2 *Let $A = (Q, \Sigma, \delta, s, F)$ be a DFA of $n > 0$ states and $L = L(A)$. Then there exists a DFA $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ of no more than 2^n states that accepts $(L^R)^*$.*

The witness DFA A for the state complexity of the star of a reversal is the same as the witness DFA for the state complexity of reversal operation on regular languages. The transition diagram of A is shown in Figure 3.5. It is the worst-case example attaining the upper bound precisely [28, 31].

Lemma 4.3 *There exists an n-state DFA A, for any $n > 0$, such that any DFA that accepts $(L(A)^R)^*$ has at least 2^n states.*

By Lemmas 4.1, 4.2 and 4.3, the following theorem has been concluded [28, 31].

Theorem 4.6 *The state complexity of the star of a reversal of an n-state DFA language is exactly 2^n, for any $n > 0$.*

4.3 State Complexity of Reversal of Union and Reversal of Intersection

The state complexity of the reversal of an m-state DFA language is known to be 2^m. As we mentioned before, the state complexity of the union of an m_1-state DFA language and an m_2-state DFA language is $m_1 \cdot m_2$. Thus, the direct composition of state complexities of the union and reversal operations on regular languages is $2^{m_1 m_2}$. In fact, this upper bound is not attainable [70]. It can be reduced to $2^{m_1+m_2} - 2^{m_1} - 2^{m_2} + 2$ which can be attained by the witness DFAs shown in Figure 4.4 [70].

Theorem 4.7 *Let L_1 and L_2 be languages accepted by m_1-state and m_2-state complete DFAs, $m_1, m_2 \geq 3$. Then the state complexity of the combined operation $(L_1 \cup L_2)^R$ is $2^{m_1+m_2} - 2^{m_1} - 2^{m_2} + 2$.*

The direct composition of state complexities of the intersection and reversal operations on regular languages is also $2^{m_1 \cdot m_2}$. However, the actual state complexity of this combined operation is the same as that of the reversal of union, $2^{m_1+m_2} - 2^{m_1} - 2^{m_2} + 2$ [70], since

$$(L_1 \cap L_2)^R = \overline{(\overline{L_1} \cup \overline{L_2})}^R = \overline{(\overline{L_1} \cup \overline{L_2})^R}.$$

Theorem 4.8 *Let L_1 and L_2 be languages accepted by m_1-state and m_2-state complete DFAs, $m_1, m_2 \geq 3$. Then the state complexity of the combined operation $(L_1 \cap L_2)^R$ is $2^{m_1+m_2} - 2^{m_1} - 2^{m_2} + 2$.*

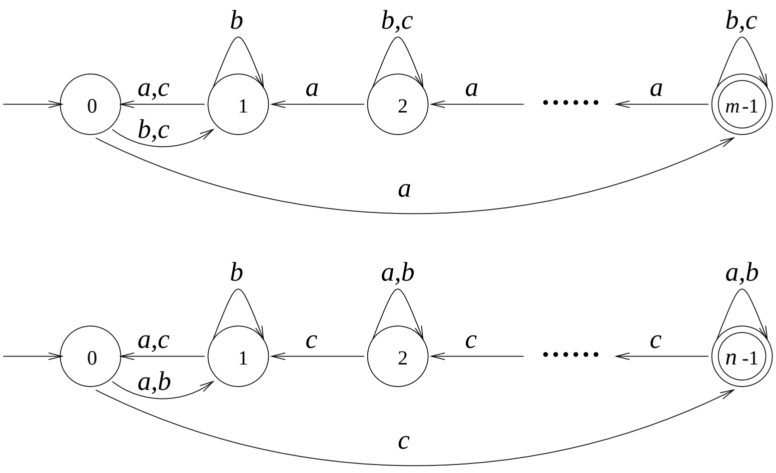

Figure 4.4: Witness DFAs A_1 and A_2 for both the reversal of a union and the reversal of an intersection

4.4 State Complexity of Reversal of Catenation and Reversal of Star

For the state complexity of the reversal of a catenation, only an upper bound has been obtained [70].

Theorem 4.9 *Let L_1 and L_2 be languages accepted by m-state and n-state complete DFAs, 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 $(L_1 L_2)^R$.*

Since $(L^*)^R = (L^R)^*$, the state complexity of $(L^*)^R$ is the same as that of $(L^R)^*$ [70].

Theorem 4.10 *Let L be a language accepted by an n-state DFA. The state complexity of the reversal of star operation on L is exactly 2^n, for any $n > 0$.*

4.5 State Complexity of Power

The state complexity of the power of a language: L^k, has been studied in [23, 86]. The state complexity of L^2 has been shown to be at most $n2^n - 2^{n-1}$, where L is a language accepted by an n-state DFA. This bound can be attained for any $n \geq 3$ over an alphabet of size two [86]. The following results concern the state complexity of L^k for $k \geq 2$ [23].

Theorem 4.11 *For every n-state regular language L, with $n \geq 1$, the language L^k requires at most $n2^{(k-1)n}$ states. Furthermore for every $k \geq 2$, $n \geq k+1$, and alphabet Σ with $|\Sigma| \geq 6$, there exists an n-state regular language $L \subseteq \Sigma^*$ such that L^k requires at least $(n-k)2^{(k-1)(n-k)}$ states.*

The worst-case example used in Theorem 4.11 is a sequence of automata $A_{k,n}$ $(2 \leq k < n)$ over the alphabet $\Sigma = \{a, b, c, d, e, f\}$. Let each $A_{k,n}$ have a set of states $Q = \{0, 1, \ldots, n-1\}$, of which 0 is the initial state, $n-1$ is the sole final state, and where the transitions are defined as follows:

$$\delta(j,a) = \begin{cases} j+1, \text{ if } 1 \leq j \leq n-k-1; \\ 1, \text{ if } j = n-k; \\ j, \text{ otherwise,} \end{cases} \qquad \delta(j,b) = \begin{cases} j+1, \text{ if} \\ \qquad n-k+1 \leq j \leq n-2; \\ n-k+1, \text{ if } j = n-1; \\ j, \text{ otherwise,} \end{cases}$$

$$\delta(j,c) = \begin{cases} 1, \text{ if } j = 0; \\ 0, \text{ if } j = 1; \\ j, \text{ otherwise,} \end{cases} \qquad \delta(j,d) = \begin{cases} 1, \text{ if } j = n-k+1; \\ j, \text{ otherwise,} \end{cases}$$

$$\delta(j,e) = \begin{cases} n-1, \text{ if } j = 0; \\ j-1, \text{ if } n-k+2 \leq j \leq n-1; \\ j, \text{ otherwise,} \end{cases} \qquad \delta(j,f) = \begin{cases} n-1, \text{ if } j = 1; \\ n-2, \text{ otherwise.} \end{cases}$$

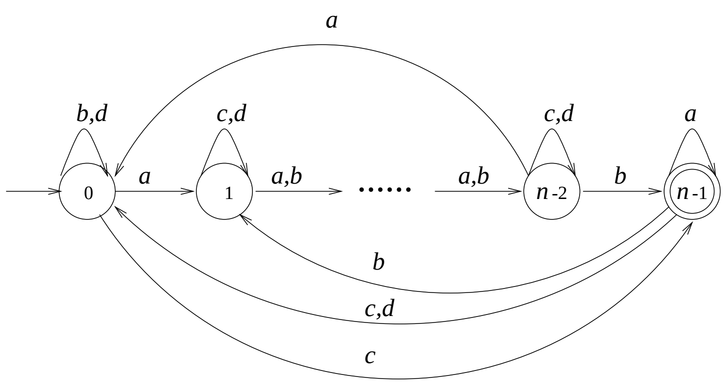

Figure 4.5: Witness DFA A_n for L^3

The state complexity of L^3 has also been obtained [23].

Theorem 4.12 *For every n-state regular language L, with $n \geq 3$, the state complexity of L^3 is at most $\frac{6n-3}{8}4^n - (n-1)2^n - n$. This upper bound is attained on every alphabet of at least four letters.*

Figure 4.5 shows the witness DFA A_n for the combined operation L^3. It has been pointed out that any witness for the worst-case state complexity of L^3 is a witness for L^2 as well [23].

4.6 Conclusion

In this chapter, we have reviewed the state complexities of 10 combined operations on regular languages. These results are shown in Table 4.1. We assume that L_1 and L_2 are accepted by an m-state DFA $A_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ and an n-state DFA $A_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$, respectively, and $m, n \geq 3$.

Table 4.1: The state complexities of 10 combined operations

Operation	State complexity
$(L_1 \cup L_2)^*$	$2^{m+n-1} - 2^{m-1} - 2^{n-1} + 1$ [92]
$(L_1 \cap L_2)^*$	$2^{mn-1} + 2^{mn-2}$ [62]
$(L_1 L_2)^*$	$2^{m+n-1} + 2^{m+n-4} - 2^{m-1} - 2^{n-1} + m + 1$ [31]
$(L_1^R)^*$	2^m [31]
$(L_1 \cup L_2)^R$	$2^{m+n} - 2^m - 2^n + 2$ [70]
$(L_1 \cap L_2)^R$	$2^{m+n} - 2^m - 2^n + 2$ [70]
$(L_1 L_2)^R$	$O(2^{mn-1})$ [70]
$(L_1{}^*)^R$	2^m [70]
$L_1{}^3$	$\frac{6m-3}{8}4^m - (m-1)2^m - m$ [23]
$L_1{}^k$	$\theta(m2^{(k-1)m})$ [23]

We can see that for most of these combined operations, their state complexities are very different from the direct compositions of the state complexities of their component operations. There is only one combined operation, star of intersection, whose state complexity is exactly the same as the direct combination of the state complexities of intersection and star.

Thus, although the direct combination of the state complexities of individual operations can provide an upper bound on the state complexity of a combined operation, this upper bound may not be tight [92].

Chapter 5

Exact State Complexity of Combined Operations

In this chapter, we investigate the exact state complexities of combined operations on regular languages. We choose 12 combined operations and investigate their exact state complexities, including combinations of union, intersection and complementation, multiple catenations, combinations of star and catenation, reversal and catenation, Boolean operations and catenation, Boolean operations and star, and Boolean operations and reversal. These combined operations are widely used in practice. For example, the state complexity of $L_1L_2^R$ is equal to that of catenation combined with antimorphic involution ($L_1\theta(L_2)$). An antimorphic involution is the natural formalization of the notion of Watson-Crick complementarity in biology. The combination of catenation and antimorphic involution can naturally formalize a basic biological operation: primer extension [2].

We will first study the state complexities of catenation combined with star and reversal in the following.

5.1 State Complexity of Catenation Combined with Star and Reversal

5.1.1 State Complexity of $L_1^*L_2$

In this subsection, we investigate the state complexity of $L(A)^*L(B)$ for two DFAs A and B of sizes $m, n \geq 1$, respectively. All the results in this subsection are from our paper [12].

We first notice that, when $n = 1$, the state complexity of $L(A)^*L(B)$ is 1 for any $m \geq 1$. 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 one-state DFA. Next, we consider the case when A has only one final state and it 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 $m \geq 1$ in this form and any B of size $n \geq 2$, 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 $n \geq 2$. We will show that its upper bound (Theorem 5.3) coincides with its lower bound (Theorem 5.4), and the state complexity is $5 \cdot 2^{m+n-3} - 2^{m-1} - 2^n + 1$ [12].

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

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

Proof: Let $A = (Q_1, \Sigma, \delta_1, s_1, \{s_1\})$ and $B = (Q_2, \Sigma, \delta_2, s_2, F_2)$ be two DFAs with m and n states, respectively. We construct the DFA $C = (Q, \Sigma, \delta, s, F)$ such that

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

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 $L(C) = L(A)L(B)$. *q.e.d.*

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

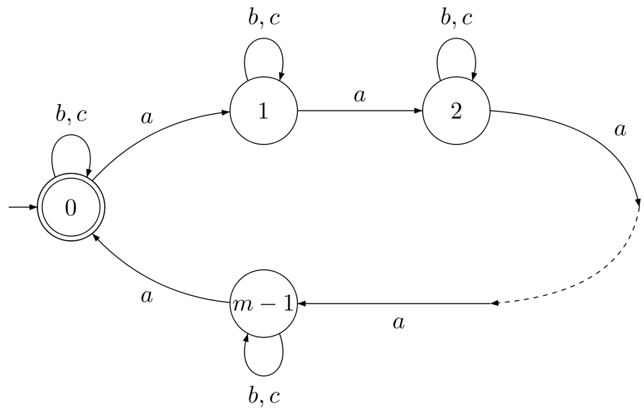

Figure 5.1: Witness DFA A for Theorem 5.2 when $m \geq 2$

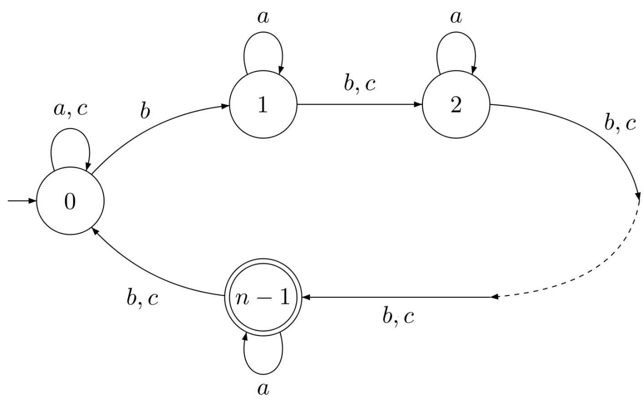

Figure 5.2: Witness DFA B for Theorem 5.2 when $m \geq 2$

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

Proof: When $m = 1$, the witness DFAs used in the proof of Theorem 1 in [111] can be used to show that the upper bound proposed in Theorem 5.1 can be attained.

Next, we consider the case when $m \geq 2$. We provide witness DFAs A and B, depicted in Figures 5.1 and 5.2, respectively, over the three-letter alphabet $\Sigma = \{a, b, c\}$.

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

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

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

- $\delta_2(i, a) = i$, 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 the 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 states in Q are reachable from s, and (II) any two different states in Q are not equivalent.

For (I), we show that all states 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 a^i, and the state $\langle i, \{j\} \rangle$ can be reached from state $\langle i, \{0\} \rangle$ on b^j, for any $i \in \{1, \ldots, m-1\}$ and $j \in Q_2$.

In the induction step, we assume that all 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 $0 \leq j_1 < j_2 < \cdots < j_k \leq n-1$. 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 $c^{j_2 - 1}$.

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 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 $w = c^{n-1}a^{m-q}b^n$ distinguishes the two states, since $\delta(\langle q, T\rangle, w) \in F$ and $\delta(\langle q', T'\rangle, w) \notin F$.

2. $q = q'$ and $T \neq T'$. Without loss of generality, we assume that $|T| \geq |T'|$. Then there exists a state $j \in T - T'$. It is clear that, when $q \neq 0$, b^{n-1-j} distinguishes the two states, and when $q = 0$, c^{n-1-j} distinguishes the two states since j cannot be 0.

From (I) and (II), the DFA C has at least $m(2^n - 1) - 2^{n-1} + 1$ states and is minimal. *q.e.d.*

In the rest of this subsection, we focus on the case when the DFA A contains at least one final state that is not the initial state. Thus, this DFA is of size at least two. 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 $(\frac{3}{4}2^m - 1)(2^n - 1) - (2^{m-1} - 2^{m-k_1-1})(2^{n-1} - 1)$ 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 the 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 $Q = \{\langle p, t\rangle \mid p \in P \text{ and } t \in T\} \cup \{\langle p', t'\rangle \mid p' \in P' \text{ and } t' \in T'\}$, where

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

The initial state 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 the DFA B by ε-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 $t \in 2^{Q_2}$ are never reached in the 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 either of them is final if $s_2 \notin F_2$, and they are both final states otherwise. Hence, we merge these 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 second component does not contain the initial state of B.

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

$$(\frac{3}{4}2^m - 1)(2^n - 1) - (2^{m-1} - 2^{m-k_1-1})(2^{n-1} - 1).$$

q.e.d.

Next we show that this upper bound is attainable by some witness DFAs.

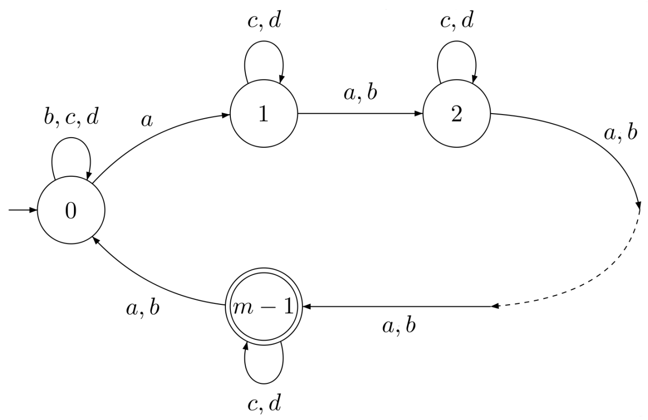

Figure 5.3: Witness DFA A for Theorem 5.4

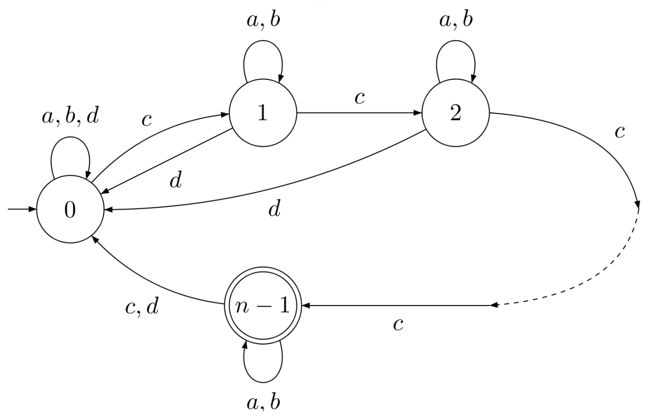

Figure 5.4: Witness DFA B for Theorem 5.4

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

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

Let $A = (Q_1, \Sigma, \delta_1, 0, \{m-1\})$, as shown in Figure 5.3, 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 $B = (Q_2, \Sigma, \delta_2, 0, \{n-1\})$, as shown in Figure 5.4, 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 for 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.

Basis: for any $i \in Q_2$, the state $\langle\{0\}, \{i\}\rangle$ can be reached from the initial state $\langle\{0\}, \{0\}\rangle$ on c^i. Then by the proof of Theorem 5 in [111], 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 strings over letters a and b.

Induction: assume that all 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 $0 \le j_1 < j_2 < \cdots < j_k \le n-1$.

Note that the 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 the 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, by using the proof of Theorem 5 in [111], the states $\langle p, t\rangle$ in Q, where $p \in P$ and $|t| = k$, can be reached from the state $\langle \{0\}, t\rangle$ on strings over letters a and b.

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. $p \neq p'$. Without loss of generality, we assume $|p| \geq |p'|$. Then there exists a state $i \in p - p'$. It is clear that $a^{m-1-i}dc^n$ 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 that $|t| \geq |t'|$ and let $j \in t - t'$. Then the state $\langle p, t\rangle$ reaches a final state on 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 $j \neq 0$.

From (I) and (II), the 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. *q.e.d.*

5.1.2 State Complexity of $L_1L_2^*$

In this subsection, we consider the state complexity of $L_1L_2^*$ where L_1 and L_2 are two languages accepted by two DFAs of sizes m and n, respectively. All the results in this subsection are from our paper [10].

We notice that if the n-state DFA has only one final state that is also its initial state, this DFA also accepts L_2^*. Thus, in such a case, an upper bound for the number of states of any DFA that accepts $L_1L_2^*$ is given by the state complexity of catenation as $m2^n - 2^{n-1}$. We first show that this upper bound is attainable by some DFAs of this form. Next we consider the state complexity of $L_1L_2^*$ in the other cases, that is when the n-state DFA contains some final states other than the initial state.

First, we show that there exist two DFAs A and B, where the latter DFA has only one final state that is also its initial state, such that the number of states of any DFA for $L(A)L(B)^*$, which is equal to $L(A)L(B)$, attains the upper bound given by the state complexity of catenation. One example can be obtained by slightly modifying the examples used in [58]. We change the initial state of the DFA B in [58] into the only final state, and obtain the following result [10]:

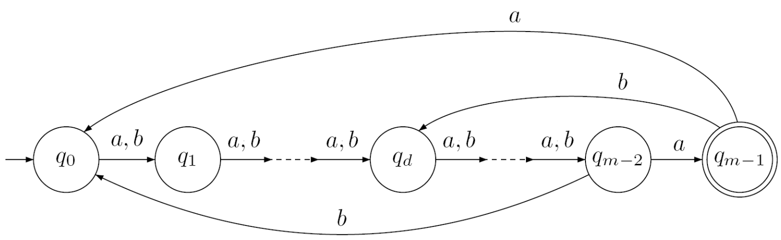

Figure 5.5: Witness DFA A for Lemma 5.1: $d = (m - n + 1) \bmod (m - 1)$

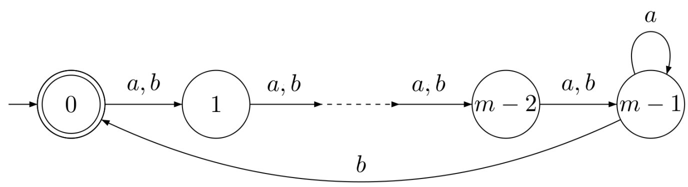

Figure 5.6: Witness DFA B for Lemma 5.1

Lemma 5.1 *For any $m \geq 2$ and $n \geq 2$, there exists a DFA A of m states and a DFA B of n states, where B has only one final state and it is also the initial state, such that any DFA for the language $L(A)L(B)$, which is equal to $L(A)L(B)^*$, has at least $m2^n - 2^{n-1}$ states.*

Proof: We use the DFAs A, as in Figure 5.5, and B, as in Figure 5.6, which are originally from [58] and we only modify the final state of the DFA B. For the sake of clarity, we repeat the definitions of these DFAs.

Let $A = (Q_A, \Sigma, \delta_A, q_0, F_A)$ be a DFA, where $Q_A = \{q_0, q_1, \ldots, q_{m-1}\}$, $\Sigma = \{a, b\}$, $F_A = \{q_{m-1}\}$, and for any $i \in \{0, 1, \ldots, m-1\}$,

$$\delta_A(q_i, X) = \begin{cases} q_j, j = (i+1) \bmod m, & \text{if } X = a; \\ q_{i+1}, & \text{if } i \leq m-3 \text{ and } X = b; \\ q_0, & \text{if } i = m-2 \text{ and } X = b; \\ q_d, d = (m-n+1) \bmod (m-1), & \text{if } i = m-1 \text{ and } X = b. \end{cases}$$

Let $B = (Q_B, \Sigma, \delta_B, 0, F_B)$ be a DFA, where $Q_B = \{0, 1, \ldots, n-1\}$, $F_B = \{0\}$, and for any $i \in \{0, 1, \ldots, n-1\}$,

$$\delta_B(i, X) = \begin{cases} i+1, & \text{if } i \leq n-2 \text{ and } X = a; \\ n-1, & \text{if } i = n-1 \text{ and } X = a; \\ (i+1) \bmod n, & \text{if } X = b. \end{cases}$$

We construct the DFA $C = (Q_C, \Sigma, \delta_C, \{q_0\}, F_C)$ that accepts $L(A)L(B)$ following the construction described in [58]. Note that the state set Q_C and transition rules δ_C are exactly the same as those of the DFA C' constructed in [58] and only the final state sets are different. In the DFA C, if a state contains the state 0 of B, it is a final state, but, in the DFA C' in [58], it is a final state if it contains the state $n-1$. Thus, the state set of Q_C is

$$Q_C = \{\{q_i\} \cup S \mid 0 \leq i \leq m-2 \text{ and } S \subseteq Q_B\} \cup \{\{q_{m-1}\} \cup S \mid S \subseteq Q_B - 0\}\},$$

and the size of Q_C is $m2^n - 2^{n-1}$. To prove the lemma, it is sufficient to show that (1) any state in Q_C is reachable and (2) no two different states in Q_C are equivalent. Since the state set Q_C and transition rules δ_C are the same as those of the DFA C' in [58], the proof for the reachability of states is the same, and hence is omitted. Therefore, we only prove (2) as follows.

Let $\{q_i\} \cup S$ and $\{q_j\} \cup T$ be two different states in Q_C with $0 \leq i \leq j \leq m-1$. There are two cases:

1. $i < j$. Then the string $a^{m-1-i}b^n$ is accepted by the DFA C starting from the state $\{q_i\} \cup S$, but it is not accepted starting from the state $\{q_j\} \cup T$.

2. $i = j$. Without loss of generality, there is a state l in Q_B such that $l \in S$ and $l \notin T$. Note that $l \geq 1$ if $i = j = m-1$. Then b^{n-l} is accepted by the DFA C starting from the state $\{q_i\} \cup S$, but not accepted starting from the state $\{q_j\} \cup T$. *q.e.d.*

Note that if $n = 1$, according to Theorem 3 in [111], for any DFA A of size $m \geq 1$, the state complexity of a DFA that accepts $L(A)L(B)$ ($L(A)L(B)^*$) is m.

In the rest of this subsection, we only consider the cases when the DFA for L_2 contains at least one final state that is not the initial state. Thus, this latter DFA is of size at least two.

When considering the size of the former DFA, we notice that, when the size of this DFA is one, the state complexity of $L_1L_2^*$ is one.

Lemma 5.2 *Let A be a DFA of one state and B be a DFA of $n \geq 1$ states. Then the sufficient and necessary number of states for a DFA to accept $L(A)L(B)^*$ is one.*

Proof: Since A is complete, $L(A)$ is either $\emptyset$ or Σ^*. We need to consider only the case when $L(A) = \Sigma^*$. Then we have $\Sigma^* \subseteq L(A)L(B)^* \subseteq \Sigma^*$. Thus, $L(A)L(B)^* = \Sigma^*$, and it is accepted by a DFA of one state. *q.e.d.*

Now, we focus on the cases when $m > 1$ and $n > 1$, and propose an upper bound for the state complexity of $L_1L_2^*$.

Theorem 5.5 *Let $A = (Q_1, \Sigma, \delta_1, s_1, F_1)$ be a DFA such that $|Q_1| = m > 1$ and $|F_1| = k_1$, and $B = (Q_2, \Sigma, \delta_2, s_2, F_2)$ be a DFA such that $|Q_2| = n > 1$ and $|F_2 - \{s_2\}| = k_2 \geq 1$. Then there exists a DFA of at most $m(2^{n-1} + 2^{n-k_2-1}) - k_1 2^{n-k_2-1}$ states that accepts $L(A)L(B)^*$.*

Proof: We denote $F_2 - \{s_2\}$ by F_0. Then $|F_0| = k_2 \geq 1$.

We construct the DFA $C = \{Q, \Sigma, \delta, s, F\}$ for the language $L_1L_2^*$, where L_1 and L_2 are the languages accepted by DFAs A and B, respectively. Intuitively, C is constructed by first constructing the DFA B' that accepts L_2^*, then catenating A with this new DFA. By careful examination, we can check that the states of B' are s_2' and the elements in $P - \{\emptyset\}$, where s_2' is the additional initial and final state in the construction and P is defined below. As the state set we choose

$$Q = \{r \cup p \mid r \in R \text{ and } p \in P\}, \text{ where for } q_i \in Q_1$$

$$R = \begin{cases} \{q_i\}, & \text{if } q_i \notin F_1; \\ \{q_i, s_2'\}, & \text{otherwise,} \end{cases}$$

$$P = \{S \mid S \subseteq Q_2 - F_0\} \cup \{T \mid T \subseteq Q_2, s_2 \in T, \text{ and } T \cap F_0 \neq \emptyset\},$$

$$s = \begin{cases} \{s_1\} \cup \{\emptyset\}, & \text{if } s_1 \notin F_1; \\ \{s_1, s_2'\} \cup \{\emptyset\}, & \text{otherwise.} \end{cases}$$

The set of final states F is chosen to be $F = \{S \in Q \mid S \cap (F_2 \cup \{s_2'\}) \neq \emptyset\}$.

We denote a state in Q by $\{q_i\} \cup G_1 \cup G_2$, where $q_i \in Q_1$, $G_1 \subseteq \{s_2'\}$, and $G_2 \subseteq Q_2$. Then the transition relation δ is defined as follows:

$$\delta(\{q_i\} \cup G_1 \cup G_2, a) = D_1 \cup D_2 \cup D_3, \text{ for any } a \in \Sigma, \text{ where}$$

D_1: If $\delta_1(q_i, a) = q_i' \in F_1$, $D_1 = \{q_i', s_2'\}$, otherwise, $D_1 = \{q_i'\}$.

$$D_2 = \begin{cases} \emptyset, & \text{if } G_1 = \emptyset; \\ \delta_2(s_2, a), & \text{if } \delta_2(s_2, a) \cap F_0 = \emptyset; \\ \delta_2(s_2, a), \cup \{s_2\} & \text{otherwise,} \end{cases}$$

$$D_3 = \begin{cases} \emptyset, & \text{if } G_2 = \emptyset; \\ \delta_2(G_2, a), & \text{if } \delta_2(G_2, a) \cap F_0 = \emptyset; \\ \delta_2(G_2, a), \cup \{s_2\} & \text{otherwise.} \end{cases}$$

We can verify that the DFA C indeed accepts $L_1 L_2^*$. The computation of C always starts with the initial state of A, and after reaching a final state of A, it also reaches s_2' by the ε-transition of the catenation operation. Up to this point, the states of Q we have visited contain only one state q of A, and s_2' if q is a final state. After reaching some states of B', the computation simulates the transition rules of both A and B'. It is clear that each state in Q should consist of exactly one state in Q_1 and the states in one element of $P - \{\emptyset\}$. Moreover, if a state of Q contains a final state of A, then this state also contains the state s_2'. The transition rules of A are simulated by D_1, and the transition rules of B' are simulated by D_2 and D_3. We should notice that the simulation of A is deterministic. Finally, due to the construction of B', any state in Q that contains either the state s_2' or a final state of B is a final state of C.

To get an upper bound for the state complexity of catenation combined with star, we should count the number of states in Q. However, as we will show in the following, some states in Q are equivalent. Thus, we calculate the number of states after merging the equivalent states.

In order to show the equivalent states, let us recall the construction for B' and D_2. We notice that, in the construction of B', states s_2' and s_2 reach the same state on any letter in Σ. This is the reason for having D_2 in the transition rules. Moreover, a state of Q contains s_2' only when it contains a final state of A. Therefore, we can formally show that a pair of two states in Q, denoted by $\{q_f, s_2', s_2\} \cup T$ and $\{q_f, s_2'\} \cup T$ such that q_f is a final state of A and T either is the empty set or consists of some states of B, are equivalent as follows. For a letter $a \in \Sigma$ and a string $w \in \Sigma^*$,

$$\delta(\{q_f, s_2', s_2\} \cup T, aw) = \delta(\{q_f, s_2'\} \cup T, aw) = \delta(\delta(\{q_f, s_2'\} \cup T, a), w).$$

Note that the equivalent states are only in the set $F_1 \times \{s_2'\} \times \{S \mid S \subseteq (Q_2 - F_0)\}$, and we can further partition this set into two sets as follows:

$$F_1 \times \{s_2'\} \times (\{s_2\} \cup \{S' \mid S' \subseteq (Q_2 - F_0 - \{s_2\})\}) \cup$$
$$F_1 \times \{s_2'\} \times \{S' \mid S' \subseteq (Q_2 - F_0 - \{s_2\})\}.$$

It is easy to see that, for each state in the former set, there exists one and only one equivalent state in the latter set, and vice versa. Thus, the number of equivalent pairs is $k_1 2^{n-k_2-1}$.

Finally, we calculate the number of inequivalent states in Q. Notice that there are m elements in R. There are 2^{n-k_2} elements in the first term of P, and $(2^{k_2}-1)2^{n-k_2-1}$ elements in the second term of P. Therefore, the size of Q is $|Q| = m(2^{n-1} + 2^{n-k_2-1})$. Then after removing one state in each equivalent pair, we obtain the following upper bound

$$m(2^{n-1} + 2^{n-k_2-1}) - k_1 2^{n-k_2-1}.$$

q.e.d.

Next, we give examples to show that this upper bound can be attained.

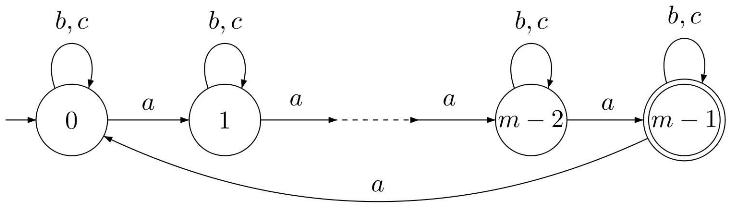

Figure 5.7: Witness DFA A for Theorem 5.6

Theorem 5.6 *For integers $m \geq 2$ and $n \geq 2$, there exists a DFA A of m states and a DFA of n states such that any DFA that accepts $L(A)L(B)^*$ has at least $m\frac{3}{4}2^n - 2^{n-2}$ states.*

Proof: We first give an example of two DFAs A and B of sizes $m \geq 2$ and $n = 2$, respectively, and we show that the number of states of a DFA that accepts $L(A)L(B)^*$ attains the upper bound given in Theorem 5.5. We use a three-letter alphabet $\Sigma = \{a, b, c\}$.

Define $A = (Q_1, \Sigma, \delta_1, q_0, \{q_{m-1}\})$, as in Figure 5.7, where $Q_1 = \{q_0, q_1, \ldots, q_{m-1}\}$, and the transitions are given as follows:

- $\delta_1(q_i, a) = q_{i+1}, i \in \{0, \ldots, m-2\}$, $\delta_1(q_{m-1}, a) = q_0$,
- $\delta_1(q_i, b) = q_{i+1}, i \in \{0, \ldots, m-3\}$, $\delta_1(q_{m-2}, b) = q_0$, $\delta_1(q_{m-1}, b) = q_{m-2}$,
- $\delta_1(q_i, c) = q_{i+1}, i \in \{0, \ldots, m-3\}$, $\delta_1(q_{m-2}, c) = q_0$, $\delta_1(q_{m-1}, c) = q_{m-1}$.

Define $B = (Q_2, \Sigma, \delta_2, 0, \{1\})$, where $Q_2 = \{0, 1\}$, and the transitions are given as follows

$$\delta_2(0, a) = 1, \quad \delta_2(0, b) = 0, \quad \delta_2(0, c) = 0,$$

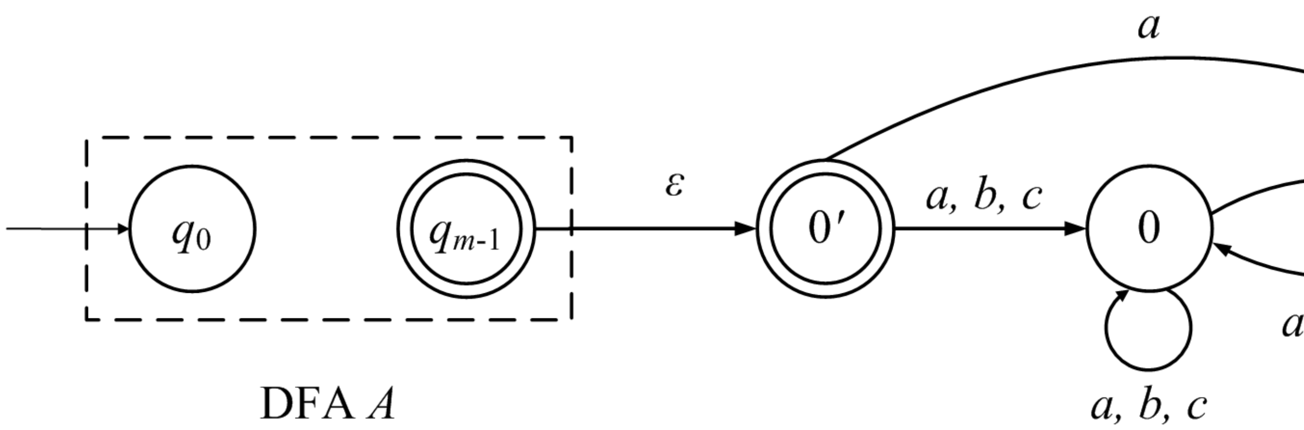

Figure 5.8: NFA for $L(A)L(B)^*$

Following the construction described in the proof of Theorem 5.5, we construct the DFA $C = (Q_3, \Sigma, \delta_3, s_3, F_3)$ that accepts $L(A)L(B)^*$. Note that set P only contains three elements $P = \{\emptyset, \{0\}, \{0, 1\}\}$. To prove that C attains the upper bound, it is sufficient to show that 1) all the states in Q_3 are reachable from s_3, 2) after merging the equivalent states $\{q_{m-1}, 0'\}$ and $\{q_{m-1}, 0', 0\}$, the remaining states are pairwise inequivalent.

We first consider the reachability of all the states. It is clear that the state $\{q_i\} \cup \{\emptyset\}$, for $i \in \{1, \ldots, m-2\}$, and the state $\{q_{m-1}, 0'\} \cup \{\emptyset\}$ are reachable from s_3 by reading the strings a^i and a^{m-1}, respectively. Then on letters b and c, we can reach states $\{q_{m-2}, 0\}$ and $\{q_{m-1}, 0', 0\}$, respectively, from the state $\{q_{m-1}, 0'\}$. Moreover, the state $\{q_i, 0\}$, $i \in \{0, \ldots, m-3\}$, can be reached from the state $\{q_{m-2}, 0\}$ by reading the string b^{i+1}. Lastly, the state $\{q_i, 0, 1\}$, $i \in \{0, \ldots, m-2\}$, and the state $\{q_{m-1}, 0', 0, 1\}$, are reachable from $\{q_{m-1}, 0'\}$ on inputs a^{i+1} and a^m, respectively.

Since states $\{q_{m-1}, 0'\}$ and $\{q_{m-1}, 0', 0\}$ are equivalent, we remove the state $\{q_{m-1}, 0', 0\}$ from Q_3, and show that the rest of the states are pairwise inequivalent. Let $\{q_i\} \cup G$ and $\{q_j\} \cup H$ be two different states in Q_3 with $0 \leq i \leq j \leq m-1$. There are three cases:

1. $i < j$. Then the string $a^{m-1-i}c$ is accepted by the DFA C starting from the state $\{q_i\} \cup G$, but it is not accepted starting from the state $\{q_j\} \cup H$. Note that after reading $a^{m-1-i}c$, the state $\{q_i\} \cup G$ reaches a state that contains states q_{m-1} and $0'$. In contrast, the state reached by $\{q_i\} \cup H$ on the same input does not contain these states. Moreover, the resulting states cannot contain the state 1, since on letter c, C remains in the state 0 from the state 0 and goes to the state 0 from the state 1.

2. $i = j \neq m-1$. Since $P = \{\emptyset, \{0\}, \{0, 1\}\}$ consists of only three elements, we consider them individually. It is obvious that the state $\{q_i, 0, 1\}$ is not equivalent to either $\{q_i\}$ or $\{q_i, 0\}$, since it is a final state but the latter two are not. States $\{q_i\}$ and $\{q_i, 0\}$ are inequivalent, since via the string ab we can reach a final state from the state $\{q_i, 0\}$ but not from the state $\{q_i\}$.

3. $i = j = m-1$. There are only two states $\{q_{m-1}, 0'\}$ and $\{q_{m-1}, 0', 0, 1\}$. They are inequivalent, because after reading the letter b, the state $\{q_{m-1}, 0', 0, 1\}$ leads to a final state of C but $\{q_{m-1}, 0'\}$ does not.

In the rest of the proof, we consider more general cases when the first DFA is of size $m \geq 2$ and the second DFA is of size $n \geq 3$. We again use the same DFA A, and give an example of the DFA D such that the number of states of a DFA that accepts $L(A)L(D)^*$ attains the upper bound. We use the same alphabet $\Sigma = \{a, b, c\}$.

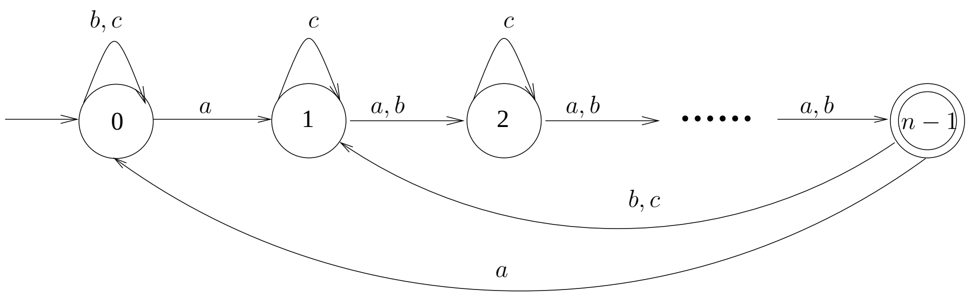

Figure 5.9: Witness DFA D for Theorem 5.6

Define $D = (Q_4, \Sigma, \delta_4, 0, \{n-1\})$, as shown in Figure 5.9, where $Q_4 = \{0, 1, \ldots, n-1\}$, and the transitions are given as follows:

- $\delta_4(i, a) = i + 1, i \in \{0, \ldots, n-2\}$, $\delta_4(n-1, a) = 0$,
- $\delta_4(0, b) = 0$, $\delta_4(i, b) = i + 1, i \in \{1, \ldots, n-2\}$, $\delta_4(n-1, b) = 1$,
- $\delta_4(i, c) = i, i \in \{0, \ldots, n-2\}$, $\delta_4(n-1, c) = 1$.

Let $E = (Q_5, \Sigma, \delta_5, s_5, F_5)$ be the DFA that accepts the language $L(A)L(D)^*$ constructed from A and D exactly as described in the proof of the previous theorem. Then we will show that (1) all the states in Q_5 are reachable from the initial state, and (2) after merging the states that are shown to be equivalent in the previous theorem, all the remaining states are pairwise inequivalent.

We first consider (1). Recall that every state in Q_5 consists of exactly one state of Q_1 and the states of an element in P defined from the states of D as in the previous theorem. Moreover, if a state of Q_5 contains a final state of A, then this state also contains $0'$. Thus, we denote each state in Q_5 as $Q'_i \cup S$, where $Q'_i = \{q_i\}$ for $i \in \{0, \ldots, m-2\}$, $Q'_{m-1} = \{q_{m-1}, 0'\}$, and $S \in P$. States $Q'_1 \cup \{\emptyset\}, \ldots, Q'_{m-1} \cup \{\emptyset\}$ are reachable since $Q'_i \cup \{\emptyset\} = \delta_5(Q'_0 \cup \{\emptyset\}, a^i)$, for $i \in \{1, 2, \ldots, m-1\}$. Then we prove that the rest of the states are reachable by induction on the size of S.

Basis: We show that, for any $i \in \{0, \ldots, m-1\}$, the state $Q'_i \cup S$ such that S contains only one state of B is reachable. We first consider two special cases when $S = \{0\}$ and $S = \{1\}$.

For the case when $S = \{0\}$, since $Q'_{m-1} \cup \{\emptyset\}$ is reachable, we have $Q'_{m-1} \cup \{0\} = \delta_5(Q'_{m-1} \cup \{\emptyset\}, c)$. Then from the state $Q'_{m-1} \cup \{0\}$, by reading the letter b, we can reach the state $Q'_{m-2} \cup \{0\}$. Furthermore, we can reach the other states where $S = \{0\}$ as follows:

$$Q'_i \cup \{0\} = \delta_5(Q'_{m-2} \cup \{0\}, c^{i+1}), \text{ for } i \in \{0, \ldots, m-3\}.$$

For the case when $S = \{1\}$, we can reach the state $Q'_i \cup \{1\}$ for $i \in \{1, \ldots, m-2\}$ from states $Q'_{i-1} \cup \{0\}$ by reading the letter a. Moreover, the state $Q'_0 \cup \{1\}$ can be reached from the state $Q'_{m-1} \cup \{0\}$ by reading the letter a. Note that the state $Q'_{m-1} \cup \{1\}$ has not been considered, but we will consider it later.

Then we consider the state $Q'_i \cup \{j\}$ where $j \geq 2$, for $i \in \{0, \dots, m-2\}$. We can easily verify that they can be reached as follows:

$$Q'_i \cup \{j\} = \delta_5(Q'_l \cup \{1\}, b^{j-1}),$$

where

$$l = \begin{cases} i - (j-1) \bmod (m-1) + m - 1, & \text{if } i - (j-1) \bmod (m-1) < 0; \\ i - (j-1) \bmod (m-1), & \text{otherwise.} \end{cases}$$

So far, the only states that have not been considered are states $Q'_{m-1} \cup \{j\}, j \geq 1$. However, it is clear that they can be reached from $Q'_{m-2} \cup \{j-1\}$ by reading the letter a.

Induction: For $i \in \{0, \dots, m-1\}$, assume that all states $Q'_i \cup S$ such that $|S| < k$ are reachable. Then we consider states $Q'_i \cup S$ where $|S| = k$. Let $S = \{j_1, j_2, \dots, j_k\}$ such that $0 \leq j_1 < j_2 < \cdots < j_k < n-1$ if $n-1 \notin S$, $j_1 = n-1$ and $0 = j_2 < \cdots < j_k < n-1$ otherwise. There are four cases:

1. $j_1 = n-1$ and $j_2 = 0$. Then for $i \in \{1, \dots, m-1\}$,

$$Q'_i \cup S = \delta_5(Q'_{i-1} \cup S', a)$$

where $S' = \{n-2, j_3 - 1, \dots, j_k - 1\}$, which contains $k-1$ states.

For the reachability of the state $Q'_0 \cup S$, we consider the following two subcases. (1) if $j_3 = 1$, $Q'_0 \cup S$ can be reached from $Q'_{m-1} \cup \{n-2, 0, j_4 - 1, \dots, j_k - 1\}$ by reading the letter a, (2) otherwise, it can be reached from $Q'_{m-2} \cup \{n-2, j_3 - 1, \dots, j_k - 1\}$ by reading the letter b. Note that in both of the two subcases, the state $Q'_0 \cup S$ is reached from a state where the size of S is $k-1$ as well.

2. $j_1 = 0$ and $j_2 = 1$. Then $Q'_0 \cup S = \delta_5(Q'_{m-1} \cup S', a)$, and for $i \in \{1, \dots, m-1\}$, $Q'_i \cup S = \delta_5(Q'_{i-1} \cup S', a)$, where $S' = \{n-1, 0, j_3 - 1, \dots, j_k - 1\}$. The state $Q'_i \cup S'$, $i \in \{0, \dots, m-1\}$, is considered in Case 1.

3. $j_1 = 0$ and $j_2 = 1 + t$, $t > 0$. Then for $i \in \{0, \dots, m-2\}$,

$$Q'_i \cup S = \delta_5(Q'_l \cup S', b^t)$$

where

$$l = \begin{cases} i - t \bmod (m-1) + m - 1, & \text{if } i - t \bmod (m-1) < 0; \\ i - t \bmod (m-1), & \text{otherwise,} \end{cases}$$

and $S' = \{0, 1, j_3 - t, \ldots, j_k - t\}$, which is considered in Case 2.

For the state $Q'_{m-1} \cup S$, we can verify that it is reachable from the state $Q'_{m-1} \cup S'$ by reading the letter c, where $S' = \{j_2, j_3, \ldots, j_k\}$ and $|S'| = k - 1$.

4. $j_1 = t > 0$. We first consider the case when $t = 1$. It is clear that the state $Q'_0 \cup S$ and the state $Q'_i \cup S$, $i \in \{1, \ldots, m-1\}$, can be reached from states $Q'_{m-1} \cup S'$ and $Q'_{i-1} \cup S'$, respectively, by reading the letter a, where $S' = \{0, j_2 - 1, \ldots, j_k - 1\}$, which is considered in either Case 2 or Case 3.

Then we consider the cases when $t > 1$. If $i \in \{0, \ldots, m-2\}$, the state $Q'_i \cup S$ is reachable as follows:

$$Q'_i \cup S = \delta_5(Q'_l \cup \{1, j_2 - t + 1, \ldots, j_k - t + 1\}, b^{t-1}),$$

where

$$l = \begin{cases} i - (t-1) \bmod (m-1) + m - 1, & \text{if } i - (t-1) \bmod (m-1) < 0; \\ i - (t-1) \bmod (m-1) & \text{otherwise,} \end{cases}$$

For the remaining states, the state $Q'_{m-1} \cup S$ can be reached from the state $Q'_{m-2} \cup \{1, j_2 - 1, \ldots, j_k - 1\}$ by reading the letter a.

Now we show that, after merging the states that are proven to be equivalent, the rest of the states are pairwise inequivalent. Let $\{q_i\} \cup G$ and $\{q_j\} \cup H$ be two different states in Q_5, where $q_i, q_j \in Q_1$, with $0 \leq i \leq j \leq m-1$. Then we consider the following three cases:

1. $i < j$. The string $a^{m-1-i}c$ is accepted by the DFA E starting from the state $\{q_i\} \cup G$, but it is not accepted starting from the state $\{q_j\} \cup H$. The reason is similar to that for the DFA C, but, on the letter c, E remains in the same state for any non-final state, and goes to the state 1 from the state $n-1$.

2. $i = j \neq m - 1$. Without loss of generality, there exists a state k of D such that $k \in G$ and $k \notin H$. We first consider a special case when $H \subset G$ and $G - H = \{0\}$. The only difference between G and H is that G contains one more state 0 than H. In such a case, we can verify that the string ab^{n-2} is accepted by the DFA C starting from the state $\{q_i\} \cup G$, but it is not accepted starting from the state $\{q_j\} \cup H$. In other cases, we can assume that $k > 0$. Then the string b^{n-1-k} is accepted by the DFA E starting from the state $\{q_i\} \cup G$, but it is not accepted starting from the state $\{q_j\} \cup H$.

3. $i = j = m - 1$. Recall from the proof of Theorem 5.5 that we can

partition the subset $\{q_{m-1}\} \times \{0'\} \times \{S \mid S \subseteq (Q_4 - F_0)\}$ of Q_5 into

$$\{q_{m-1}\} \times \{0'\} \times (\{0\} \cup \{S' \mid S' \subseteq (Q_4 - F_0 - \{0\})\}) \cup$$
$$\{q_{m-1}\} \times \{0'\} \times \{S' \mid S' \subseteq (Q_4 - F_0 - \{0\})\}.$$

Moreover, for each state in the former set, there exists one and only one equivalent state in the latter set, and vice versa. Thus, we remove all the states in the former set from Q_5. Then, without loss of generality, there exists a state k of D such that $k \neq 0'$, $k \neq 0$, $k \in G$, and $k \notin H$. We can verify that the string b^{2n-2-k} is accepted starting from the state $\{q_i\} \cup G$, but it is not accepted starting from the state $\{q_j\} \cup H$. *q.e.d.*

5.1.3 State Complexity of $L_1^R L_2$

In this subsection, we study the state complexity of $L_1^R L_2$ for an m-state DFA language L_1 and an n-state DFA language L_2. All the results in this subsection are from our paper [12].

We first show that the state complexity of $L_1^R L_2$ is upper bounded by $3 \cdot 2^{m+n-2}$ in general (Theorem 5.7). Then we prove that this upper bound can be attained when $m, n \geq 2$ (Theorems 5.8 and 5.9). 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 5.10). Finally, we show that the state complexity of $L_1^R L_2$ is $2^{m-1} + 1$ when $m \geq 2$ and $n = 1$ (Theorems 5.11, 5.12, 5.13 and Lemma 5.3).

Now, we start with a general upper bound on state complexity of $L_1^R L_2$ for integers $m, n \geq 1$.

Theorem 5.7 *For two integers $m, n \geq 1$, let L_1 and L_2 be two regular languages accepted by an m-state DFA and an n-state DFA, respectively. Then there exists a DFA of at most $3 \cdot 2^{m+n-2}$ 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, k_1 final 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 k_1 initial states. The transition function $\delta_{M'}(p, a) = q$ 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 the DFA $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ that accepts the language $L_1^R L_2$, where

$$Q_B = \{\langle i,j\rangle \mid i \in Q_A,\ j \subseteq Q_N\},$$

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

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

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

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 that don't satisfy this condition.

Thus, the number of states of the minimal DFA that accepts $L_1^R L_2$ is no more than

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

q.e.d.

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

Theorem 5.8 *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 that accepts $L(M)^R L(N)$ has at least $3 \cdot 2^{m+n-2}$ states.*

Proof: Let $M = (Q_M, \Sigma, \delta_M, 0, \{m-1\})$ be a DFA, as shown in Figure 5.10, where $Q_M = \{0, 1, \ldots, m-1\}$, $\Sigma = \{a, b, c, d\}$, and the transitions are given as follows:

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

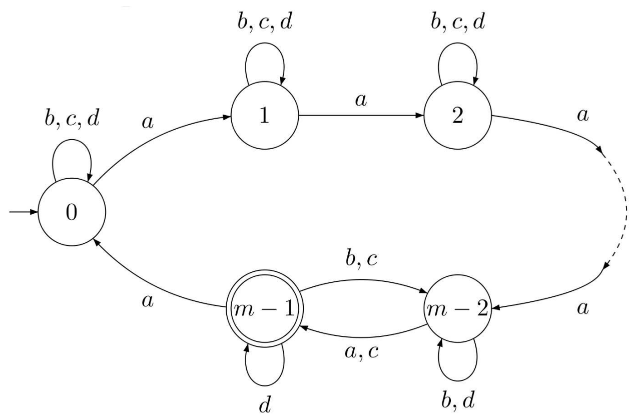

Figure 5.10: Witness DFA M of Theorem 5.8 showing that the upper bound in Theorem 5.7 is attainable when $m, n \geq 2$

- $\delta_M(i, d) = i$, $i = 0, \ldots, m-1$.

Let $N = (Q_N, \Sigma, \delta_N, 0, \{n-1\})$ be a DFA, as shown in Figure 5.11, where $Q_N = \{0, 1, \ldots, n-1\}$, $\Sigma = \{a, b, c, d\}$, and the transitions are given as follows:

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

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

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

It is easy to see that A is a DFA that accepts $L(M)^R$. We prove that A is minimal before using it.

(I) We first show that every state $I \in Q_A$ is reachable from $\{m-1\}$. There are three cases.

1. $|I| = 0$. $|I| = 0$ if and only if $I = \emptyset$. Then $\delta_A(\{m-1\}, b) = I = \emptyset$.

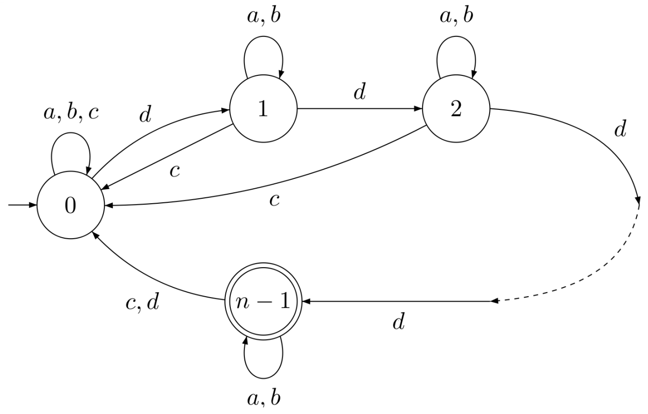

Figure 5.11: Witness DFA N of Theorem 5.8 showing that the upper bound in Theorem 5.7 is attainable when $m, n \geq 2$

2. $|I| = 1$. Let $I = \{i\}$, $0 \leq i \leq m-1$. Then $\delta_A(\{m-1\}, a^{m-1-i}) = I$.
3. $2 \leq |I| \leq m$. Let $I = \{i_1, i_2, \ldots, i_k\}$, $0 \leq i_1 < i_2 < \cdots < i_k \leq m-1$, $2 \leq k \leq m$. Then $\delta_A(\{m-1\}, w) = I$, 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_A are distinguishable.

Without loss of generality, we may assume that $|I| \geq |J|$. Let $x \in I - J$. Then the string a^x distinguishes these two states because

$$\begin{aligned} \delta_A(I, a^x) &\in F_A, \\ \delta_A(J, a^x) &\notin F_A. \end{aligned}$$

From (I) and (II), A is a minimal DFA with 2^m states that accepts $L(M)^R$. Now let $B = (Q_B, \Sigma, \delta_B, s_B, F_A\}$ be another DFA, where

$$\begin{aligned} Q_B &= \{\langle p, q\rangle \mid p \in Q_A - F_A,\, q \subseteq Q_N\} \\ &\qquad \cup \{\langle p', q'\rangle \mid p' \in F_A,\, q' \subseteq Q_N,\, 0 \in q'\}, \\ \Sigma &= \{a, b, c, d\}, \\ s_B &= \langle \{m-1\}, \emptyset\rangle, \\ F_B &= \{\langle p, q\rangle \mid n-1 \in q,\, \langle p, q\rangle \in Q_B\}, \end{aligned}$$

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 previous proof, all the states in B that starts 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 a sink state of B. $\delta_B(\langle \{m-1\}, \emptyset\rangle, b) = \langle p, q\rangle$.
2. $p \neq \emptyset$, $q = \emptyset$. Let $p = \{p_1, p_2, \ldots, p_k\}$, $1 \leq p_1 < p_2 < \cdots < p_k \leq m-1$, $1 \leq k \leq m-1$. 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

   $$w = ab(ac)^{p_2-p_1-1} ab(ac)^{p_3-p_2-1} \cdots ab(ac)^{p_k-p_{k-1}-1} a^{m-1-p_k}.$$

   Note that $w = a^{m-1-p_1}$ when $k = 1$.
3. $p = \emptyset$, $q \neq \emptyset$. In this case, let $q = \{q_1, q_2, \ldots, q_l\}$, $0 \leq q_1 < q_2 < \cdots < q_l \leq n-1$, $1 \leq l \leq n$. Then $\delta_B(\langle \{m-1\}, \emptyset\rangle, x) = \langle p, q\rangle$, 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 $p = \{p_1, p_2, \ldots, p_k\}$, $1 \leq p_1 < p_2 < \cdots < p_k \leq m-1$, $1 \leq k \leq m-1$ and $q = \{q_1, q_2, \ldots, q_l\}$, $0 \leq q_1 < q_2 < \cdots < q_l \leq n-1$, $1 \leq l \leq n$. 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 $p = \{p_1, p_2, \ldots, p_k\}$, $0 = p_1 < p_2 < \cdots < p_k < m-1$, $1 \leq k \leq m-1$ and $q = \{q_1, q_2, \ldots, q_l\}$, $0 = q_1 < q_2 < \cdots < q_l \leq n-1$, $1 \leq l \leq n$. 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 $p = \{p_1, p_2, \ldots, p_k\}$, $0 = p_1 < p_2 < \cdots < p_k = m-1$, $2 \leq k \leq m$ and $q = \{q_1, q_2, \ldots, q_l\}$, $0 = q_1 < q_2 < \cdots < q_l \leq n-1$, $1 \leq l \leq n$. 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. $q_1 \neq q_2$. Without loss of generality, we may assume that $|q_1| \geq |q_2|$. Let $x \in q_1 - q_2$. The string d^{n-1-x} distinguishes them because

$$\begin{aligned} \delta_B(\langle p_1, q_1\rangle, d^{n-1-x}) &\in F_B, \\ \delta_B(\langle p_2, q_2\rangle, d^{n-1-x}) &\notin F_B. \end{aligned}$$

2. $p_1 \neq p_2$, $q_1 = q_2$. Without loss of generality, we assume that $|p_1| \geq |p_2|$. Let $y \in p_1 - p_2$. Then there always exists a string $a^y c^2 d^n$ such that

$$\begin{aligned} \delta_B(\langle p_1, q_1\rangle, a^y c^2 d^n) &\in F_B, \\ \delta_B(\langle p_2, q_2\rangle, a^y c^2 d^n) &\notin F_B. \end{aligned}$$

Since all the states in B are reachable and pairwise distinguishable, the DFA B is minimal. Thus, any DFA that accepts $L(M)^R L(N)$ has at least $3 \cdot 2^{m+n-2}$ states. *q.e.d.*

This result 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 5.7 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 5.9 *For 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 subsection, 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 [111] that 2^{n-1} states are both sufficient and necessary in the worst case for a DFA to accept the catenation of a one-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 one 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 the following theorem concerning the state complexity of $L_1^R L_2$ for $m = 1$ and $n \geq 1$.

Theorem 5.10 *Let L_1 be a one-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 5.11 *For any integer $m \geq 2$, let L_1 and L_2 be two regular languages accepted by an m-state DFA and a one-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 one 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 when $L_2 = L(N) = \Sigma^*$.

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 5.7. $\delta_{M'}(p, a) = q$ 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 the subset construction on 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 $L(B) = L_1^R \Sigma^*$. *q.e.d.*

This theorem shows an upper bound for the state complexity of $L_1^R L_2$ for $m \geq 2$ and $n = 1$. Next we prove that this upper bound is attainable.

Lemma 5.3 *Given an integer $m = 2$ or 3, there exists an m-state DFA M and a one-state DFA N such that any DFA that accepts $L(M)^R L(N)$ has 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 follows:

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

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

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

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 given as follows:

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

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

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

q.e.d.

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

Theorem 5.12 *Given an integer $m \geq 4$, there exists a DFA M of m states and a DFA N of one state such that any DFA that accepts $L(M)^R L(N)$ has at least $2^{m-1} + 1$ states.*

Proof: Let $M = (Q_M, \Sigma, \delta_M, 0, \{m-1\})$ be a DFA, as shown in Figure 5.12, where $Q_M = \{0, 1, \ldots, m-1\}$, $m \geq 4$, $\Sigma = \{a, b, c, d\}$, and the transitions are given as follows:

- $\delta_M(i, a) = i + 1 \bmod m$, $i = 0, \ldots, m-1$;

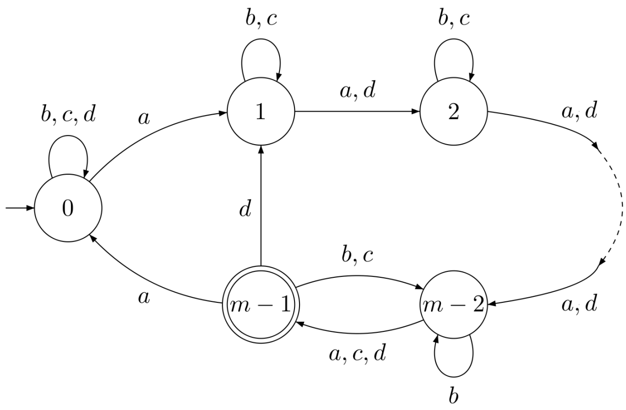

Figure 5.12: Witness DFA M of Theorem 5.12 showing that the upper bound in Theorem 5.11 is attainable when $m \geq 4$ and $n = 1$

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

Let N be the DFA that accepts Σ^*. Then $L(M)^R L(N) = L(M)^R \Sigma^*$. Now we construct the DFA $A = (Q_A, \Sigma, \delta_A, \{m-1\}, F_A)$ similar to the proof of Theorem 5.8, where $Q_A = \{q \mid q \subseteq Q_M\}$, $\Sigma = \{a, b, c, d\}$, $F_A = \{q \mid 0 \in q,\, q \in Q_A\}$, and the transitions are defined as follows:

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

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 the DFA M in the proof of Theorem 5.8, we can say that A is minimal and it has 2^m states, among which 2^{m-1} states are final.

Let $B = (Q_B, \Sigma, \delta_B, s_B, \{f_B\})$ be a DFA 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 $e \in \Sigma$ 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 5.11. 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 5.8. We consider the following four cases:

1. $I = \emptyset$. Then $\delta_A(\{m-1\}, b) = I = \emptyset$.
2. $I = f_B$. Then $\delta_A(\{m-1\}, a^{m-1}) = I = f_B$.
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. $2 \leq |I| \leq m$. Assume that $I = \{i_1, i_2, \ldots, i_k\}$, $1 \leq i_1 < i_2 < \cdots < i_k \leq m-1$, $2 \leq k \leq m$. $\delta_A(\{m-1\}, w) = I$, 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.

Without loss of generality, we may assume that $|I| \geq |J|$. Let $x \in I - J$. x is always greater than 0 because all the states which include 0 have been merged into f_B. Then the string $d^{x-1}a$ distinguishes these two states because

$$\begin{aligned} \delta_B(I, d^{x-1}a) &= f_B, \\ \delta_B(J, d^{x-1}a) &\neq f_B. \end{aligned}$$

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

After summarizing Theorem 5.11, Theorem 5.12 and Lemma 5.3, we obtain the state complexity of the combined operation $L_1^R L_2$ for $m \geq 2$ and $n = 1$.

Theorem 5.13 *For any integer $m \geq 2$, let L_1 be an m-state DFA language and L_2 be a one-state DFA language. Then $2^{m-1} + 1$ states are both sufficient and necessary in the worst case for a DFA to accept $L_1^R L_2$.*

5.1.4 State Complexity of $L_1 L_2^R$

In this subsection, we study the state complexity of $L_1 L_2^R$ for regular languages L_1 and L_2. All the results in this subsection are from our paper [10]. We will first look at an upper bound on this state complexity.

Theorem 5.14 *For two integers $m, n \geq 1$, let L_1 and L_2 be two regular languages accepted by an m-state DFA with k_1 final states and an n-state DFA with k_2 final states, respectively. Then there exists a DFA of at most $m2^n - k_1 2^{n-k_2}(2^{k_2} - 1) - m + 1$ states that accepts $L_1 L_2^R$.*

Proof: Let $M = (Q_M, \Sigma, \delta_M, s_M, F_M)$ be a DFA of m states, k_1 final states and $L_1 = L(M)$. Let $N = (Q_N, \Sigma, \delta_N, s_N, F_N)$ be another DFA of n states, k_2 final states and $L_2 = L(N)$. Let $N' = (Q_N, \Sigma, \delta_{N'}, F_N, \{s_N\})$ be an NFA with k_2 initial states. $\delta_{N'}(p, a) = q$ if $\delta_N(q, a) = p$ where $a \in \Sigma$ and $p, q \in Q_N$. Clearly,

$$L(N') = L(N)^R = L_2^R.$$

After performing the subset construction on N', we can get an equivalent, 2^n-state DFA $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ such that $L(A) = L_2^R$. Note that A may not be minimal and since A has 2^n states, one of its final states must be Q_N. Now we construct the DFA $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ that accepts the language $L_1 L_2^R$, where

$$Q_B = \{\langle i, j \rangle \mid i \in Q_M, \ j \in Q_A\},$$

$$F_B = \{\langle i, j \rangle \in Q_B \mid j \in F_A\},$$

$$s_B = \begin{cases} \langle s_M, \emptyset \rangle, & \text{if } s_M \notin F_M; \\ \langle s_M, F_N \rangle, & \text{otherwise,} \end{cases}$$

$$\delta_B(\langle i, j \rangle, a) = \begin{cases} \langle i', j' \rangle, & \text{if } \delta_M(i, a) = i', \ \delta_A(j, a) = j', \ a \in \Sigma, \ i' \notin F_M; \\ \langle i', j' \cup F_N \rangle, & \text{if } \delta_M(i, a) = i', \ \delta_A(j, a) = j', \ a \in \Sigma, \ i' \in F_M. \end{cases}$$

It is easy to see that $\delta_B(\langle i, Q_N\rangle, a) \in F_B$ for any $i \in Q_M$ and $a \in \Sigma$. This means all the states (two-tuples) ending with Q_N are equivalent. There are m such states.

On the other hand, since NFA N' has k_2 initial states, the states in B starting with $i \in F_M$ must end with j such that $F_N \subseteq j$. There are in total $k_1 2^{n-k_2}(2^{k_2}-1)$ states which don't satisfy this condition.

Thus, the number of states of the minimal DFA that accepts $L_1 L_2^R$ is no more than

$$m2^n - k_1 2^{n-k_2}(2^{k_2}-1) - m + 1.$$

q.e.d.

This result gives an upper bound for the state complexity of $L_1 L_2^R$. Next we show that this bound is attainable.

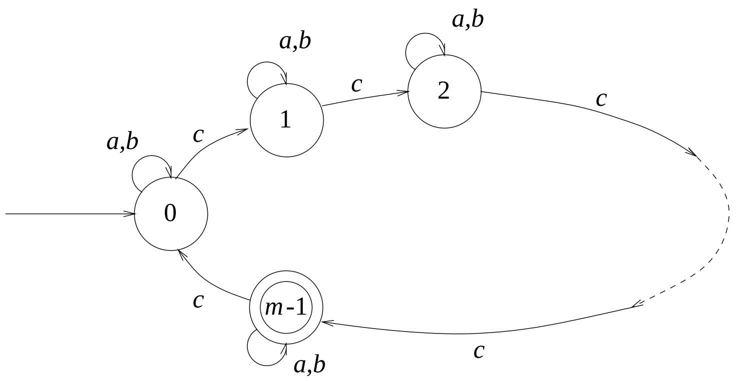

Figure 5.13: Witness DFA M of Theorem 5.15 showing that the upper bound in Theorem 5.14 is attainable when $m \geq 2$ and $n \geq 2$

Theorem 5.15 *Given two integers $m \geq 2$, $n \geq 2$, there exists a DFA M of m states and a DFA N of n states such that any DFA that accepts $L(M)L(N)^R$ has at least $m2^n - 2^{n-1} - m + 1$ states.*

Proof: Let $M = (Q_M, \Sigma, \delta_M, 0, \{m-1\})$ be a DFA, as shown in Figure 5.13, where $Q_M = \{0, 1, \ldots, m-1\}$, $\Sigma = \{a, b, c\}$, and the transitions are given by

- $\delta_M(i, x) = i$, $i = 0, \ldots, m-1, x \in \{a, b\}$;
- $\delta_M(i, c) = i + 1 \bmod m$, $i = 0, \ldots, m-1$.

Let $N = (Q_N, \Sigma, \delta_N, 0, \{0\})$ be a DFA, where $Q_N = \{0, 1, \ldots, n-1\}$, $\Sigma = \{a, b, c\}$, and the transitions are given by

- $\delta_N(0, a) = n-1$, $\delta_N(i, a) = i-1$, $i = 1, \ldots, n-1$;
- $\delta_N(0, b) = 1$, $\delta_N(i, b) = i$, $i = 1, \ldots, n-1$;
- $\delta_N(0, c) = 1$, $\delta_N(1, c) = 0$, $\delta_N(j, c) = j$, $j = 2, \ldots, n-1$, if $n \geq 3$.

N is the same as the witness DFA for the state complexity of reversal operation on regular languages. The transition diagram of N is shown in Figure 3.5.

Now we construct the DFA $A = (Q_A, \Sigma, \delta_A, \{0\}, F_A)$, where $Q_A = \{q \mid q \subseteq Q_N\}$, $\Sigma = \{a, b, c\}$, $F_A = \{q \mid 0 \in q,\ q \in Q_A\}$, and the transitions are defined as:

$$\delta_A(p, e) = \{j \mid \delta_N(j, e) = i,\ i \in p\},\ p \in Q_A,\ e \in \Sigma.$$

It has been shown in [111] that A is a minimal DFA that accepts $L(N)^R$. Let $B = (Q_B, \Sigma, \delta_B, s_B, F_A)$ be another DFA, where

$$\begin{aligned}
Q_B &= \{\langle p, q\rangle \mid p \in Q_M - \{m-1\},\ q \in Q_A - \{Q_N\}\} \cup \{\langle 0, Q_N\rangle\} \\
&\qquad \cup \{\langle m-1, q\rangle \mid q \in Q_A - \{Q_N\},\ \{0\} \in q\}, \\
\Sigma &= \{a, b, c\}, \\
s_B &= \langle 0, \emptyset\rangle, \\
F_B &= \{\langle p, q\rangle \mid q \in F_A,\ \langle p, q\rangle \in Q_B\},
\end{aligned}$$

and for each state $\langle p, q\rangle \in Q_B$ and each letter $e \in \Sigma$,

$$\delta_B(\langle p, q\rangle, e) = \begin{cases}
\langle p', q'\rangle, & \text{if } \delta_M(p, e) = p' \neq m-1,\ \delta_A(q, e) = q' \neq Q_N; \\
\langle p', q'\rangle, & \text{if } \delta_M(p, e) = p' = m-1, \\
 & \qquad \delta_A(q, e) = r',\ q' = r' \cup \{0\},\ q' \neq Q_N; \\
\langle 0, Q_N\rangle, & \text{if } \delta_M(p, e) = m-1,\ \delta_A(q, e) = r',\ r' \cup \{0\} = Q_N; \\
\langle 0, Q_N\rangle, & \text{if } \delta_M(p, e) \neq m-1,\ \delta_A(q, e) = Q_N.
\end{cases}$$

As we mentioned in the previous proof, all the states (two-tuples) ending with Q_N are equivalent. So, we replace them with one state: $\langle 0, Q_N\rangle$. According to the definition of B, all the states in B that starts with $m-1$ must end with $j \in Q_A$ such that $0 \in j$. It is easy to see that B accepts the language $L(M)L(N)^R$. It has $m2^n - 2^{n-1} - m + 1$ states. Now we show that B is a minimal DFA.

(I) We first show that every state $\langle i, j\rangle \in Q_B$ is reachable by induction on the size of j. Let $k = |j|$ and $k \le n-1$. Note that the state $\langle 0, Q_N\rangle$ is reachable from the state $\langle 0, \emptyset\rangle$ via the string $c^m b(ab)^{n-2}$.

When $k = 0$, i is less than $m-1$ according to the definition of B. Then there always exists a string $w = c^i$ such that $\delta_B(\langle 0, \emptyset\rangle, w) = \langle i, \emptyset\rangle$.

Basis ($k = 1$): The state $\langle m-1, \{0\}\rangle$ can be reached from the state $\langle m-2, \emptyset\rangle$ on c. The State $\langle 0, \{0\}\rangle$ can be reached from the state $\langle m-1, \{0\}\rangle$ on ca^{n-1}. Then for $i \in \{1, \dots, m-2\}$, the state $\langle i, \{0\}\rangle$ is reachable from the state $\langle i-1, \{0\}\rangle$ on ca^{n-1}. Moreover, for $i \in \{0, \dots, m-2\}$, the state $\langle i, j\rangle$ is reachable from the state $\langle i, \{0\}\rangle$ on a^j.

Induction: Assume that all states $\langle i, j\rangle$ such that $|j| < k$ are reachable. Then we consider the states $\langle i, j\rangle$ where $|j| = k$. Let $j = \{j_1, j_2, \dots, j_k\}$ such that $0 \le j_1 < j_2 < \cdots < j_k \le n-1$. We consider the following four cases:

1. $j_1 = 0$ and $j_2 = 1$. The state $\langle m-1, \{0, 1, j_3, \dots, j_k\}\rangle$ is reachable from the state $\langle m-2, \{0, j_3, \dots, j_k\}\rangle$ on c. Then for $i \in \{0, \dots, m-2\}$, the state $\langle i, j\rangle$ can be reached from the state $\langle m-1, \{0, 1, j_3, \dots, j_k\}\rangle$ on c^{i+1}.

2. $i = 0$, $j_1 = 0$, and $j_2 > 1$. The state $\langle 0, j\rangle$ can be reached as follows:

$$\langle 0, \{j_1, j_2, \dots, j_k\}\rangle = \delta_B(\langle m-2, \{j_3 - j_2 + 1, \dots, j_k - j_2 + 1, n - j_2 + 1\}\rangle, c^2 a^{j_2 - 1}).$$

3. $i = 0$ and $j_1 > 0$. The state $\langle 0, j\rangle$ is reachable from the state $\langle 0, \{0, j_2 - j_1, \dots, j_k - j_1\}\rangle$ on a^{j_1}.

4. We consider the remaining states. For $i \in \{1, \dots, m-1\}$, the state $\langle i, j\rangle$ such that $j_1 = 0$ and $j_2 > 1$ can be reached from the state $\langle i-1, \{1, j_2, \dots, j_k\}\rangle$ on the letter c, and for $i \in \{1, \dots, m-2\}$, the state $\langle i, j\rangle$ such that $j_1 > 0$ is reachable from the state $\langle i, \{0, j_2 - j_1, \dots, j_k - j_1\}\rangle$ via the string a^{j_1}. Recall that we do not have states $\langle i, j\rangle$ such that $i = m-1$ and $j_1 > 0$.

(II) We then show that any two different states $\langle i_1, j_1\rangle$ and $\langle i_2, j_2\rangle$ in Q_B are distinguishable. Let us consider the following three cases:

1. $j_1 \ne j_2$. Without loss of generality, we may assume that $|j_1| \ge |j_2|$. Let $x \in j_1 - j_2$. We don't need to consider the case when $x = 0$, since the two states are clearly in different equivalence classes if $0 \in j_1 - j_2$. For $0 < x \le n-1$, there always exists a string t such that

$$\begin{aligned} \delta_B(\langle i_1, j_1\rangle, t) &\in F_B, \\ \delta_B(\langle i_2, j_2\rangle, t) &\notin F_B, \end{aligned}$$

where

$$t = \begin{cases} a^{n-x}, & \text{if } i_2 \neq m-1,\ j_1 \neq j_2; \\ a^{n-x-1}ca, & \text{if } i_2 = m-1,\ j_1 \neq j_2,\ n > 2; \\ c, & \text{if } i_2 = m-1,\ j_1 \neq j_2,\ n = 2. \end{cases}$$

Note that under the second condition, after reading the prefix a^{n-x-1} of t, the state $n-1$ cannot be in the second component of the resulting state since $x \notin j_2$.

Also note that when $n = 2$, $j_1, j_2 \in \{Q_N, \{0\}, \{1\}\}$, where $Q_N = \{0, 1\}$. Moreover, when $i_2 = m-1$, the state $\langle i_2, j_2 \rangle$ can only be $\langle m-1, \{0\} \rangle$. Due to the definition of B, we have that, for $s \geq 1$, $\langle s, Q_N \rangle \notin Q_B$. Thus, it is easy to see that $\langle i_1, j_1 \rangle$ is either $\langle i_1, \{1\} \rangle$ or $\langle 0, \{0, 1\} \rangle$. When $\langle i_1, j_1 \rangle = \langle i_1, \{1\} \rangle$, $0 \in j_1 - j_2$, so the two states are distinguishable. When $\langle i_1, j_1 \rangle = \langle 0, \{0, 1\} \rangle$, the letter c distinguishes them because

$$\begin{aligned} \delta_B(\langle 0, \{0, 1\} \rangle, c) &\in F_B, \\ \delta_B(\langle m-1, \{0\} \rangle, c) &\notin F_B. \end{aligned}$$

2. $j_1 = j_2 \neq Q_N$, $i_1 \neq i_2$. Without loss of generality, we may assume that $i_1 > i_2$. In this case, $i_2 \neq m-1$. Let $x \in Q_N - j_1$. There always exists a string $u = a^{n-x+1}bc^{m-1-i_1}$ such that

$$\begin{aligned} \delta_B(\langle i_1, j_1 \rangle, u) &\in F_B, \\ \delta_B(\langle i_2, j_2 \rangle, u) &\notin F_B. \end{aligned}$$

Let $\langle i_1, j_1' \rangle$ and $\langle i_2, j_1' \rangle$ be two states reached from states $\langle i_1, j_1 \rangle$ and $\langle i_2, j_2 \rangle$ on the prefix a^{n-x+1} of u, respectively. We notice that the state 1 of N cannot be in j_1'. Then after reading another letter b, we reach states $\langle i_1, j_1'' \rangle$ and $\langle i_2, j_1'' \rangle$, respectively. It is easy to see that states 0 and 1 of N are not in j_1''. Lastly, after reading the remaining string c^{m-1-i_1} from the state $\langle i_1, j_1'' \rangle$, the first component of the resulting state is the final state of the DFA M and therefore its second component contains the state 0 of the DFA N. In contrast, the second component of the resulting state reached from the state $\langle i_2, j_1'' \rangle$ on the same string cannot contain the state 0, and hence it is not a final state of B. Note that this includes the case when $j_1 = j_2 = \emptyset$, $i_1 \neq i_2$.

3. We don't need to consider the case when $j_1 = j_2 = Q_N$, because there is only one state in Q_B that ends with Q_N. It is $\langle 0, Q_N \rangle$.

Since all the states in B are reachable and pairwise distinguishable, the DFA B is minimal. Thus, any DFA that accepts $L(M)L(N)^R$ has at least $m2^n - 2^{n-1} - m + 1$ states. *q.e.d.*

This result gives a lower bound for the state complexity of $L(M)L(N)^R$ when $m, n \geq 2$. It coincides with the upper bound when $k_1 = 1$ and $k_2 = 1$. In the rest of this subsection, we consider the remaining cases when either $m = 1$ or $n = 1$. We first consider the case when $m = 1$ and $n \geq 3$. We have $L_1 = \emptyset$ or $L_1 = \Sigma^*$. When $L_1 = \emptyset$, for any L_2, a one-state DFA always accepts $L_1L_2^R$, since $L_1L_2^R = \emptyset$. The following theorem provides a lower bound for the latter case.

Theorem 5.16 *Given an integer $n \geq 3$, there exists a DFA M of one state and a DFA N of n states such that any DFA that accepts $L(M)L(N)^R$ has at least 2^{n-1} states.*

Proof: Let $M = (Q_M, \Sigma, \delta_M, 0, \{0\})$ be a DFA, where $Q_M = \{0\}$, $\Sigma = \{a, b\}$, and $\delta_M(0, e) = 0$ for any $e \in \Sigma$. Clearly, $L(M) = \Sigma^*$.

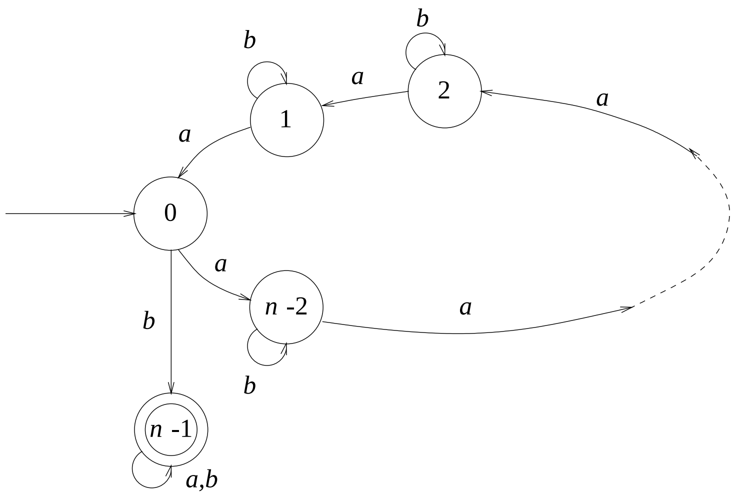

Figure 5.14: Witness DFA N showing that the upper bound in Theorem 5.14 is attainable when $m \geq 1$ and $n \geq 3$

Let $N = (Q_N, \Sigma, \delta_N, 0, \{n-1\})$ be a DFA, as shown in Figure 5.14, where $Q_N = \{0, 1, \ldots, n-1\}$, $\Sigma = \{a, b\}$, and the transitions are given by

- $\delta_N(0, a) = n - 2$, $\delta_N(i, a) = i - 1$, $i = 1, \ldots, n-2$, $\delta_N(n-1, a) = n - 1$

- $\delta_N(0, b) = n-1$, $\delta_N(j, b) = j$, $j = 1, \ldots, n-1$.

Now we design a 2^n-state DFA $A = (Q_A, \Sigma, \delta_A, \{n-1\}, F_A)$, where $Q_A = \{q \mid q \subseteq Q_N\}$, $\Sigma = \{a, b\}$, $F_A = \{q \mid 0 \in q, q \in Q_A\}$, and the transitions are defined as follows:

$$\delta_A(p, e) = \{j \mid \delta_N(j, e) = i, i \in p\}, p \in Q_A, e \in \Sigma.$$

It is easy to see that A is a DFA that accepts $L(N)^R$. Let $B = (Q_B, \Sigma, \delta_B, s_B, F_A\}$ be another DFA, where $\Sigma = \{a, b\}$,

$$\begin{aligned} Q_B &= \{\langle 0, q\rangle \mid q \in Q_A, n-1 \in q\}, \\ s_B &= \langle 0, \{n-1\}\rangle, \\ F_B &= \{\langle 0, q\rangle \mid q \in F_A, \langle 0, q\rangle \in Q_B\}, \end{aligned}$$

and for each state $\langle 0, q\rangle \in Q_B$ and each letter $e \in \Sigma$,

$$\delta_B(\langle 0, q\rangle, e) = \langle 0, q'\rangle \text{ if } \delta_A(q, e) = q'' \text{ and } q' = q'' \cup \{n-1\}.$$

Clearly, the DFA B accepts $L(M)L(N)^R$. Since $n-1 \in j$ for any state $\langle 0, j\rangle \in Q_B$, B has 2^{n-1} states in total. Now we show that B is a minimal DFA.

(I) We first show that every state $\langle 0, j\rangle \in Q_B$ is reachable. We omit the case when $|j| = 1$ because the only state in Q_B satisfying this condition is the initial state $\langle 0, \{n-1\}\rangle$. When $|j| > 1$, assume that $j = \{n-1, j_1, j_2, \ldots, j_k\}$ where $0 \le j_1 < j_2 < \cdots < j_k \le n-2$, $1 \le k \le n-1$. There always exists a string

$$w = ba^{j_k - j_{k-1}} ba^{j_{k-1} - j_{k-2}} \cdots ba^{j_2 - j_1} ba^{j_1}$$

such that $\delta_B(\langle 0, \{n-1\}\rangle, w) = \langle 0, j\rangle$.

(II) We then show that any two different states $\langle 0, j_1\rangle$ and $\langle 0, j_2\rangle$ in Q_B are distinguishable. Without loss of generality, we may assume that $|j_1| \ge |j_2|$. Then let $x \in j_1 - j_2$. Note that $x \ne n-1$ because $n-1$ has to be in both j_1 and j_2. We can always find the string $u = a^{n-1-x}$ such that

$$\delta_B(\langle 0, j_1\rangle, u) \in F_B, \text{ and } \delta_B(\langle 0, j_2\rangle, u) \notin F_B.$$

Since all the states in B are reachable and pairwise distinguishable, B is a minimal DFA. Thus, any DFA that accepts $L(M)L(N)^R$ has at least 2^{n-1} states. *q.e.d.*

Now, we consider the case when $m = 1$ and $n = 2$.

Lemma 5.4 *There exists a one-state DFA M and a two-state DFA N such that any DFA that accepts $L(M)L(N)^R$ has at least two states.*

Proof: M is defined the same as in Theorem 5.16, and let $N = (Q_N, \Sigma, \delta_N, 0, \{1\})$ be a DFA, where $Q_N = \{0, 1\}$, $\Sigma = \{a, b\}$, and the transitions are given by

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

It is easy to see that $L(N)$ contains all the strings over $\{a, b\}$ that has at least one b. So $L(N)^R = L(N)$ and

$$L(M)L(N)^R = \Sigma^* L(N) = L(N).$$

N is a minimal DFA that accepts $L(M)L(N)^R$. Its two states are reachable and distinguishable obviously. *q.e.d.*

Lastly, we consider the case when $m \geq 1$ and $n = 1$. When $L_2 = \emptyset$, for any L_1, a one-state DFA always accepts $L_1 L_2^R = \emptyset$. When $L_2 = \Sigma^*$, $L_1 L_2^R = L_1 \Sigma^*$, since $(\Sigma^*)^R = \Sigma^*$. According to Theorem 3 in [111], which states that, for any DFA A of size $m \geq 1$, the state complexity of $L(A)\Sigma^*$ is m, we can get the following corollary immediately.

Corollary 5.1 *Given an integer $m \geq 1$, there exists an m-state DFA M and a one-state DFA N such that any DFA that accepts $L(M)L(N)^R$ has at least m states.*

After summarizing Theorems 5.14, 5.15, and 5.16, Lemma 5.4 and Corollary 5.1, we obtain the state complexity of the combined operation $L_1 L_2^R$.

Theorem 5.17 *For integers $m \geq 1$, $n \geq 1$, $m2^n - 2^{n-1} - m + 1$ states are both necessary and sufficient in the worst case for a DFA to accept $L(M)L(N)^R$, where M is an m-state DFA and N is an n-state DFA.*

5.2 State Complexity of Catenation Combined with Union and Intersection

In this section, we will present and prove the state complexities of $L_1(L_2 \cup L_3)$ and $L_1(L_2 \cap L_3)$. All the results in this section are from our paper [11].

5.2.1 State Complexity of $L_1(L_2 \cup L_3)$

In this subsection, we consider the state complexity of $L(A)(L(B) \cup L(C))$ for three DFAs A, B, C of sizes $m, n, p \geq 1$, respectively [11]. We first obtain the following upper bound $(m-k)(2^{n+p} - 2^n - 2^p + 2) + k2^{n+p-2}$ (Theorem 5.18), and then show that this bound is tight for $m, n, p \geq 1$, except the situations when $m \geq 2$ and $n = p = 1$ (Theorems 5.19 and 5.20).

Theorem 5.18 *For integers $m, n, p \geq 1$, let A, B and C be three DFAs with m, n and p states, respectively, where A has k final states. Then there exists a DFA of at most $(m-k)(2^{n+p} - 2^n - 2^p + 2) + k2^{n+p-2}$ states that accepts $L(A)(L(B) \cup L(C))$.*

Proof: Let $A = (Q_1, \Sigma, \delta_1, s_1, F_1)$ where $|F_1| = k$, $B = (Q_2, \Sigma, \delta_2, s_2, F_2)$, and $C = (A_3, \Sigma, \delta_3, s_3, F_3)$. We construct $D = (Q, \Sigma, \delta, s, F)$ such that

$$
\begin{aligned}
Q = &\{\langle q_1, q_2, q_3 \rangle \mid q_1 \in Q_1 - F_1, q_2 \in 2^{Q_2} - \{\emptyset\}, q_3 \in 2^{Q_3} - \{\emptyset\}\} \\
&\cup \{\langle q_1, \emptyset, \emptyset \rangle \mid q_1 \in Q_1 - F_1\} \\
&\cup \{\langle q_1, \{s_2\} \cup q_2, \{s_3\} \cup q_3 \rangle \mid q_1 \in F_1, q_2 \in 2^{Q_2 - \{s_2\}}, q_3 \in 2^{Q_3 - \{s_3\}}\}, \\
s = &\langle s_1, \emptyset, \emptyset \rangle \text{ if } s_1 \notin F_1, s = \langle s_1, \{s_2\}, \{s_3\} \rangle \text{ otherwise}, \\
F = &\{\langle q_1, q_2, q_3 \rangle \in Q \mid q_2 \cap F_2 \neq \emptyset \text{ or } q_3 \cap F_3 \neq \emptyset\}, \\
\delta(&\langle q_1, q_2, q_3 \rangle, a) = \langle q_1', q_2', q_3' \rangle, \text{ for } a \in \Sigma, \text{ where } q_1' = \delta_1(q_1, a) \text{ and}, \\
&\qquad \text{for } i \in \{2, 3\}, q_i' = S_i \cup \{s_i\} \text{ if } q_1' \in F_1, q_i' = S_i \text{ otherwise}, \\
&\qquad S_i = \cup_{r \in q_i} \{\delta_i(r, a)\}.
\end{aligned}
$$

Intuitively, Q is a set of triples such that the first component of each triple is a state in Q_1 and the second and the third components are subsets of Q_2 and Q_3, respectively.

We notice that if the first component of a state is a non-final state of Q_1, the other two components are either both the empty set or both nonempty sets. This is because the two components always change from the empty set to a non-empty set at the same time. This is the reason to have the first and second terms of Q.

Also, we notice that if the first component of a state of D is a final state of A, then the second component and the third component of the state must contain the initial state of B and C, respectively. This is described by the third term of Q.

Clearly, the size of Q is $(m-k)(2^{n+p}-2^n-2^p+2)+k2^{n+p-2}$. Moreover, one can easily verify that $L(D)=L(A)(L(B)\cup L(C))$. *q.e.d.*

In the following, we consider the conditions under which this bound is tight. We know that a complete DFA of size one only accepts either $\emptyset$ or Σ^*. Thus, when $n=p=1$, $L(A)(L(B)\cup L(C))=L(A)\Sigma^*$ if either $L(B)=\Sigma^*$ or $L(C)=\Sigma^*$, and $L(A)(L(B)\cup L(C))=\emptyset$ otherwise. Therefore, in such cases, the state complexity of $L(A)(L(B)\cup L(C))$ is m as shown in [111].

Now, we consider the case when $n=1$ and $p\geq 2$. Since $L(B)\cup L(C)=L(C)$ when $L(B)=\emptyset$, it is clear that the state complexity of $L(A)(L(B)\cup L(C))$ is equal to that of $L(A)L(C)$, $m2^p-k2^{p-1}$ given in [111], which coincides with the upper bound obtained in Theorem 5.18. The situation is analogous to the case when $n\geq 2$ and $p=1$.

Next, we consider the case when $m=1$ and $n,p\geq 2$.

Theorem 5.19 *Let A be a DFA of size 1. Then for integers $n,p\geq 2$, there exist DFAs B and C with n and p states, respectively, such that any DFA that accepts $L(A)(L(B)\cup L(C))$ has at least 2^{n+p-2} states.*

Proof: We use a four-letter alphabet $\Sigma=\{a,b,c,d\}$, and let A be the DFA that accepts Σ^*.

Let $B=(Q_2,\Sigma,\delta_2,0,\{n-1\})$, as shown in Figure 5.15, where $Q_2=\{0,1,\ldots,n-1\}$, and the transitions are given by

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

Let $C=(Q_3,\Sigma,\delta_3,0,\{p-1\})$, whose transition diagram is similar to the one shown in Figure 5.15, where $Q_3=\{0,1,\ldots,p-1\}$, and the transitions are given by

- $\delta_3(i,x)=i$ for $i\in Q_3$, where $x\in\{a,c\}$;
- $\delta_3(i,b)=i+1 \bmod p$, for $i\in\{0,\ldots,p-1\}$;

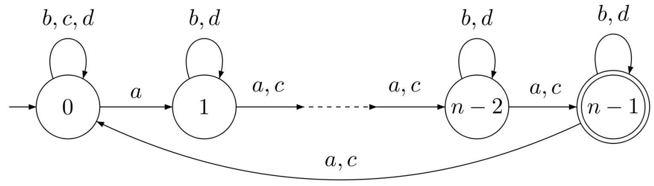

Figure 5.15: The DFA B used for showing that the upper bound in Theorem 5.18 is attainable when $m = 1$ and $n, p \geq 2$

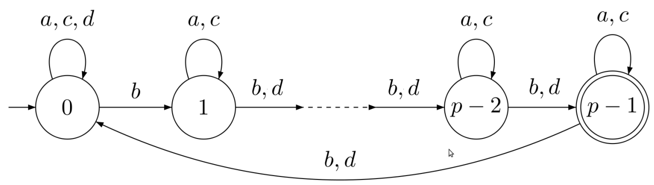

Figure 5.16: The DFA C used for showing that the upper bound in Theorem 5.18 is attainable when $m = 1$ and $n, p \geq 2$

- $\delta_3(0, d) = 0$, $\delta_3(i, d) = i + 1 \bmod p$, for $i \in \{1, \ldots, p-1\}$.

Let $D = (Q, \{a, b, c, d\}, \delta, \langle 0, \{0\}, \{0\}\rangle, F)$ be the DFA that accepts the language $L(A)(L(B) \cup L(C))$ constructed from those DFAs exactly as described in the proof of the previous theorem, where

$$
\begin{aligned}
Q &= \{\langle 0, \{0\} \cup q_2, \{0\} \cup q_3\rangle \mid q_2 \in 2^{Q_2 - \{0\}}, q_3 \in 2^{Q_3 - \{0\}}\}, \\
F &= \{\langle q_1, q_2, q_3\rangle \in Q \mid n - 1 \in q_2 \text{ or } p - 1 \in q_3\}.
\end{aligned}
$$

We omit the definition of the transitions.

Then we prove that the size of Q 2^{n+p-2} 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.

For (I), we first show that all states $\langle 0, q_2, q_3\rangle$ such that $q_3 = \{0\}$ are reachable by induction on the size of q_2.

The basis clearly holds, since the initial state is the only state whose second component is of size 1.

In the induction step, we assume that all states $\langle 0, q_2, \{0\}\rangle$ such that $|q_2| < k$ are reachable. Then we consider the states $\langle 0, q_2, \{0\}\rangle$ where $|q_2| = k$. Let $q_2 = \{0, j_2, \ldots, j_k\}$ such that $0 < j_2 < j_3 < \cdots < j_k \le n-1$. Note that the states such that $j_2 = 1$ can be reached as follows

$$\langle 0, \{0, 1, j_3, \ldots, j_k\}, \{0\}\rangle = \delta(\langle 0, \{0, j_3 - 1, \ldots, j_k - 1\}, \{0\}\rangle, a),$$

where $\{0, j_3 - 1, \ldots, j_k - 1\}$ is of size $k-1$. Then the states such that $j_2 > 1$ can be reached from these states as follows

$$\langle 0, \{0, j_2, \ldots, j_k\}, \{0\}\rangle = \delta(\langle 0, \{0, 1, j_3 - t, \ldots, j_k - t\}, \{0\}\rangle, c^t), \text{ where } t = j_2 - 1.$$

After this induction, all states such that the third component is $\{0\}$ have been reached. Then it is clear that, from each of these states $\langle 0, q_2, \{0\}\rangle$, all states in Q such that the second component is q_2 and the size of their third component is larger than one can be reached by using the same induction step but using the transitions on letters b and d.

Next, we show that any two distinct states $\langle 0, q_2, q_3\rangle$ and $\langle 0, q_2', q_3'\rangle$ in Q are not equivalent. We only consider the situations where $q_2 \ne q_2'$, since the other case can be shown analogously. Without loss of generality, there exists a state r such that $r \in q_2$ and $r \notin q_2'$. It is clear that $r \ne 0$. Let $w = d^{p-1}c^{n-1-r}$. Then $\delta(\langle 0, q_2, q_3\rangle, w) \in F$ but $\delta(\langle 0, q_2', q_3'\rangle, w) \notin F$. *q.e.d.*

Next we consider the more general case when $m, n, p \ge 2$.

Example 5.1 We use a five-letter alphabet $\Sigma = \{a, b, c, d, e\}$ in the following three DFAs, which are modified from the two DFAs in the proof of Theorem 1 in [111].

Let $A = (Q_1, \Sigma, \delta_1, 0, \{m-1\})$, where $Q_1 = \{0, \ldots, m-1\}$ and, for each state $i \in Q_1$, $\delta_1(i, a) = j$, $j = (i+1) \bmod m$, $\delta_1(i, x) = 0$, if $x \in \{b, d\}$, and $\delta_1(i, x) = i$, if $x \in \{c, e\}$.

Let $B = (Q_2, \Sigma, \delta_2, 0, \{n-1\})$, where $Q_2 = \{0, \ldots, n-1\}$ and, for each state $i \in Q_2$, $\delta_2(i, b) = j$, $j = (i+1) \bmod m$, $\delta_2(i, c) = 1$, and $\delta_2(i, x) = i$, if $x \in \{a, d, e\}$.

Let $C = (Q_3, \Sigma, \delta_3, 0, \{p-1\})$, where $Q_3 = \{0, \ldots, p-1\}$ and, for each state $i \in Q_3$, $\delta_3(i, d) = j$, $j = (i+1) \bmod m$, $\delta_3(i, e) = 1$, and $\delta_3(i, x) = i$, if $x \in \{a, b, c\}$. $\square$

Following the construction in the proof of Theorem 5.18, the DFA D can be constructed from the DFAs in Example 5.1. It shows that the upper bound is attainable for $m, n, p \geq 2$. We note that similar to the proof of Theorem 5.19, the DFAs B and C in this example change their states on disjoint letter sets, $\{b, c\}$ and $\{d, e\}$. Thus, by using a proof that is similar to the proof of Theorem 1 in [111] that shows the upper bound on the state complexity of catenation can be attained, we can easily verify that there are at least $(m-1)(2^{n+p}-2^n-2^p+2)+2^{n+p-2}$ distinct equivalence classes of the right-invariant relation induced by $L(A)(L(B) \cup L(C))$ [53]. Therefore, the upper bound can be attained and the following theorem holds.

Theorem 5.20 *Given three integers $m, n, p \geq 2$, there exists a DFA A of m states, a DFA B of n states, and a DFA C of p states such that any DFA that accepts $L(A)(L(B) \cup L(C))$ has at least $(m-1)(2^{n+p}-2^n-2^p+2)+2^{n+p-2}$ states.*

A natural question is that, if we reduce the size of the alphabet used in DFAs A, B, C, using a three-letter alphabet, can we attain the upper bound as well? We give a positive answer in the next theorem under the condition $m, n, p \geq 3$.

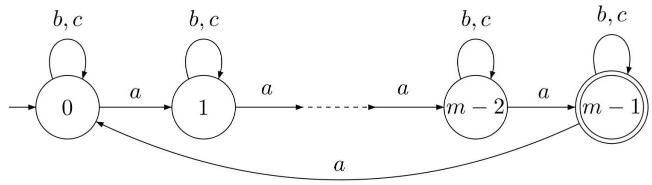

Figure 5.17: Witness DFA A for Theorem 5.21

Theorem 5.21 *For integers $m, n, p \geq 3$, there exist DFAs A, B and C of m, n, and p states, respectively, defined over a three-letter alphabet, such that any DFA that accepts $L(A)(L(B) \cup L(C))$ has at least $(m-1)(2^{n+p}-2^n-2^p+2)+2^{n+p-2}$ states.*

Proof: We define the following three automata over the three-letter alphabet $\Sigma = \{a, b, c\}$.

Let $A = (Q_1, \Sigma, \delta_1, 0, \{m-1\})$ be a DFA, as shown in Figure 5.17, where $Q_1 = \{0, 1, \ldots, m-1\}$, and the transitions are given as follows:

- $\delta_1(i, a) = i + 1$ for $i \in \{0, \ldots, m-2\}$, $\delta_1(m-1, a) = 0$;
- $\delta_1(i, e) = i$ for $i \in Q_1$, where $e \in \{b, c\}$.

Let $B = (Q_2, \Sigma, \delta_2, 0, \{n-1\})$ be a DFA, as shown in Figure 5.18, where $Q_2 = \{0, 1, \ldots, n-1\}$, and the transitions are given as follows:

- $\delta_2(i, a) = i$ for $i \in \{0, \ldots, n-3\}$, $\delta_2(n-2, a) = n-1$, $\delta_2(n-1, a) = n-2$;
- $\delta_2(i, b) = i + 1$ for $i \in \{0, \ldots, n-2\}$, $\delta_2(n-1, b) = n-1$;
- $\delta_2(i, c) = i$ for $i \in Q_2$.

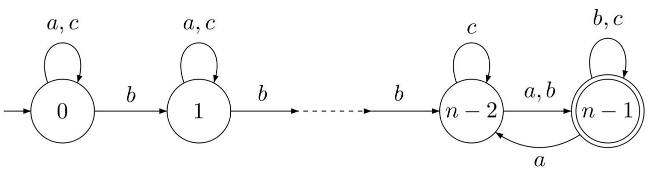

Figure 5.18: Witness DFA B for Theorem 5.21

Let $C = (Q_3, \Sigma, \delta_3, 0, \{p-1\})$ be a DFA, as shown in Figure 5.19, whose transition diagram is similar to that of the DFA B, where $Q_3 = \{0, 1, \ldots, p-1\}$, and the transitions are given as follows:

- $\delta_3(i, a) = i$ for $i \in \{0, \ldots, p-3\}$, $\delta_3(p-2, a) = p-1$, $\delta_3(p-1, a) = p-2$;
- $\delta_3(i, b) = i$ for $i \in Q_3$;
- $\delta_3(i, c) = i + 1$ for $i \in \{0, \ldots, p-2\}$, $\delta_3(p-1, c) = p-1$.

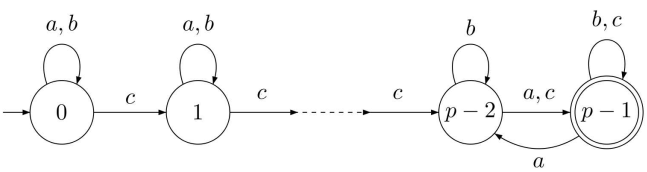

Figure 5.19: Witness DFA C for Theorem 5.21

Let $D = (Q, \{a, b, c\}, \delta, \langle 0, \emptyset, \emptyset\rangle, F)$ be the DFA that accepts the language $L(A)(L(B) \cup L(C))$ constructed from those DFAs exactly as described in the proof of the previous theorem, where

$$
\begin{aligned}
Q &= \{\langle q_1, q_2, q_3\rangle \mid q_1 \in Q_1 - \{m-1\}, q_2 \in 2^{Q_2} - \{\emptyset\}, q_3 \in 2^{Q_3} - \{\emptyset\}\} \\
&\quad \cup \{\langle q_1, \emptyset, \emptyset\rangle \mid q_1 \in Q_1 - \{m-1\}\} \\
&\quad \cup \{\langle m-1, \{0\} \cup q_2, \{0\} \cup q_3\rangle \mid q_2 \in 2^{Q_2 - \{0\}}, q_3 \in 2^{Q_3 - \{0\}}\}, \\
F &= \{\langle q_1, q_2, q_3\rangle \in Q \mid n-1 \in q_2 \text{ or } p-1 \in q_3\}.
\end{aligned}
$$

We omit the definition of transitions.

Then we prove that the size of Q $m(2^{n+p} - 2^n - 2^p + 2) + 2^{n+p-2}$ 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.

Now we consider (I). It is clear that states $\langle q_1, \emptyset, \emptyset\rangle$, for $q_1 \in Q_1 - \{m-1\}$, are reachable from the initial state on strings a^{q_1}, and the state $\langle m-1, \{0\}, \{0\}\rangle$ can be reached from $\langle m-2, \emptyset, \emptyset\rangle$ on the letter a.

We first show by induction on the size of the second component that any remaining state in Q such that its third component is $\{0\}$ can be reached from the state $\langle m-1, \{0\}, \{0\}\rangle$. We only use strings over the letters a, b. Thus, the last component remains $\{0\}$.

Basis: for any $i \in \{0, \dots, m-2\}$, the state $\langle i, \{0\}, \{0\}\rangle$ can be reached from the state $\langle m-1, \{0\}, \{0\}\rangle$ on the string a^{i+1}. Then for any $i \in \{0, \dots, m-2\}$ and $j \in \{1, \dots, n\}$,

$$\langle i, \{j\}, \{0\}\rangle = \delta(\langle i, \{0\}, \{0\}\rangle, b^j).$$

Induction step: for $i \in \{0, \dots, m-1\}$, assume that all states $\langle i, q_2, \{0\}\rangle$ such that $|q_2| < k$ are reachable. Then we consider the states $\langle i, q_2, \{0\}\rangle$ where $|q_2| = k$. Let $q_2 = \{j_1, j_2, \dots, j_k\}$ such that $0 \le j_1 < j_2 < \cdots < j_k \le n-1$.

Note that the states such that $j_1 = 0$ are reachable as follows. If either $j_k \le n-3$ or $j_{k-1} = n-2$ and $j_k = n-1$, we have

$$\langle m-1, \{0, j_2, \dots, j_k\}, \{0\}\rangle = \delta(\langle m-2, \{j_2, \dots, j_k\}, \{0\}\rangle, a).$$

If $j_k = n-2$, the states $\langle m-1, \{0, j_2, \dots, j_k\}, \{0\}\rangle$ can be reached from the states $\langle m-2, \{j_2, \dots, j_{k-1}, n-1\}, \{0\}\rangle$ by reading the letter a. If $j_k = n-1$, the states $\langle m-1, \{0, j_2, \dots, j_k\}, \{0\}\rangle$ can be reached from states $\langle m -$

$2, \{j_2, \ldots, j_{k-1}, n-2\}, \{0\}\rangle$ by reading the letter a. In all the cases, we reach the state from a state such that $|q_2| = k-1$. Similarly, we can easily verify that, by reading the letter a, states $\langle 0, \{0, \ldots, j_k\}, \{0\}\rangle$ can be reached from the states $\langle m-1, \{0, \ldots, j_k\}, \{0\}\rangle$ and, for $i \in \{1, \ldots, m-2\}$, the states $\langle i, \{0, \ldots, j_k\}, \{0\}\rangle$ can be reached from the states $\langle i-1, \{0, \ldots, j_k\}, \{0\}\rangle$.

Next, we show that all states such that $0 \notin q_2$ are reachable. Note that the first component of these states cannot be $m-1$. Thus, for $i \in \{0, \ldots, m-2\}$, we have

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

After the induction step, we can verify that all states in Q such that the third component is $\{0\}$ have been reached. Then we consider the states whose third component is non-empty but not $\{0\}$. Note that if the second component of a state does not contain the states $n-2$ and $n-1$ or contains both of them, this component does not change by reading the letter a. Thus, by using the letter c instead of the letter b in the same induction step, we can show that, for $i \in \{0, \ldots, m-1\}$, the states $\langle i, q_2, q_3\rangle$ in Q such that $q_2 \cap \{n-2, n-1\} = \emptyset$ or $\{n-2, n-1\} \subseteq q_2$ are reachable from the state $\langle 0, q_2, \{0\}\rangle$. The remaining states to be considered are the states $\langle i, q_2, q_3\rangle$ such that q_2 contains either $n-2$ or $n-1$ but not both, for $i \in \{0, \ldots, m-1\}$. Assume q_2 contains $n-2$. Then by the same induction with the letters a, c, we can reach the states $\langle i, q_2, q_3\rangle$ and states $\langle i', q_2', q_3'\rangle$, $i, i' \in \{0, \ldots, m-1\}$, from the state $\langle 0, q_2, \{0\}\rangle$ such that $q_2' = (q_2 \cup \{n-1\}) \setminus \{n-2\}$. Moreover, if we replace q_2' with q_2, the union of these two types of states is exactly all states in Q such that their second component is q_2. It is clear that those states $\langle i', q_2, q_3'\rangle$ are reachable from the state $\langle 0, q_2', \{0\}\rangle$ by following the same induction step with letters a, c. An analogous argument can be applied to the states such that q_2 contains $n-1$ but not $n-2$.

Now all the states in Q are reachable, and next we will show that the states of the DFA D are pairwise inequivalent. Let $\langle i, q_2, q_3\rangle$ and $\langle j, q_2', q_3'\rangle$ be two different states. We consider the following two cases:

1. $i < j$. Then the string $a^{m-1-i}b^{n-1}c^{p-1}a$ is accepted by the DFA D starting from the state $\langle i, q_2, q_3\rangle$, but it is not accepted starting from the state $\langle j, q_2', q_3'\rangle$.

2. $i = j$. We only prove for the situation where $q_2 \neq q_2'$, since the proof is analogous when $q_3 \neq q_3'$. Without loss of generality, there exists a state r such that $r \in q_2$ and $r \notin q_2'$.
   If $i = j \neq m - 1$, we can verify that $c^{p-1}b^{n-r-2}a$ is accepted by D from the state $\langle i, q_2, q_3 \rangle$ but not from the state $\langle j, q_2', q_3' \rangle$.
   If $i = j = m - 1$, it is clear that $r \neq 0$. We consider the following three cases.
   (a) $r \in \{1, \ldots, n-3\}$. After reading the letter a, i and j become 0 and we still have $r \in q_2$ and $r \notin q_2'$. Thus, the resulting situation has just been considered.
   (b) $r = n-2$. Then the state $\langle i, q_2, q_3 \rangle$ reaches a final state on $ac^{p-1}ab$, but the state $\langle j, q_2', q_3' \rangle$ does not on the same string.
   (c) $r = n - 1$. Then the state $\langle i, q_2, q_3 \rangle$ reaches a final state by reading $ac^{p-1}a$, but the state $\langle j, q_2', q_3' \rangle$ does not.

q.e.d.

5.2.2 State Complexity of $L_1(L_2 \cap L_3)$

In this subsection, we investigate the state complexity of $L_1(L_2 \cap L_3)$, and show that its upper bound (Theorem 5.22) coincides with its lower bound (Theorems 5.23 and 5.24) [11]. The following theorem gives an upper bound for the state complexity of this combined operation.

Theorem 5.22 *Let L_1, L_2 and L_3 be three regular languages accepted by an m-state, an n-state and a p-state DFA, respectively, for m, n, $p \geq 1$. Then there exists a DFA of at most $m2^{np} - 2^{np-1}$ states that accepts $L_1(L_2 \cap L_3)$.*

We omit the proof of Theorem 5.22 because $m2^{np} - 2^{np-1}$ is the mathematical composition of the state complexities of the individual component operations, which is obviously an upper bound on the state complexity of $L_1(L_2 \cap L_3)$. In the following, we investigate lower bounds on the state complexity of this combined operation under different conditions.

When $n = p = 1$, $L(A)(L(B) \cap L(C)) = L(A)\Sigma^*$ if both $L(B)$ and $L(C)$ are Σ^*. The resulting language is $\emptyset$ otherwise. Thus, the state complexity of

$L(A)(L(B) \cap L(C))$ in this case is the same as that of $L(A)\Sigma^*$: namely, m [111]. When $n = 1$, $p \geq 2$,

$$L(A)(L(B) \cap L(C)) = \begin{cases} \emptyset, & \text{if } L(B) = \emptyset; \\ L(A)L(C), & \text{if} L(B) = \Sigma^*. \end{cases}$$

In this case, the state complexity of the combined operation is $m2^p - 2^{p-1}$, which is the same as that of $L(A)L(C)$ [111]. Similarly, when $n \geq 2$, $p = 1$, the state complexity of $L(A)(L(B) \cap L(C))$ is $m2^n - 2^{n-1}$. Next, we show the upper bound $m2^{np} - 2^{np-1}$ is attainable when $m, n, p \geq 2$.

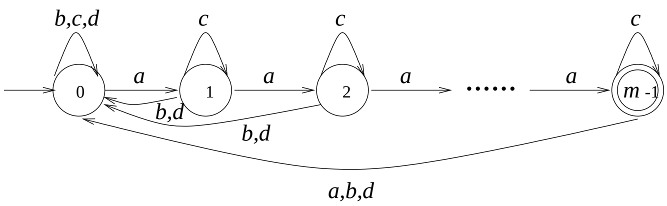

Figure 5.20: The DFA A used for showing that the upper bound in Theorem 5.22 is attainable when $m \geq 2$ and $n, p \geq 1$

Theorem 5.23 *Given three integers $m, n, p \geq 2$, there exists a DFA A of m states, a DFA B of n states and a DFA C of p states such that any DFA that accepts $L(A)(L(B) \cap L(C))$ has at least $m2^{np} - 2^{np-1}$ states.*

Proof: Let $A = (Q_A, \Sigma, \delta_A, 0, F_A)$ be a DFA, as shown in Figure 5.20, where $Q_A = \{0, 1, \ldots, m-1\}$, $F_A = \{m-1\}$, $\Sigma = \{a, b, c, d\}$ and the transitions are given by

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

Let $B = (Q_B, \Sigma, \delta_B, 0, F_B)$ be a DFA, as shown in Figure 5.21, where $Q_B = \{0, 1, \ldots, n-1\}$, $F_B = \{n-1\}$ and the transitions are given by

- $\delta_B(i, x) = i$, $i = 0, \ldots, n-1$, where $x \in \{a, d\}$;
- $\delta_B(i, b) = i + 1 \bmod n$, $i = 0, \ldots, n-1$;

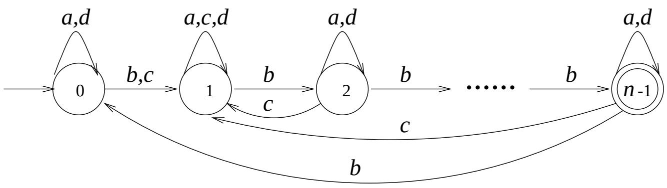

Figure 5.21: The DFA B used for showing that the upper bound in Theorem 5.22 is attainable when $m \geq 2$ and $n, p \geq 1$

- $\delta_B(i, c) = 1$, $i = 0, \ldots, n-1$.

Let $C = (Q_C, \Sigma, \delta_C, 0, F_C)$ be a DFA whose transition diagram is shown in Figure 5.22, where $Q_C = \{0, 1, \ldots, p-1\}$, $F_C = \{p-1\}$ and the transitions are given by

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

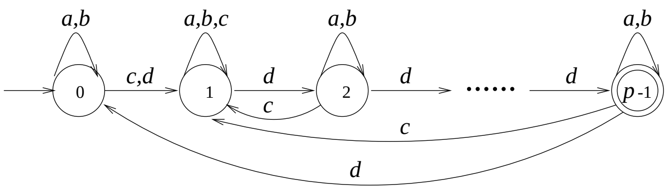

Figure 5.22: The DFA C used for showing that the upper bound in Theorem 5.22 is attainable when $m \geq 2$ and $n, p \geq 1$

We construct the DFA $D = (Q_D, \Sigma, \delta_D, s_D, F_D\}$, where

$$\begin{aligned} Q_D &= \{\langle u, v \rangle \mid u \in Q_B,\ v \in Q_C\}, \\ s_D &= \langle 0, 0 \rangle, \\ F_D &= \{\langle n-1, p-1 \rangle\}, \end{aligned}$$

and for each state $\langle u, v \rangle \in Q_D$ and each letter $e \in \Sigma$,

$$\delta_D(\langle u, v \rangle, e) = \langle u', v' \rangle \text{ if } \delta_B(u, e) = u',\ \delta_C(v, e) = v'.$$

Clearly, there are $n \cdot p$ states in D and $L(D) = L(B) \cap L(C)$. Now we construct another DFA $E = (Q_E, \Sigma, \delta_E, s_E, F_E\}$, where

$$\begin{aligned}
Q_E &= \{\langle q, R\rangle \mid q \in Q_A - F_A,\ R \subseteq Q_D\} \cup \{\langle m-1, S\rangle \mid s_D \in S,\ S \subseteq Q_D\},\\
s_E &= \langle 0, \emptyset\rangle,\\
F_E &= \{\langle q, R\rangle \mid R \cap F_D \neq \emptyset,\ \langle q, R\rangle \in Q_E\},
\end{aligned}$$

and for each state $\langle q, R\rangle \in Q_E$ and each letter $e \in \Sigma$,

$$\delta_E(\langle q, R\rangle, e) = \begin{cases} \langle q', R'\rangle, & \text{if } \delta_A(q, e) = q' \neq m-1,\ \delta_D(R, e) = R'; \\ \langle q', R'\rangle, & \text{if } \delta_A(q, e) = q' = m-1,\ R' = \delta_D(R, e) \cup \{s_D\}. \end{cases}$$

It is easy to see that $L(E) = L(A)(L(B) \cap L(C))$. There are $(m-1) \cdot 2^{np}$ states in the first term of the union for Q_E. In the second term, there are $1 \cdot 2^{np-1}$ states. Thus,

$$|Q_E| = (m-1) \cdot 2^{np} + 1 \cdot 2^{np-1} = m2^{np} - 2^{np-1}.$$

In order to show that E is minimal, we need to show that (I) every state in E is reachable from the initial state and (II) each state defines a distinct equivalence class.

We prove (I) by induction on the size of the second component of states in Q_E. First, any state $\langle q, \emptyset\rangle$, $0 \leq q \leq m-2$, is reachable from s_E by reading the string a^q. Then we consider all states $\langle q, R\rangle$ such that $|R| = 1$. In this case, let $R = \{\langle x, y\rangle\}$. We have

$$\langle q, \{\langle x, y\rangle\}\rangle = \delta_E(\langle 0, \emptyset\rangle, a^m b^x d^y a^q).$$

Notice that the only state $\langle q, R\rangle$ in Q_E such that $q = m-1$ and $|R| = 1$ is $\langle m-1, \{\langle 0, 0\rangle\}\rangle$ since the fact that $q = m-1$ guarantees $\langle 0, 0\rangle \in R$.

Assume that all states $\langle q, R\rangle$ such that $|R| < k$ are reachable. Consider $\langle q, R\rangle$ where $|R| = k$. Let $R = \{\langle x_i, y_i\rangle \mid 1 \leq i \leq k\}$ such that $0 \leq x_1 \leq x_2 \leq \cdots \leq x_k \leq n-1$ if $q \neq m-1$ and $0 = x_1 \leq x_2 \leq \cdots \leq x_k \leq n-1$, $y_1 = 0$, otherwise. We have $\langle q, R\rangle = \delta_E(\langle 0, R'\rangle, a^m b^{x_1} d^{y_1} a^q)$, where

$$R' = \{\langle x_j - x_1, (y_j - y_1 + n) \bmod n\rangle \mid 2 \leq j \leq k\}.$$

The state $\langle 0, R'\rangle$ is reachable from the initial state, since $|R| = k-1$. Thus, $\langle q, R\rangle$ is also reachable.

To prove (II), let $\langle q_1, R_1\rangle$ and $\langle q_2, R_2\rangle$ be two different states in E. We consider the following two cases.

1. $q_1 \neq q_2$. Without loss of generality, we may assume that $q_1 > q_2$. There always exists a string $t = ca^{m-1-q_1}b^{n-1}d^{p-1}$ such that

$$\begin{aligned} \delta_E(\langle q_1, R_1\rangle, t) &\in F_E, \\ \delta_E(\langle q_2, R_2\rangle, t) &\notin F_E. \end{aligned}$$

2. $q_1 = q_2$, $R_1 \neq R_2$. Without loss of generality, we may assume that $|R_1| \geq |R_2|$. Let $\langle x, y\rangle \in R_1 - R_2$. Then

$$\begin{aligned} \delta_E(\langle q_1, R_1\rangle, b^{n-1-x}d^{p-1-y}) &\in F_E, \\ \delta_E(\langle q_2, R_2\rangle, b^{n-1-x}d^{p-1-y}) &\notin F_E. \end{aligned}$$

Thus, the minimal DFA that accepts $L(A)(L(B) \cap L(C))$ has at least $m2^{np} - 2^{np-1}$ states for $m, n, p \geq 2$. *q.e.d.*

Now we consider the case when $m = 1$, i.e., $L(A) = \Sigma^*$.

Theorem 5.24 *Given two integers $n, p \geq 2$, there exists a DFA A of one state, a DFA B of n states and a DFA C of p states such that any DFA that accepts $L(A)(L(B) \cap L(C))$ has at least 2^{np-1} states.*

Proof: As we mentioned in the proof of Theorem 5.23, when $n = 1$, $L(A)(L(B) \cap L(C))$ is either $\emptyset$ or $L(A)L(C)$. It has been proved in [111] that the state complexity of $L(A)L(C)$ is 2^{p-1} for $m = 1$, $p \geq 2$. If $m = n = p = 1$, $L(A)(L(B) \cap L(C))$ is either $\emptyset$ or Σ^*, which are both accepted by one-state DFAs. Similarly, when $n \geq 2$, $p = 1$, the state complexity of $L(A)(L(B) \cap L(C))$ is 2^{n-1}.

When $m = 1$, $n \geq 2$, $p \geq 2$, we give the following construction. Let $A = (\{0\}, \Sigma, \delta_A, 0, \{0\})$ be a DFA, where $\Sigma = \{a, b, c, d, e\}$ and $\delta_A(0, t) = 0$ for any letter $t \in \Sigma$. It is clear that $L(A) = \Sigma^*$.

Let $B = (Q_B, \Sigma, \delta_B, 0, F_B)$ be a DFA, as shown in Figure 5.23, where $Q_B = \{0, 1, \ldots, n-1\}$, $F_B = \{n-1\}$ and the transitions are given by

- $\delta_B(i, a) = i + 1 \bmod n$, $i = 0, \ldots, n-1$;
- $\delta_B(i, b) = i$, $i = 0, \ldots, n-1$;
- $\delta_B(0, c) = 1$, $\delta_B(j, c) = j$, $j = 1, \ldots, n-1$;
- $\delta_B(0, d) = 0$, $\delta_B(j, d) = j + 1$, $j = 1, \ldots, n-2$, $\delta_B(n-1, d) = 1$;
- $\delta_B(i, e) = i$, $i = 0, \ldots, n-1$.

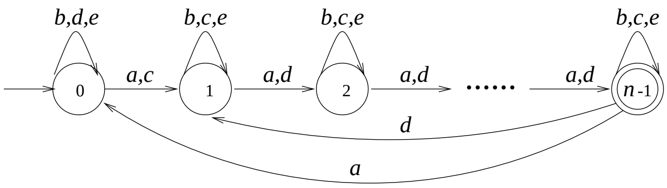

Figure 5.23: Witness DFA B for Theorems 5.24

Let $C = (Q_C, \Sigma, \delta_C, 0, F_C)$ be a DFA, as shown in Figure 5.24, where $Q_C = \{0, 1, \dots, p-1\}$, $F_C = \{p-1\}$ and the transitions are given by

- $\delta_C(i, a) = i$, $i = 0, \dots, p-1$;
- $\delta_C(i, b) = i + 1 \bmod p$, $i = 0, \dots, p-1$;
- $\delta_C(0, c) = 1$, $\delta_C(j, c) = j$, $j = 1, \dots, p-1$;
- $\delta_C(i, d) = i$, $i = 0, \dots, p-1$;
- $\delta_C(0, e) = 0$, $\delta_C(j, e) = j + 1$, $j = 1, \dots, p-2$, $\delta_C(p-1, e) = 1$.

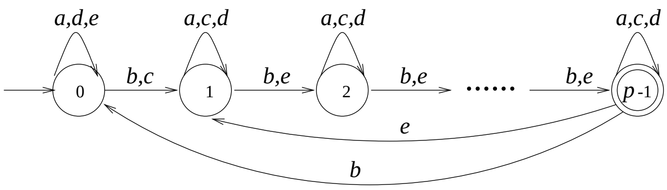

Figure 5.24: Witness DFA C for Theorems 5.24

Construct the DFA $D = (Q_D, \Sigma, \delta_D, s_D, F_D\}$ that accepts $L(B) \cap L(C)$ in the same way as the proof of Theorem 5.23, where

$$
\begin{aligned}
Q_D &= \{\langle u, v\rangle \mid u \in Q_B,\ v \in Q_C\},\\
s_D &= \langle 0, 0\rangle,\\
F_D &= \{\langle n-1, p-1\rangle\},
\end{aligned}
$$

and for each state $\langle u, v\rangle \in Q_D$ and each letter $t \in \Sigma$,

$$\delta_D(\langle u, v\rangle, t) = \langle u', v'\rangle \text{ if } \delta_B(u, t) = u',\ \delta_C(v, t) = v'.$$

Now we construct the DFA $E = (Q_E, \Sigma, \delta_E, s_E, F_E\}$, where

$$
\begin{aligned}
Q_E &= \{\langle 0, R\rangle \mid \langle 0, 0\rangle \in R, R \subseteq Q_D\},\\
s_E &= \langle 0, \{\langle 0, 0\rangle\}\rangle,\\
F_E &= \{\langle 0, R\rangle \mid R \cap F_D \neq \emptyset,\ \langle 0, R\rangle \in Q_E\},
\end{aligned}
$$

and for each state $\langle 0, R\rangle \in Q_E$ and each letter $t \in \Sigma$,

$$\delta_E(\langle 0, R\rangle, t) = \langle 0, R'\rangle \text{ where } R' = \delta_D(R, t).$$

Note that $\langle 0, 0\rangle \in R$ for every state $\langle 0, R\rangle \in Q_E$, since 0 is the only state in A and it is both initial and final. It is easy to see that $L(E) = L(A)(L(B) \cap L(C))$ and E has $2^{np} - 2^{np-1} = 2^{np-1}$ states in total. Now we show that E is a minimal DFA by (I) every state in E is reachable from the initial state and (II) each state defines a distinct equivalence class.

We again prove (I) by induction on the size of the second component of states in Q_E. First, the only state in $\langle 0, R\rangle \in Q_E$ such that $|R| = 1$ is the initial state, $\langle 0, \{\langle 0, 0\rangle\}\rangle$.

Assume that all states $\langle 0, R\rangle$ such that $|R| \leq k$ are reachable. Consider $\langle 0, R\rangle$ where $|R| = k + 1$. Let $R = \{\langle 0, 0\rangle, \langle x_1, y_1\rangle, \ldots, \langle x_k, y_k\rangle\}$ such that $0 \leq x_1 \leq x_2 \leq \ldots \leq x_k \leq n - 1$. We consider the following three cases.

Case 1. $\langle 0, y_1\rangle \in R$, $y_1 \geq 1$. If there exists $\langle 0, y_i\rangle \in R$, $y_i \geq 1$, $1 \leq i \leq k$, then $x_1 = 0$ and $y_1 \geq 1$, since $0 \leq x_1 \leq x_2 \leq \cdots \leq x_k \leq n - 1$. For this case, we have

$$\langle 0, R\rangle = \delta_E(\langle 0, R_1\rangle, be^{y_1 - 1}),$$

where

$$R_1 = \{\langle 0, 0\rangle\} \cup S_1 \cup T_1,$$

$$S_1 = \{\langle x_j, p - 1\rangle \mid \langle x_j, 0\rangle \in R, x_j \neq 0\},$$

$$T_1 = \{\langle x_j, (y_j - y_1 + p - 1) \bmod (p - 1)\rangle \mid \langle x_j, y_j\rangle \in R, y_j \neq 0, 2 \leq j \leq k\}.$$

Notice that $\langle 0, 0\rangle \notin S_1 \cup T_1$ and $S_1 \cap T_1 = \emptyset$. So the state $\langle 0, R\rangle$ is reachable from the initial state, since $|R_1| = k$ and $\langle 0, R_1\rangle$ is reachable.

Case 2. $x_1 \geq 1$, $\langle x_i, 0\rangle \in R$, $1 \leq i \leq k$. It is easy to see that every $x_i \geq 1$ because $x_i \geq x_1$. We have

$$\langle 0, R\rangle = \delta_E(\langle 0, R_2\rangle, ad^{x_i - 1}),$$

where

$$R_2 = \{\langle 0, 0\rangle\} \cup T_2,$$

$$T_2 = \{\langle (x_j - x_i + n - 1) \bmod (n - 1), y_j\rangle \mid \langle x_j, y_j\rangle \in R, 1 \leq j \leq k, j \neq i\}.$$

There are k elements in R_2. So the state $\langle 0, R\rangle$ is also reachable for this case.

Case 3. $x_1 \geq 1$, $y_i \geq 1$, $1 \leq i \leq k$, because every $x_i \geq x_1 \geq 1$, we have

$$\langle 0, R\rangle = \delta_E(\langle 0, R_3\rangle, cd^{x_1-1}e^{y_1-1}),$$

where

$$R_3 = \{\langle 0,0\rangle\} \cup T_3,$$

$T_3 = \{\langle (x_j - x_1 + 1), (y_j - y_1 + p - 1) \bmod (p-1)+1\rangle \mid \langle x_j, y_j\rangle \in R, 2 \leq j \leq k\}$.

So every state $\langle 0, R\rangle$ in E is reachable when $|R| = k+1$.

To prove (II), let $\langle 0, R\rangle$ and $\langle 0, R'\rangle$ be two different states in E. Without loss of generality, we may assume that $|R| \geq |R'|$. So we can always find $\langle x, y\rangle \in R - R'$. Clearly, $\langle x, y\rangle \neq \langle 0, 0\rangle$. So there exists a string $w = a^{n-1-x}b^{p-1-y}$ such that

$$\begin{aligned} \delta_E(\langle 0, R\rangle, w) &\in F_E, \\ \delta_E(\langle 0, R'\rangle, w) &\notin F_E. \end{aligned}$$

Thus, the minimal DFA that accepts $\Sigma^*(L(B) \cap L(C))$ has at least 2^{np-1} states for $n \geq 1$, $p \geq 1$. *q.e.d.*

This lower bound coincides with the upper bound given in Theorem 5.22. Thus, the bounds are also tight for the case when $m = 1$, $n, p \geq 2$.

5.3 State Complexity of Union and Intersection Combined with Star and Reversal

In this section, we will show the state complexities of $L_1^* \cup L_2$ and $L_1^* \cap L_2$. All the results in this section are from our paper [33].

5.3.1 State Complexity of $L_1^* \cup L_2$

We consider the state complexity of $L_1^* \cup L_2$, where L_1 and L_2 are regular languages accepted by m-state and n-state DFAs, respectively. It has been proved that the state complexity of L_1^* is $\frac{3}{4}2^m$ and the state complexity of $L_1 \cup L_2$ is mn [72, 111]. The mathematical composition of these functions is $\frac{3}{4}2^m \cdot n$. In the following, we show that this upper bound can be decreased [33].

Theorem 5.25 *For any m-state DFA $M = (Q_M, \Sigma, \delta_M, s_M, F_M)$ and n-state DFA $N = (Q_N, \Sigma, \delta_N, s_N, F_N)$ such that $|F_M - \{s_M\}| = k \geq 1$, $m \geq 2$, $n \geq 1$, there exists a DFA of at most $(2^{m-1} + 2^{m-k-1}) \cdot n - n + 1$ states that accepts $L(M)^* \cup L(N)$.*

Proof: Let $M = (Q_M, \Sigma, \delta_M, s_M, F_M)$ be a complete DFA of m states. Denote $|F_M - \{s_M\}|$ by F_0. Then $F_0 = k \geq 1$. Let $N = (Q_N, \Sigma, \delta_N, s_N, F_N)$ be another complete DFA of n states. Let $M' = (Q_{M'}, \Sigma, \delta_{M'}, s_{M'}, F_{M'})$ be a DFA where

$$\begin{aligned}
&s_{M'} \notin Q_M \text{ is a new initial state,} \\
&Q_{M'} = \{s_{M'}\} \cup \{P \mid P \subseteq (Q_M - F_0) \ \& \ P \neq \emptyset\} \\
&\qquad \cup \{R \mid R \subseteq Q_M \ \& \ s_M \in R \ \& \ R \cap F_0 \neq \emptyset\}, \\
&F_{M'} = \{s_{M'}\} \cup \{R \mid R \subseteq Q_M \ \& \ R \cap F_M \neq \emptyset\},
\end{aligned}$$

and for $R \subseteq Q_M$ and $a \in \Sigma$,

$$\delta_{M'}(s_{M'}, a) = \begin{cases} \{\delta_M(s_M, a)\}, & \text{if } \delta_M(s_M, a) \cap F_0 = \emptyset; \\ \{\delta_M(s_M, a)\} \cup \{s_M\}, & \text{otherwise,} \end{cases}$$

$$\delta_{M'}(R, a) = \begin{cases} \{\delta_M(R, a)\}, & \text{if } \delta_M(R, a) \cap F_0 = \emptyset; \\ \{\delta_M(R, a)\} \cup \{s_M\}, & \text{otherwise.} \end{cases}$$

It is clear that M' accepts $L(M)^*$. In the second term of the union for $Q_{M'}$ there are $2^{m-k} - 1$ states. And in the third term, there are $(2^k - 1)2^{m-k-1}$ states. So M' has $2^{m-1} + 2^{m-k-1}$ states in total. Now we construct another DFA $A = (Q, \Sigma, \delta, s, F)$ where

$$\begin{aligned}
&s = \langle s_{M'}, s_N \rangle, \\
&Q = \{\langle i, j \rangle \mid i \in Q_{M'} - \{s_{M'}\}, j \in Q_N\} \cup \{s\}, \\
&\delta(\langle i, j \rangle, a) = \langle \delta_{M'}(i, a), \delta_N(j, a) \rangle, \ \langle i, j \rangle \in Q, \ a \in \Sigma, \\
&F = \{\langle i, j \rangle \mid i \in F_{M'} \text{ or } j \in F_N\}.
\end{aligned}$$

We can see that

$$L(A) = L(M') \cup L(N) = L(M)^* \cup L(N).$$

Note $\langle s_{M'}, j \rangle \notin Q$, for $j \in Q_N - \{s_N\}$, because there is no transition going into $s_{M'}$ in the DFA M'. So there are at least $n - 1$ states in Q that are not reachable. Thus, the number of states of minimal DFA that accepts $L(M)^* \cup L(N)$

is no more than

$$|Q| = (2^{m-1} + 2^{m-k-1}) \cdot n - n + 1.$$

q.e.d.

If s_M is the only final state of $M(k = 0)$, then $L(M)^* = L(M)$.

Corollary 5.2 *For any m-state DFA $M = (Q_M, \Sigma, \delta_M, s_M, F_M)$ and n-state DFA $N = (Q_N, \Sigma, \delta_N, s_N, F_N)$, $m > 1$, $n > 0$, there exists a DFA A of at most $\frac{3}{4}2^m \cdot n - n + 1$ states such that $L(A) = L(M)^* \cup L(N)$.*

Proof: Let k be defined as in the above proof. There are two cases in the following.

(I) $k = 0$. In this case, $L(M)^* = L(M)$. Then A needs at most $m \cdot n$ states, which is less than $\frac{3}{4}2^m \cdot n - n + 1$ when $m > 1$.

(II) $k \geq 1$. The claim is clearly true by Theorem 5.25.

q.e.d.

Next, we show that the upper bound $\frac{3}{4}2^m \cdot n - n + 1$ is attainable.

Theorem 5.26 *Given two integers $m \geq 2$, $n \geq 2$, there exists a DFA M of m states and a DFA N of n states such that any DFA that accepts $L(M)^* \cup L(N)$ has at least $\frac{3}{4}2^m \cdot n - n + 1$ states.*

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

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

The transition diagram of M is shown in Figure 5.25.
Let $N = (Q_N, \Sigma, \delta_N, 0, \{n-1\})$ be another DFA, where $Q_N = \{0, 1, \ldots, n-1\}$ and

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

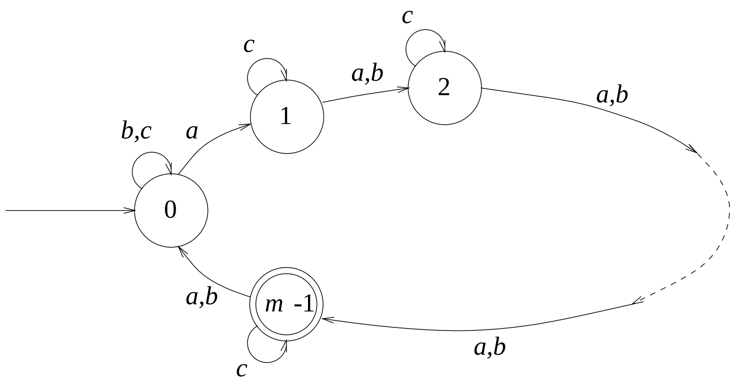

Figure 5.25: Witness DFA M for Theorems 5.26 and 5.29

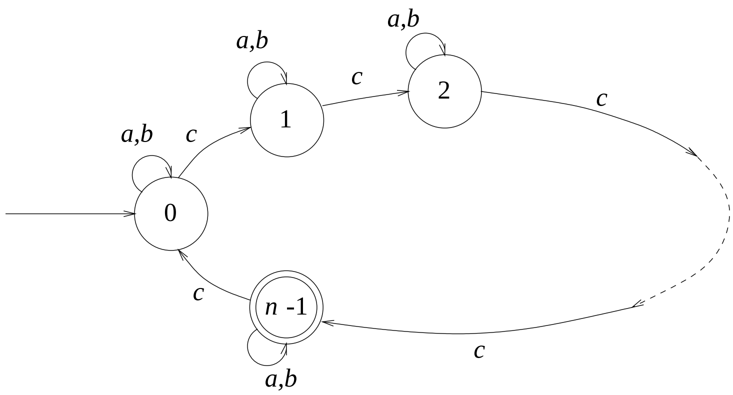

Figure 5.26: Witness DFA N for Theorems 5.26 and 5.29

The transition diagram of N is shown in Figure 5.26.

It has been proved in [111] that the minimal DFA that accepts the star of an m-state DFA language has $\frac{3}{4}2^m$ states in the worst case. M is a modification of the worst-case example given in [111] by adding a c-loop to every state. So we design a $\frac{3}{4}2^m$-state, minimal DFA $M' = (Q_{M'}, \Sigma, \delta_{M'}, s_{M'}, F_{M'})$ that accepts $L(M)^*$, where

$$\begin{aligned}
&s_{M'} \notin Q_M \text{ is a new initial state,}\\
&Q_{M'} = \{s_{M'}\} \cup \{P \mid P \subseteq \{0, 1, \ldots, m-2\} \;\&\; P \neq \emptyset\}\\
&\qquad \cup \{R \mid R \subseteq \{0, 1, \ldots, m-1\} \;\&\; 0 \in R \;\&\; m-1 \in R\},\\
&F_{M'} = \{s_{M'}\} \cup \{R \mid R \subseteq \{0, 1, \ldots, m-1\} \;\&\; m-1 \in R\},
\end{aligned}$$

and for $R \subseteq Q_M$ and $a \in \Sigma$,

$$\delta_{M'}(s_{M'}, a) = \{\delta_M(0, a)\},$$

$$\delta_{M'}(R,a) = \begin{cases} \{\delta_M(R,a)\}, & \text{if } m-1 \notin \delta_M(R,a); \\ \{\delta_M(R,a)\} \cup \{0\}, & \text{otherwise.} \end{cases}$$

Then we construct the DFA $A = (Q, \Sigma, \delta, s, F)$ that accepts $L(M)^* \cup L(N)$ exactly as described in the proof of Theorem 5.25, where

$$\begin{aligned} &s = \langle s_{M'}, 0\rangle, \\ &Q = \{\langle i,j\rangle \mid i \in Q_{M'} - \{ \quad s_{M'}\}, j \in Q_N\} \cup \{s\}, \\ &\delta(\langle i,j\rangle, a) = \langle \delta_{M'}(i,a), \delta_N(j,a)\rangle,\ \langle i,j\rangle \in Q,\ a \in \Sigma, \\ &F = \{\langle i,j\rangle \mid i \in F_{M'} \text{ or } j = n-1\}. \end{aligned}$$

Now we need to show that A is a minimal DFA.

(I) All the states in Q are reachable.
For an arbitrary state $\langle i,j\rangle$ in Q, there always exists a string w_1w_2 such that $\delta(\langle s'_M, 0\rangle, w_1w_2) = \langle i,j\rangle$, where

$$\begin{aligned} &\delta_{M'}(s_{M'}, w_1) = i,\ w_1 \in \{a,b\}^*, \\ &\delta_N(0, w_2) = j,\ w_2 \in \{c\}^*. \end{aligned}$$

(II) Any two different states $\langle i_1, j_1\rangle$ and $\langle i_2, j_2\rangle$ in Q are distinguishable.

1. $i_1 \neq i_2$, $j_2 \neq n-1$. There exists a string w_1 such that

$$\begin{aligned} &\delta(\langle i_1, j_1\rangle, w_1) \in F, \\ &\delta(\langle i_2, j_2\rangle, w_1) \notin F, \end{aligned}$$

where $w_1 \in \{a,b\}^*$, $\delta_{M'}(i_1, w_1) \in F_{M'}$ and $\delta'_M(i_2, w_1) \notin F'_M$.

2. $i_1 \neq i_2$, $j_2 = n-1$. There exists a string w_1 such that

$$\begin{aligned} &\delta(\langle i_1, j_1\rangle, w_1c) \in F, \\ &\delta(\langle i_2, j_2\rangle, w_1c) \notin F, \end{aligned}$$

where $w_1 \in \{a,b\}^*$, $\delta_{M'}(i_1, w_1) \in F_{M'}$ and $\delta_{M'}(i_2, w_1) \notin F_{M'}$.

3. $i_1 = i_2 \notin F_{M'}$, $j_1 \neq j_2$. For this case, the string c^{n-1-j_1} distinguishes the two states, since $\delta(\langle i_1, j_1\rangle, c^{n-1-j_1}) \in F$ and $\delta(\langle i_2, j_2\rangle, c^{n-1-j_1}) \notin F$.

4. $i_1 = i_2 \in F_{M'}$, $j_1 \neq j_2$. The string $b^m c^{n-1-j_1}$ distinguishes them, because $\delta(\langle i_1, j_1 \rangle, b^m c^{n-1-j_1}) \in F$ and $\delta(\langle i_2, j_2 \rangle, b^m c^{n-1-j_1}) \notin F$.

Since all the states in A are reachable and distinguishable, the DFA A is minimal. Thus, any DFA that accepts $L(M)^* \cup L(N)$ has at least $\frac{3}{4}2^m \cdot n - n + 1$ states. *q.e.d.*

This result gives a lower bound for the state complexity of $L(M)^* \cup L(N)$. It coincides with the upper bound in Corollary 5.2. So we have the following theorem.

Theorem 5.27 *For integers $m \geq 2$, $n \geq 2$, $\frac{3}{4}2^m \cdot n - n + 1$ states are both sufficient and necessary in the worst case for a DFA to accept $L(M)^* \cup L(N)$, where M is an m-state DFA and N is an n-state DFA.*

5.3.2 State Complexity of $L_1^* \cap L_2$

Since the state complexity of intersection on regular languages is the same as that of union [111], the mathematical composition of the state complexities of star and intersection is also $\frac{3}{4}2^m$. In this subsection, we show that the state complexity of $L_1^* \cap L_2$ is $\frac{3}{4}2^m \cdot n - n + 1$ which is the same as the state complexity of $L_1^* \cup L_2$ [33].

Theorem 5.28 *For any m-state DFA $M = (Q_M, \Sigma, \delta_M, s_M, F_M)$ and n-state DFA $N = (Q_N, \Sigma, \delta_N, s_N, F_N)$ such that $|F_M - \{s_M\}| = k \geq 1$, $m > 1$, $n > 0$, there exists a DFA of at most $(2^{m-1} + 2^{m-k-1}) \cdot n - n + 1$ states that accepts $L(M)^* \cap L(N)$.*

Proof: We construct the DFA A for $L(M)^* \cap L(N)$ which is the same as in the proof of Theorem 5.25, except that its set of final states is

$$F = \{\langle i, j \rangle \mid i \in F_{M'}, j \in F_N\}.$$

Thus, after removing the $n - 1$ unreachable states $\langle s_{M'}, j \rangle \notin Q$, for $j \in Q_N - \{s_N\}$, the number of states of A is sill no more than $(2^{m-1} + 2^{m-k-1}) \cdot n - n + 1$. *q.e.d.*

Similarly to the proof of Corollary 5.2, we consider both the case when M has no other final state except s_M ($L(M)^* = L(M)$) and the case when

M has some other final states (Theorem 5.28). Then we obtain the following corollary.

Corollary 5.3 *For any m-state DFA $M = (Q_M, \Sigma, \delta_M, s_M, F_M)$ and n-state DFA $N = (Q_N, \Sigma, \delta_N, s_N, F_N)$, $m > 1$, $n > 0$, there exists a DFA A of at most $\frac{3}{4}2^m \cdot n - n + 1$ states such that $L(A) = L(M)^* \cap L(N)$.*

Next, we show that this general upper bound on the state complexity of $L(M)^* \cap L(N)$ can be attained by some witness DFAs.

Theorem 5.29 *Given two integers $m \geq 2$, $n \geq 2$, there exists a DFA M of m states and a DFA N of n states such that any DFA that accepts $L(M)^* \cap L(N)$ has at least $\frac{3}{4}2^m \cdot n - n + 1$ states.*

Proof: We use the same DFAs M and N as in the proof of Theorem 5.26. Their transition diagrams are shown in Figure 5.25 and Figure 5.26, respectively. Construct the DFA $M' = (Q_{M'}, \Sigma, \delta_{M'}, s_{M'}, F_{M'})$ that accepts $L(M)^*$ in the same way.

Then we construct the DFA $A = (Q, \Sigma, \delta, s, F)$ that accepts $L(M)^* \cap L(N)$ exactly as described in the proof of Theorem 5.26 except that

$$F = \{\langle i, n-1 \rangle \mid i \in F_{M'}\}.$$

Now we prove that A is minimal.

(I) Every state of A is reachable.

Let $\langle i, j \rangle$ be an arbitrary state of A. Then there always exists a string $w_1 w_2$ such that $\delta(\langle s_{M'}, 0 \rangle, w_1 w_2) = \langle i, j \rangle$, where

$$\delta_{M'}(s_{M'}, w_1) = i,\ w_1 \in \{a, b\}^*,$$
$$\delta_N(0, w_2) = j,\ w_2 \in \{c\}^*.$$

(II) Any two different states $\langle i_1, j_1 \rangle$ and $\langle i_2, j_2 \rangle$ of A are distinguishable.

1. $i_1 \neq i_2$.

   We can find a string w_1 such that

   $$\delta(\langle i_1, j_1 \rangle, w_1 c^{n-1-j_1}) \in F,$$
   $$\delta(\langle i_2, j_2 \rangle, w_1 c^{n-1-j_1}) \notin F,$$

   where $w_1 \in \{a, b\}^*$, $\delta_{M'}(i_1, w_1) \in F_{M'}$ and $\delta_{M'}(i_2, w_1) \notin F_{M'}$.

2. $i_1 = i_2 \notin F_{M'}$, $j_1 \neq j_2$.

   There exists a string w_2 such that

$$\delta(\langle i_1, j_1\rangle, w_2 c^{n-1-j_1}) \in F,$$
$$\delta(\langle i_2, j_2\rangle, w_2 c^{n-1-j_1}) \notin F,$$

   where $w_2 \in \{a, b\}^*$ and $\delta_{M'}(i_1, w_2) \in F_{M'}$.

3. $i_1 = i_2 \in F_{M'}$, $j_1 \neq j_2$.

$$\delta(\langle i_1, j_1\rangle, c^{n-1-j_1}) \in F,$$
$$\delta(\langle i_2, j_2\rangle, c^{n-1-j_1}) \notin F.$$

From (I) and (II), A is a minimal DFA with $\frac{3}{4}2^m \cdot n - n + 1$ states which accepts $L(M)^* \cap L(N)$. *q.e.d.*

This lower bound coincides with the upper bound in Corollary 5.3. Thus, the bounds are tight.

Theorem 5.30 *For integers $m \geq 2$, $n \geq 2$, $\frac{3}{4}2^m \cdot n - n + 1$ states are both sufficient and necessary in the worst case for a DFA to accept $L(M)^* \cap L(N)$, where M is an m-state DFA and N is an n-state DFA.*

5.3.3 State Complexity of $L_1^R \cup L_2$

In this subsection, we study the state complexity of $L_1^R \cup L_2$, where L_1 and L_2 are regular languages [33]. It has been proved that the state complexity of L_1^R is 2^m and the state complexity of $L_1 \cup L_2$ is mn [72, 111]. Thus, the composition of these two expressions is $2^m \cdot n$. In this subsection we will prove that this upper bound on state complexity of $L_1^R \cup L_2$ cannot be attained in any case. We will first decrease the upper bound in the following.

Theorem 5.31 *Let L_1 and L_2 be two regular languages accepted by an m-state and n-state DFAs, respectively. Then there exists a DFA of at most $2^m \cdot n - n + 1$ states that accepts $L_1^R \cup L_2$.*

Proof: Let $M = (Q_M, \Sigma, \delta_M, s_M, F_M)$ be a complete DFA of m states and $L_1 = L(M)$. Let $N = (Q_N, \Sigma, \delta_N, s_N, F_N)$ be another complete 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. $\delta_{M'}(p, a) = q$ if $\delta_M(q, a) = p$ where $a \in \Sigma$ and $p, q \in Q_M$. Clearly, $L(M') = L(M)^R = L_1^R$. After performing the subset construction, we get a 2^m-state DFA $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ that is equivalent to M'. Since A has 2^m states, one of its final states must be Q_M. Now we construct the DFA $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$, where

$$Q_B = \{\langle i, j\rangle \mid i \in Q_A,\ j \in Q_N\},$$

$$s_B = \langle s_A, s_N\rangle,$$

$$F_B = \{\langle i, j\rangle \in Q_B \mid i \in F_A \text{ or } j \in F_N\},$$

$$\delta_B(\langle i, j\rangle, a) = \langle i', j'\rangle, \text{ if } \delta_A(i, a) = i' \text{ and } \delta_N(j, a) = j',\ a \in \Sigma.$$

It is easy to see that $\delta_B(\langle Q_M, j\rangle, a) \in F_B$ for any $j \in Q_N$ and $a \in \Sigma$. This means all the states (two-tuples) starting with Q_1 are equivalent. There are n such states in total. Thus, the minimal DFA that accepts $L_1^R \cup L_2$ has no more than $2^m \cdot n - n + 1$ states. *q.e.d.*

This result gives an upper bound for the state complexity of $L_1^R \cup L_2$. Now let's see if this bound is attainable.

Theorem 5.32 *Given two integers $m \geq 2$, $n \geq 2$, there exists a DFA M of m states and a DFA N of n states such that any DFA that accepts $L(M)^R \cup L(N)$ has at least $2^m \cdot n - n + 1$ states.*

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

$$\delta_M(0, a) = m - 1,\ \delta_M(i, a) = i - 1,\ i = 1, \ldots, m - 1,$$

$$\delta_M(0, b) = 1,\ \delta_M(i, b) = i,\ i = 1, \ldots, m - 1,$$

$$\delta_M(0, c) = 1,\ \delta_M(1, c) = 0,\ \delta_M(j, c) = i,\ j = 2, \ldots, m - 1,$$

$$\delta_M(k, d) = k,\ k = 0, \ldots, m - 1.$$

The transition diagram of M is shown in Figure 5.27. Let $N = (Q_N, \Sigma, \delta_N, 0, \{0\})$ be another DFA, where $Q_N = \{0, 1, \ldots, n - 1\}$, $\Sigma = \{a, b, c, d\}$ and the transitions are

$$\delta_N(i, a) = i,\ i = 0, \ldots, n - 1,$$

$$\delta_N(i, b) = i,\ i = 0, \ldots, n - 1,$$

$$\delta_N(i, c) = i,\ i = 0, \ldots, n - 1,$$

$$\delta_N(i, d) = i + 1 \bmod n,\ i = 0, \ldots, n - 1.$$

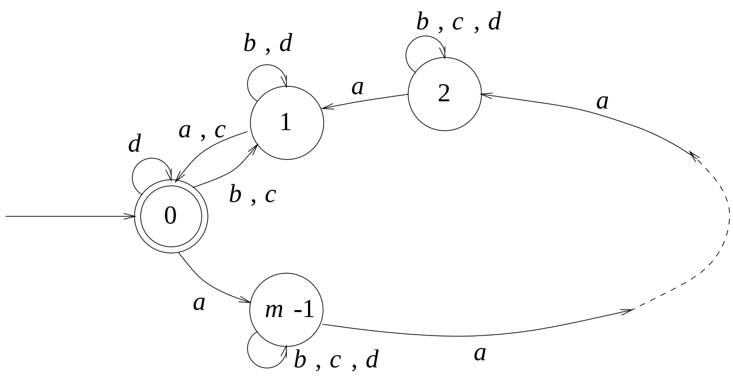

Figure 5.27: Witness DFA M of Theorem 5.32

The transition diagram of N is shown in Figure 5.28.

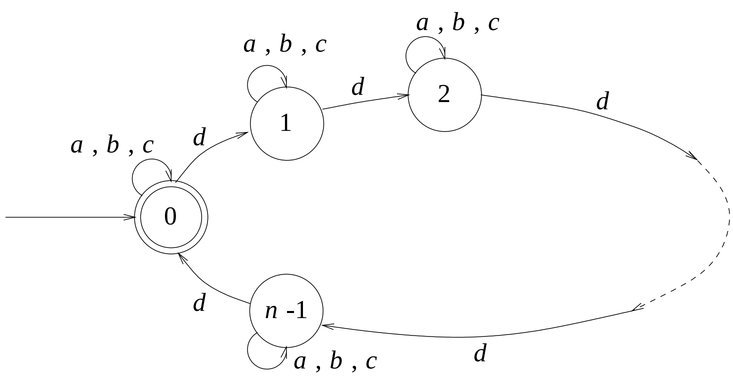

Figure 5.28: Witness DFA N of Theorem 5.32

Note that M is a modification of the worst-case example given in [111] for reversal, by adding a d-loop to every state. Intuitively, the minimal DFA that accepts $L(M)^R$ should also have 2^m states. Before using this result, we will prove it first. Let $A = (Q_A, \Sigma, \delta_A, \{0\}, F_A)$ be a DFA, where

$$Q_A = \{q \mid q \subseteq Q_M\},$$
$$\Sigma = \{a, b, c, d\},$$
$$\delta_A(p, e) = \{j \mid \delta_M(i, e) = j,\ i \in p\},\ p \in Q_A,\ e \in \Sigma,$$
$$F_A = \{q \mid \{0\} \in q,\ q \in Q_A\}.$$

Clearly, A has 2^m states and it accepts $L(M)^R$. Now let's prove A is minimal.

(i) Every state $i \in Q_A$ is reachable.

1. $i = \emptyset$.
   $|i| = 0$ if and only if $i = \emptyset$. $\delta_A(\{0\}, b) = i = \emptyset$.
2. $|i| = 1$.
   Assume that $i = \{p\}$, $0 \le p \le m-1$. $\delta_A(\{0\}, a^p) = i$.
3. $2 \le |i| \le m$.
   Assume that $i = \{i_1, i_2, \ldots, i_k\}$, $0 \le i_1 < i_2 < \cdots < i_k \le m-1$, $2 \le k \le m$. $\delta_A(\{0\}, w) = i$, where

$$w = ab(ac)^{i_k - i_{k-1} - 1} ab(ac)^{i_{k-1} - i_{k-2} - 1} \cdots ab(ac)^{i_2 - i_1 - 1} a^{i_1}.$$

(ii) Any two different states i and j in Q_A are distinguishable.
Without loss of generality, we may assume that $|i| \ge |j|$. Let $x \in i - j$. Then the string a^{m-x} distinguishes these two states because

$$\begin{aligned} \delta_A(i, a^{m-x}) &\in F_A, \\ \delta_A(j, a^{m-x}) &\notin F_A. \end{aligned}$$

Thus, A is a minimal DFA with 2^m states that accepts $L(M)^R$. Now let $B = (Q_B, \Sigma, \delta_B, \{\langle\{0\}, 0\rangle\}, F_B)$ be a DFA, where

$$\begin{aligned} &Q_B = \{\langle p, q\rangle \mid p \in Q_A - \{Q_M\},\, q \in Q_N\} \cup \{\langle Q_M, 0\rangle\}, \\ &\Sigma = \{a, b, c, d\}, \\ &F_B = \{\langle p, q\rangle \mid p \in F_A \text{ or } q \in F_N,\, \langle p, q\rangle \in Q_B\}, \end{aligned}$$

and for $\langle p, q\rangle \in Q_B$, $e \in \Sigma$

$$\delta_B(\langle p, q\rangle, e) = \begin{cases} \langle p', q'\rangle, & \text{if } \delta_A(p, e) = p',\ \delta_N(q, e) = q',\ p' \ne Q_M; \\ \langle Q_M, 0\rangle, & \text{if } \delta_A(p, e) = Q_M. \end{cases}$$

As we mentioned in the previous proof, all the states (two-tuples) starting with Q_M are equivalent. Thus, we replace them with one state: $\langle Q_M, 0\rangle$. It is easy to see that B accepts the language $L(M)^R \cup L(N)$. It has $2^m \cdot n - n + 1$ states. Now let us prove that B is a minimal DFA.

(I) All the states in Q_B are reachable.
For an arbitrary state $\langle p, q\rangle$ in Q_B, there always exists a string $d^q w$ such that $\delta_B(\langle\{0\}, 0\rangle, d^q w) = \langle p, q\rangle$, where $w \in \{a, b, c\}^*$ and $\delta_A(\{0\}, w) = p$.

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

1. $q_1 = q_2$.

   We can easily find a string $d^i w$ such that

$$\begin{aligned} \delta_B(\langle p_1, q_1\rangle, d^i w) &\in F_B, \\ \delta_B(\langle p_2, q_2\rangle, d^i w) &\notin F_B, \end{aligned}$$

   where $(i + q_1) \bmod n \neq 0$, $w \in \{a, b, c\}^*$, $\delta_A(p_1, w) \in F_A$ and $\delta_A(p_2, w) \notin F_A$.

2. $p_1 = p_2$, $q_1 \neq q_2$.

   The string $d^{n-q_1}w$ distinguishes these two states where $w \in \{a, b, c\}^*$ and $\delta_A(p_1, w) \notin F_A$, because

$$\begin{aligned} \delta_B(\langle p_1, q_1\rangle, d^{n-q_1} w) &\in F_B, \\ \delta_B(\langle p_2, q_2\rangle, d^{n-q_1} w) &\notin F_B. \end{aligned}$$

3. $p_1 \neq p_2$, $q_1 \neq q_2$.

   We first find a string $w \in \{a, b, c\}^*$ such that $\delta_A(p_1, w) \in F_A$ and $\delta_A(p_2, w) \notin F_A$. Then it is clear that

$$\begin{aligned} \delta_B(\langle p_1, q_1\rangle, d^{n-q_1} w) &\in F_B, \\ \delta_B(\langle p_2, q_2\rangle, d^{n-q_1} w) &\notin F_B. \end{aligned}$$

Since all the states in B are reachable and distinguishable, the DFA B is minimal. Thus, any DFA that accepts $L(M)^R \cup L(N)$ has at least $2^m \cdot n - n + 1$ states. *q.e.d.*

This result gives a lower bound for the state complexity of $L(M)^R \cup L(N)$. It coincides with the upper bound. So we have the following theorem.

Theorem 5.33 *For integers $m \geq 2$, $n \geq 2$, $2^m \cdot n - n + 1$ states are both sufficient and necessary in the worst case for a DFA to accept $L(M)^R \cup L(N)$, where M is an m-state DFA and N is an n-state DFA.*

5.3.4 State Complexity of $L_1^R \cap L_2$

The state complexity of $L_1^R \cap L_2$ is the same as that of $L_1^R \cup L_2$, namely, $2^m \cdot n - n + 1$, since

$$L_1^R \cap L_2 = \overline{\overline{L_1^R} \cup \overline{L_2}} = \overline{\overline{L_1}^R \cup \overline{L_2}}$$

according to De Morgan's laws and $\overline{L^R} = \overline{L}^R$, where $\overline{L}$ denotes the complementation of L, and the state complexity of the complementation of an n-state DFA language is n.

5.4 State Complexity of Combined Boolean Operations

In this section, we will present and prove the state complexity of combined Boolean operations. All the results in this section have been published in our paper [27].

A combined Boolean operation in k operands (over languages over an alphabet) is a function $f(x_1, \ldots, x_k)$ which can be constructed from the projection functions and the binary union, intersection and the unary complementation operations by function composition [27]. In other words, there is an expression denoting f which is built from the variables $x_1, \ldots, x_k$ and the boolean operations of conjunction, disjunction and complementation. Each variable may be used any number of times. We say that such a combined operation f *depends* on its ith operand, for $i = 1, \ldots, k$, if there exist languages $L_1, \ldots, L_k$ and L'_i such that $f(L_1, \ldots, L_k) \neq f(L_1, \ldots, L_{i-1}, L'_i, L_{i+1}, \ldots, L_k)$. Any combined Boolean operation $f(x_1, \ldots, x_k)$ may be viewed as a Boolean function on truth values. It is clear that f depends on its ith operand iff there exist $c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_k$ in $\{0, 1\}$ such that, as a Boolean function on truth values, it satisfies $f(c_1, \ldots, c_{i-1}, 0, c_{i+1}, \ldots, c_k) \neq f(c_1, \ldots, c_{i-1}, 1, c_{i+1}, \ldots, c_k)$. For example, $x_1 \cup (x_1 \cap x_2)$ depends on its first operand, but does not depend on its second. However, if there is an expression for f containing exactly one occurrence of each x_i, $i = 1, \ldots, k$, then f depends on each of its operands [27].

Theorem 5.34 *Let f be a combined Boolean operation in k operands. Suppose that f depends on each of its operands. Then for all integers $n_1, \ldots, n_k$ greater than 1, the state complexity of f is $n_1 \cdots n_k$, where for each i, n_i denotes the state complexity of the ith regular operand language.*

Proof: It is clear that $n_1 \cdots n_k$ is an upper bound. To prove that it is also a lower bound, we construct an example that attains the bound. For this reason, consider regular languages R_i of state complexity n_i over pairwise disjoint alphabets Σ_i, $i = 1, \ldots, k$. In our argument, we will need the additional property that for each R_i and for any two not necessarily different left quotients $u^{-1}R_i$ and $v^{-1}R_i$ there is a string x such that both $\varepsilon \in x^{-1}u^{-1}R_i$ and $\varepsilon \in x^{-1}v^{-1}R_i$, and symmetrically, there is a string y such that neither $\varepsilon \in y^{-1}u^{-1}R_i$ nor $\varepsilon \in y^{-1}v^{-1}R_i$. For the minimal automaton of R_i this means that for any two not necessarily different states q and q' there is a string x which brings both q and q' to a final state, and there is also a string y which brings neither of them to a final state. By minimality, we also know that when q and q' are different, then there is a string z which brings exactly one of q and q' to a final state. For example, we may define R_i as the set of all strings over the two-letter alphabet $\{a_i, b_i\}$, ending in at least $n_i - 1$ occurrences of the letter a_i, i.e., $R_i = (a_i \cup b_i)^* a_i^{n_i - 1}$.

Let Σ be the union of the Σ_i, and for each i, let $R'_i = h_i^{-1}(R_i)$, where $h_i : \Sigma^* \to \Sigma_i$ is the homomorphism which is the identity function on Σ_i and maps any other letter to the empty word. Then each R'_i is a regular language of state complexity n_i over the alphabet Σ. Indeed, the minimal automaton $A'_i = (Q_i, \Sigma, \delta'_i, s_i, F_i)$ for R'_i can be constructed from the minimal automaton $A_i = (Q_i, \Sigma_i, \delta_i, s_i, F_i)$ for R_i by adding a transition from any state to itself under any letter in $\Sigma - \Sigma_i$. We show that the minimal automaton for $R' = f(R'_1, \ldots, R'_k)$ is the (usual) direct product

$$A' = (Q = Q_1 \times \cdots \times Q_k, \Sigma, \delta', s = (s_1, \ldots, s_k), F)$$

of the A'_i with set of final states $F = \{q \in Q : \exists u \in R' \ \delta'(s, u) = q\}$. Thus, $\delta'((q_1, \ldots, q_k), a) = (\delta'_1(q_1, a), \ldots, \delta'_k(q_k, a))$ for all $(q_1, \ldots, q_k) \in Q_1 \times \cdots \times Q_k$ and $a \in \Sigma$.

First we show that $L(A') = R'$. It is clear that $R' \subseteq L(A')$. Suppose now that $u \in L(A')$. Then there is a string $v \in R'$ with $\delta'(s, v) = \delta'(s, u)$, so that $\delta'_i(s_i, v) = \delta'_i(s_i, u)$ for all i. But this implies that for all i, $v \in R'_i$ iff $u \in R'_i$. Thus, since $v \in R'$, it follows that $u \in R'$.

Now each state in Q is accessible from the initial state s. Indeed, given a k-tuple $q = (q_1, \ldots, q_k)$, we can choose strings $u_i \in \Sigma_i^*$ with $\delta_i(s_i, u_i) = q_i$, for all i. Then let $u = u_1 \cdots u_k$. We have that $\delta'(s, u) = q$.

So to complete the proof of the fact that A' is the minimal automaton for R', we have to show that for any two different tuples $q = (q_1, \ldots, q_k)$ and $q' = (q'_1, \ldots, q'_k)$ there is a string $v \in \Sigma^*$ such that exactly one of $\delta'(q, v)$ and $\delta'(q', v)$

is in F. Since q is different from q', there is some i_0 with $q_{i_0} \neq q'_{i_0}$. Let us choose strings $u, u' \in \Sigma^*$ with $\delta'(s,u) = q$ and $\delta'(s,u') = q'$. For each i, let $u_i = h_i(u)$ and $u'_i = h_i(u')$. By the minimality of A_{i_0}, there exists a string $v_{i_0} \in \Sigma^*_{i_0}$ such that exactly one of the states $\delta_{i_0}(q_{i_0}, v_{i_0})$ and $\delta_{i_0}(q'_{i_0}, v_{i_0})$ is in F_{i_0}. Since f depends on each of its arguments, for some bits $c_1, \ldots, c_{i_0-1}, c_{i_0+1}, \ldots, c_k$ in $\{0,1\}$ we have that $f(c_1, \ldots, c_{i_0-1}, 0, c_{i_0+1}, \ldots, c_k) \neq f(c_1, \ldots, c_{i_0-1}, 1, c_{i_0+1}, \ldots, c_k)$. Now, for each $i \neq i_0$, by our assumption on the language R_i, we can select a string $v_i \in \Sigma_i^*$ with $\delta_i(s_i, u_i v_i)$, $\delta_i(s_i, u'_i v_i) \in F_i$ if $c_i = 1$ and $\delta_i(s_i, u_i v_i)$,$\delta_i(s_i, u'_i v_i) \notin F_i$ if $c_i = 0$. Thus, if $c_i = 1$, then both $u_i v_i$ and $u'_i v_i$ are in R_i and if $c_i = 0$, then neither of these strings is in R_i. Then let $v = v_1 \cdots v_k$ and consider the strings uv and $u'v$. It is clear that exactly one of them is in R'. Since $\delta'(s,u) = q$ and $\delta'(s,u') = q'$, and since $L(A) = R'$, this means that exactly one of $\delta'(q,v)$ and $\delta'(q',v)$ is in F. *q.e.d.*

Remark 5.35 *The above proof shows that the upper bound $n_1 \cdots n_k$ can be attained over an alphabet of size $2k$. We conjecture that it cannot be attained in general over an alphabet of a fixed size. The proof also shows that the bound $n_1 \cdots n_k$ can be attained by the same regular languages $R_1, \ldots, R_k$ for all combined Boolean operations which depend on k operands.*

Example 5.2 Let $f(x,y)$ be the "equivalence function" $(x \cap y) \cup (\overline{x} \cap \overline{y})$ which depends on both of its operands. When R is the set of all strings over $\{a,b\}$ with an even number of occurrences of letter a and S is the set of all strings with an even number of occurrences of letter b, then both R and S have state complexity two. Now $f(R,S)$ is the set of all strings over $\{a,b\}$ of even length, which also has state complexity two. So this example shows that the state complexity of $f(R,S)$ may be smaller than the product of the state complexities of the operand languages R, S. $\square$

Although we conjecture that $n_1 \cdots n_k$ cannot be attained in general for all the combined Boolean operations on languages over an alphabet of a fixed size, we show that the bound can be attained in infinitely many cases. We have the following results. The first is a case over a one-letter alphabet. The next two cases are over a two-letter alphabet. Note that the following results only involve intersections. In the following, gcd and lcm stand for greatest common divisor and least common multiple, respectively.

Theorem 5.36 *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 $1 \leq i < j \leq k$. 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 $n_1 \cdots n_k$.*

We only give a brief proof of Theorem 5.36 here. Consider languages $R_i = \{a^{n_i}\}^*$ of state complexity n_i. Then $R_1 \cap \cdots \cap R_k = \{a^{\mathrm{lcm}(n_1,\ldots,n_k)}\}^*$. Since $n_1, \ldots, n_k$ are mutually prime, $\mathrm{lcm}(n_1, \ldots, n_k) = n_1 \cdots n_k$.

Although this result is about languages over a one-letter alphabet, it clearly holds on languages over an alphabet of any positive size.

Theorem 5.37 *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, $n_1, \ldots, n_k > 0$. 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 $n_1, \ldots, n_l$ are mutually prime and $n_{l+1}, \ldots, n_k$ are also 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 $\{n_1, n_2, \ldots, n_k\}$, $n_i > 0$, $1 \leq i \leq k$, 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 the DFA $A_i = (Q_i, \{a, b\}, \delta_i, 0, \{0\})$, where $Q_i = \{0, \ldots, n_i - 1\}$ and δ_i is given by

$$\begin{aligned}
\delta_i(t, a) &= t + 1 \bmod n_i,\ t = 0, 1, \ldots, n_i - 1,\\
\delta_i(t, b) &= t,\ t = 0, 1, \ldots, n_i - 1.
\end{aligned}$$

We denote $L(A_i)$ by R_i.

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

$$\begin{aligned}
\delta_p(t, b) &= t + 1 \bmod n_p,\ t = 0, 1, \ldots, n_p - 1,\\
\delta_p(t, a) &= t,\ t = 0, 1, \ldots, n_p - 1.
\end{aligned}$$

We denote $L(A_p)$ by R_p.

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

$$Q_C = \{0, 1, \ldots, \prod_{n_e \in M} n_e - 1\},$$

$$\delta_C(t, a) = t + 1 \bmod \prod_{n_e \in M} n_e,\ t = 0, 1, \ldots, \prod_{n_e \in M} n_e - 1,$$

$$\delta_C(t, b) = t,\ t = 0, 1, \ldots, \prod_{n_e \in M} n_e - 1.$$

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

$$Q_D = \{0, 1, \ldots, \prod_{n_g \in N} n_g - 1\},$$

$$\delta_D(t, b) = t + 1 \bmod \prod_{n_g \in N} n_g,\ t = 0, 1, \ldots, \prod_{n_g \in N} n_g - 1,$$

$$\delta_D(t, a) = t,\ t = 0, 1, \ldots, \prod_{n_g \in N} n_g - 1.$$

Now we have

$$\begin{aligned} L(C) &= \{w \mid w \in \{a, b\}^*,\ |w|_a \bmod \prod_{n_e \in M} n_e = 0\}, \\ L(D) &= \{w \mid w \in \{a, b\}^*,\ |w|_b \bmod \prod_{n_g \in N} n_g = 0\}. \end{aligned}$$

Clearly, we have

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

Let $E = (Q_E, \{a, b\}, \delta_E, \langle 0, 0\rangle, \{\langle 0, 0\rangle\})$ be a DFA, where

$$Q_E = \{\langle X, Y\rangle \mid X \in Q_C, Y \in Q_D\},$$

$$\delta_E(\langle X, Y\rangle, a) = \langle \delta_C(X, a), \delta_D(Y, a)\rangle,$$

$$\delta_E(\langle X, Y\rangle, b) = \langle \delta_C(X, b), \delta_D(Y, b)\rangle.$$

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 $\langle X, Y\rangle \in Q_E$, $\delta_E(\langle 0, 0\rangle, a^X b^Y) = \langle X, Y\rangle$. So every state in Q_E is reachable.

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

$$\begin{aligned} \delta_E(\langle X_1, Y_1\rangle, a^{|Q_C| - X_1} b^{|Q_D| - Y_1}) &= \langle 0, 0\rangle, \\ \delta_E(\langle X_2, Y_2\rangle, a^{|Q_C| - X_1} b^{|Q_D| - Y_1}) &\neq \langle 0, 0\rangle. \end{aligned}$$

   So any two distinct states of E are not equivalent.

Thus, E is the minimal DFA that accepts $R_1 \cap R_2 \cap \cdots \cap R_k$. *q.e.d.*

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

Corollary 5.4 *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 5.37 is stated in the following.

Theorem 5.38 *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 $n_1, \ldots, n_{k+1}$ states, respectively, $n_1, \ldots, n_k \geq 1$, $n_{k+1} \geq 3$. 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 easy to see 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 as that of Theorem 5.37. Assume that a set of integers $\{n_1, n_2, \ldots, n_k\}$, $n_i \geq 1$, $1 \leq i \leq k$, can be divided into two sets M and N such that both of them are mutually prime, i.e., $\gcd(n_e, n_f) = 1$ for any $n_e, n_f \in M$, $e \neq f$, and $\gcd(n_g, n_h) = 1$ for any $n_g, n_h \in N$, $g \neq h$. Then construct the DFA C that accepts the intersection of all R_e for $n_e \in M$ and the DFA D that accepts the intersection of all R_g for $n_g \in N$. Let Q_C and Q_D be the state sets of C and D, respectively, and $u = |Q_C|$ and $v = |Q_D|$.

Let n_{k+1} be an arbitrary integer such that $n_{k+1} \geq 3$. Define an n_{k+1}-state DFA $F = \{Q_F, \{a, b\}, \delta_F, 0, \{0\}\}$ where

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

We denote $L(F)$ by R_{k+1}. Let $G = \{Q_G, \{a,b\}, \delta_G, q_0, F_G\}$ be a DFA, where

$Q_G = \{\langle X, Y, Z\rangle \mid X \in Q_C, Y \in Q_D, Z \in Q_F\}$,

$q_0 = \langle 0, 0, 0\rangle$,

$F_G = \{\langle 0, 0, 0\rangle\}$,

$\delta_G(\langle X, Y, Z\rangle, a) = \langle \delta_C(X, a), \delta_D(Y, a), \delta_F(Z, a)\rangle$,

$\delta_G(\langle X, Y, Z\rangle, b) = \langle \delta_C(X, b), \delta_D(Y, b), \delta_F(Z, b)\rangle$, for each $\langle X, Y, Z\rangle \in Q_G$.

It is easy to see that $L(G) = L(C) \cap L(D) \cap R_{k+1}$.

Now we check if G is a minimal DFA.

1. For any state $\langle X, Y, Z\rangle \in Q_G$, $Z \neq 0, 1, 2$,

$$\delta_G(\langle 0, 0, 0\rangle, a^{n_{k+1}+T} b^{v+Y} a^{Z-2}) = \langle X, Y, Z\rangle$$

   where T is a positive integer such that $(n_{k+1} + T + Z - 2) \equiv X \pmod u$. For $\langle X, Y, Z\rangle \in Q_G$, $Z = 0$ or 1 or 2,

$$\delta_G(\langle 0, 0, 0\rangle, a^T b^{n_{k+1}v+Y-Z} a^{n_{k+1}-2} b^Z) = \langle X, Y, Z\rangle$$

   where T is a positive integer such that $(n_{k+1} + T - 2) \equiv X \pmod u$. So every state in Q_G is reachable.

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

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

$$\delta_G(\langle X_1, Y_1, Z_1\rangle, a^{n_{k+1}+T} b^{2v-Y_1} a^{n_{k+1}-2}) = \langle 0, 0, 0\rangle,$$
$$\delta_G(\langle X_2, Y_2, Z_2\rangle, a^{n_{k+1}+T} b^{2v-Y_1} a^{n_{k+1}-2}) \neq \langle 0, 0, 0\rangle,$$

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

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

   (I) $Z_1 \geq 0$, $Z_2 > 2$, $Z_2 > Z_1$
   Let $t_1 = b^{2v-Y_1-1} a^{n_{k+1}-Z_2} b a^{n_{k+1}+T}$, where T is a positive integer such that $(2n_{k+1} - Z_2 + T) \equiv u - X_1 \pmod u$. Then

$$\begin{aligned}\delta_G(\langle X_1, Y_1, Z_1\rangle, t_1) &= \langle 0, 0, 0\rangle,\\ \delta_G(\langle X_2, Y_2, Z_2\rangle, t_1) &\neq \langle 0, 0, 0\rangle,\end{aligned}$$

(II) $Z_1 > 2$, $Z_2 \geq 0$, $Z_1 > Z_2$
It is symmetric to (I), let $t_1' = b^{2v-Y_2-1}a^{n_{k+1}-Z_1}ba^{n_{k+1}+T}$, where T is a positive integer such that $(2n_{k+1} - Z_1 + T) \equiv u - X_1 \pmod{u}$. In this case, t_1' distinguishes the two states.

(III) $Z_1 = 0$, $Z_2 = 1$ or 2
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 $(T + n_{k+1}(2v - Y_1)) \equiv u - X_1 \pmod{u}$. Then one of $\delta_G(\langle X_1, Y_1, 0\rangle, t_2)$ and $\delta_G(\langle X_2, Y_2, Z_2\rangle, t_2)$ is $\langle 0, 0, 0\rangle$ but the other is not.

(IV) $Z_1 = 1$ or 2, $Z_2 = 0$
It is symmetric to (III). The string t_2 also works for distinguishing the two states.

(V) $Z_1 = 1$, $Z_2 = 2$
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 $(T + n_{k+1}(2v - Y_1 + 1)) \equiv u - X_1 \pmod{u}$. Then one of $\delta_G(\langle X_1, Y_1, 1\rangle, t_3)$ and $\delta_G(\langle X_2, Y_2, 2\rangle, t_3)$ is $\langle 0, 0, 0\rangle$ but the other is not.

(VI) $Z_1 = 2$, $Z_2 = 1$
It is symmetric to (V). The string t_3 also works for distinguishing the two states.

So any two states of G are distinguishable.

Thus, G is the minimal DFA for $R_1 \cap R_2 \cap \cdots \cap R_k \cap R_{k+1}$ that has $n_1 \cdot n_2 \cdot \cdots \cdot n_k \cdot n_{k+1}$ states. *q.e.d.*

5.5 State Complexity of Multiple Catenations

5.5.1 State Complexity of $L_1L_2L_3$

In this subsection, we study the state complexity of $L_1L_2L_3$, where L_1, L_2 and L_3 are three regular languages accepted by DFAs of m, n and p states, respectively. All the results in this subsection have been published in our paper [27].

The direct composition of the state complexity of the catenation of L_1, L_2 and L_3 is $m2^{n+p} - 2^{n+p-1} - 2^{p-1}$ which is an upper bound for the state complexity of $L_1L_2L_3$ but cannot be attained [27, 29].

Theorem 5.39 *For integers $m, n, p \geq 2$, there exist DFAs A, B, and C of m, n, and p states, respectively, such that any DFA that accepts $L(A)L(B)L(C)$ has 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 $\Sigma = \{a, b, c, d, e\}$. Let $A = (Q_A, \Sigma, \delta_A, 0, \{m-1\})$ be a DFA, where $Q_A = \{0, \ldots, m-1\}$ and δ_A is defined as follows. For the state $t = 0, 1, \ldots, m-1$, $\delta_A(t, a) = t+1 \bmod m$, $\delta_A(t, x) = t, x \in \{b, c, e\}$ and $\delta_A(t, d) = 0$. Let $B = (Q_B, \Sigma, \delta_B, 0, \{n-1\})$ be a DFA, where $Q_B = \{0, \ldots, n-1\}$ and δ_B is defined as follows. For the state $t = 0, 1, \ldots, n-1$, $\delta_B(t, b) = t + 1 \bmod n$, $\delta_B(t, y) = t, y \in \{a, d, e\}$ and $\delta_B(t, c) = 1$. Let $C = (Q_C, \Sigma, \delta_C, 0, \{p-1\})$ be a DFA, where $Q_C = \{0, \ldots, p-1\}$. For the state $t = 0, 1, \ldots, p-1$, $\delta_C(t, d) = t + 1 \bmod p$, $\delta_C(t, z) = t, z \in \{a, b, c\}$ and $\delta_C(t, e) = 1$.

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

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

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

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

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

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

Consider $u, v \in \{a, b, d\}^*$ such that $R(u) \bmod m > R(v) \bmod m$. 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 string w such that $0 \notin T(w)$ and $R(w) = m-1$, since $R(w) = m-1$ guarantees that $0 \in T(w)$. For the same reason, there does not exist a string w 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 $s = \{i_1, \ldots, i_k\}$ of $\{0, \ldots, p-1\}$ and each subset $t = \{j_1, \ldots, j_l\}$ of $\{0, \ldots, n-1\}$ where $i_1 > \cdots > i_k$ and $j_1 > \cdots > j_l$, and an integer $r \in \{0, \ldots, m-1\}$, 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 string

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

such that $S(x) = s$, $T(x) = t$ and $R(x) = r$. In total, 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. *q.e.d.*

We now show an upper bound for this combined operation.

Theorem 5.40 *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 $0 < l < n$. Then there exists a DFA of at most $(2m-k)2^{n+p-2} + (2m-k)2^{n+p-l-2} - (m-k)(2^p-1)$ 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 $C = (Q_C, \Sigma, \delta_C, t_0, F_C)$ be three DFAs. Construct the DFA $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ such that

$$\begin{aligned} Q_E \quad = \quad & Q_A \times 2^{Q_B} \times 2^{Q_C} - F_A \times 2^{Q_B - \{s_0\}} \times 2^{Q_C} \\ & -(Q_A - F_A) \times ((2^{F_B} - \{\emptyset\}) \cup 2^{Q_B - F_B}) \times 2^{Q_C - \{t_0\}} \\ & -F_A \times ((2^{F_B} - \{\emptyset\}) \cup 2^{Q_B - F_B - \{s_0\}}) \times 2^{Q_C - \{t_0\}} \\ & -(Q_A - F_A) \times \{\emptyset\} \times (2^{Q_C} - \{\emptyset\}), \end{aligned}$$

$$q_0 = \begin{cases} \langle r_0, \emptyset, \emptyset \rangle, & \text{if } r_0 \notin F_A \text{ and } s_0 \notin F_B; \\ \langle r_0, \{s_0\}, \emptyset \rangle, & \text{if } r_0 \in F_A \text{ and } s_0 \notin F_B; \\ \langle r_0, \{s_0\}, \{t_0\} \rangle, & \text{if } r_0 \in F_A \text{ and } s_0 \in F_B, \end{cases}$$

$$\begin{aligned} F_E &= \{\langle r, S, T \rangle \in Q_E \mid T \cap F_C \neq \emptyset\}, \\ \delta_E(\langle r, S, T \rangle, a) &= \langle r', S', T' \rangle, \text{ for } a \in \Sigma, \text{ where } r' = \delta_A(r, a), \\ S' &= \begin{cases} \delta_B(S, a) \cup \{s_0\}, & \text{if } r' \in F_A; \\ \delta_B(S, a), & \text{otherwise}, \end{cases} \\ T' &= \begin{cases} \delta_C(T, a) \cup \{t_0\}, & \text{if } S' \cap F_B \neq \emptyset; \\ \delta_C(T, a), & \text{otherwise}. \end{cases} \end{aligned}$$

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

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

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

Finally, Q_E also does not contain the three-tuples whose first component is a non-final state of A, second component is $\emptyset$, and 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. *q.e.d.*

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 5.39. Thus, this bound is the state complexity of the catenation of three regular languages.

5.5.2 State Complexity of $L_1L_2\cdots L_k$

In this subsection, we prove the exact state complexities of the catenation of k regular languages for arbitrary $k \geq 2$. All the results shown in this subsection are from our paper [32]. We first consider a lower bound on the state complexity of $L_1L_2\cdots L_k$.

Theorem 5.41 *For integers $n_i \geq 2$, $1 \leq i \leq k$, there exist DFAs A_i of n_i states, respectively, such that any DFA that accepts $L(A_1)\cdots L(A_k)$ has at least*

$$n_1 2^{n_2+\cdots+n_k} - D - \sum_{h=1}^{k-1} E_h$$

states, where

$$\begin{aligned}
D &= \sum_{j=1}^{k-2}(n_1 \cdot (\prod_{r=2}^{j}(2^{n_r}-1)) \cdot (2^{\sum_{q=j+2}^{k} n_q}-1)); \\
E_h &= \sum_{\alpha=0}^{2^{h-1}-1}((\prod_{\beta=1}^{h} G_{\alpha,\beta}) \cdot (1+(2^{n_{h+1}-1}-1)\cdot R_{h,1})); \\
R_{h,\mu} &= (1+(2^{n_{\mu+h+1}}-1)\cdot R_{h,\mu+1}) \text{ for } 1 \leq \mu \leq k-h-2; \\
R_{h,k-h-1} &= 2^{n_k};
\end{aligned}$$

and for $w_\gamma \in \{0,1\}$, $1 \leq \gamma \leq h-1$ such that $w_1w_2w_3\cdots w_{h-1}$ is a binary number whose length is $h-1$ and value is α,

$$G_{\alpha,1} = \begin{cases} n_1 - 1, \text{ if } w_1 = 0 \text{ and } h \geq 2; \\ 1, \text{ if } w_1 = 1 \text{ and } h \geq 2, \end{cases}$$

$$\text{for } 2 \leq \theta \leq h-1,\ G_{\alpha,\theta} = \begin{cases} 2^{n_\theta - 1} - 1, \text{ if } w_{\theta-1} = 0 \text{ and } w_\theta = 0; \\ 2^{n_\theta-1}, \text{ if } w_{\theta-1} = 0 \text{ and } w_\theta = 1; \\ 2^{n_\theta-2}, \text{ if } w_{\theta-1} = 1, \end{cases}$$

$$G_{\alpha,h} = \begin{cases} 1, \text{ if } h = 1; \\ 2^{n_h-1}, \text{ if } w_{h-1} = 0 \text{ and } h \geq 2; \\ 2^{n_h-2}, \text{ if } w_{h-1} = 1 \text{ and } h \geq 2. \end{cases}$$

Proof: Let $\Sigma = \{a_j \mid 1 \leq j \leq 2k-1\}$. Let $A_1 = (Q_1, \Sigma, \delta_1, 0, F_1)$ be a DFA, where

$$
\begin{aligned}
Q_1 &= \{0, 1, \ldots, n_1 - 1\}; \\
F_1 &= \{n_1 - 1\}; \\
\delta_1(t, a_1) &= t + 1 \bmod n_1,\ 0 \le t \le n_1 - 1; \\
\delta_1(t, a_{2k-2}) &= 0,\ 0 \le t \le n_1 - 1; \\
\delta_1(t, b) &= t,\ b \in \Sigma \setminus \{a_1, a_{2k-2}\},\ 0 \le t \le n_1 - 1.
\end{aligned}
$$

Figure 5.29 shows the transition diagram of A_1.

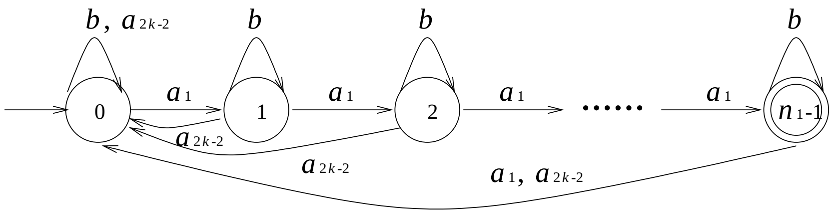

Figure 5.29: Witness DFA A_1 for Theorem 5.41

Let $A_i = (Q_i, \Sigma, \delta_i, 0, F_i)$, $2 \le i \le k$ be a DFA, where

$$
\begin{aligned}
Q_i &= \{0, 1, \ldots, n_i - 1\}; \\
F_i &= \{n_i - 1\}; \\
\delta_i(t, a_{2i-2}) &= t + 1 \bmod n_i, \quad 0 \le t \le n_i - 1; \\
\delta_i(t, a_{2i-1}) &= 1,\ 0 \le t \le n_i - 1; \\
\delta_i(t, b) &= t,\ b \in \Sigma \setminus \{a_{2i-2}, a_{2i-1}\},\ 0 \le t \le n_i - 1.
\end{aligned}
$$

Figure 5.30 shows the transition diagram of A_i.

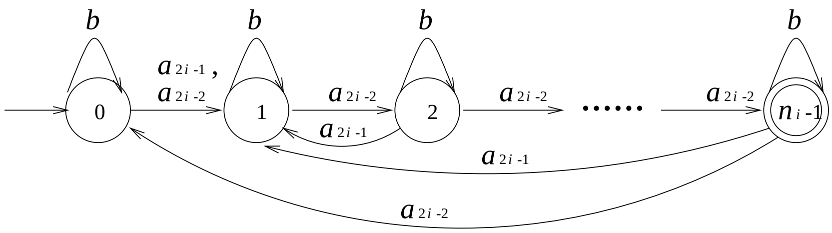

Figure 5.30: Witness DFA A_i for Theorem 5.41

For each $x \in \{a_1, a_2, a_4, \ldots, a_{2k-2}\}^*$, we define

$$
\begin{aligned}
P_s(x) = \{p \mid x &= u_1 u_2 \cdots u_s,\ u_l \in L(A_l),\ 1 \le l \le s - 1, \\
p &= |u_s|_{a_{2s-2}} \bmod n_s,\ 2 \le s \le k\}.
\end{aligned}
$$

Consider that $x, y \in \{a_1, a_2, a_4, \ldots, a_{2k-2}\}^*$ such that $P_s(x) \neq P_s(y)$. Let $c \in P_s(x) - P_s(y)$(or $P_s(y) - P_s(x)$) and $w = a_{2s-2}^{n_s-1-c} a_{2s+1} a_{2s}^{n_{s+1}-1} \cdots a_{2k-1} a_{2k-2}^{n_k-1}$. Then it is clear that $xw \in L(A_1) \cdots L(A_k)$ but $yw \notin L(A_1) \cdots L(A_k)$. So, x and y are in different equivalence classes of the right-invariant relation induced by $L(A_1) \cdots L(A_k)$.

For each $x \in \{a_1, a_2, a_4, \ldots, a_{2k-2}\}^*$, define

$$\begin{aligned} P_1(x) &= |z|_{a_1} \text{ where } x = ydz,\ y \in \{a_1, a_2, a_4, \ldots, a_{2k-2}\}^*, \\ &\quad z \in \{a_1, a_2, a_4, \ldots, a_{2k-4}\}^*, \text{ if } a_{2k-2} \text{ occurs in } x; \\ P_1(x) &= |x|_{a_1}, \text{ otherwise.} \end{aligned}$$

Consider $u, v \in \{a_1, a_2, a_4, \ldots, a_{2k-2}\}^*$ such that $P_1(u) \bmod n_1 > P_1(v) \bmod n_1$. Let $i = P_1(u) \bmod n_1$ and $w = a_1^{n_1-1-i} a_3 a_2^{n_2-1} \cdots a_{2k-1} a_{2k-2}^{n_k-1}$. Then clearly $uw \in L(A_1) \cdots L(A_k)$ but $vw \notin L(A_1) \cdots L(A_k)$.

Notice that there does not exist a string w such that $0 \notin P_2(w)$ and $P_1(w) = n_1 - 1$, since $P_1(w) = n_1 - 1$ guarantees that $0 \in P_2(w)$. Because of the same reason, there does not exist a string w such that $n_t - 1 \in P_t(w)$ and $0 \notin P_{t+1}(w)$, $2 \leq t \leq k-1$. It is also impossible that $P_t(w) = \emptyset$ but $P_{t+1}(w) \neq \emptyset$.

For each subset $p_s = \{d_{1,s}, \ldots, d_{e_s,s}\}$ of $\{0, \ldots, n_s - 1\}$ where $d_{1,s} > \cdots > d_{e_s,s}$ and $2 \leq s \leq k$, and an integer $p_1 \in \{0, \ldots, n_1 - 1\}$, except the cases we mentioned above, there exists a string

$$\begin{aligned} x \;=\; & a_1^{n_1} a_2^{n_2} a_4^{n_3} \cdots a_{2k-4}^{n_{k-1}} a_{2k-2}^{d_{1,k}-d_{2,k}} a_1^{n_1} a_2^{n_2} a_4^{n_3} \cdots a_{2k-4}^{n_{k-1}} a_{2k-2}^{d_{2,k}-d_{3,k}} \cdots \\ & a_1^{n_1} a_2^{n_2} a_4^{n_3} \cdots a_{2k-4}^{n_{k-1}} a_{2k-2}^{d_{e_k-1,k}-d_{e_k,k}} a_1^{n_1} a_2^{n_2} a_4^{n_3} \cdots a_{2k-4}^{n_{k-1}} a_{2k-2}^{d_{e_k,k}} \\ & a_1^{n_1} a_2^{n_2} a_4^{n_3} \cdots a_{2k-4}^{d_{1,k-1}-d_{2,k-1}} \cdots a_1^{n_1} a_2^{n_2} a_4^{n_3} \cdots a_{2k-4}^{d_{e_{k-1},k-1}} \cdots \\ & a_1^{n_1} a_2^{d_{1,2}-d_{2,2}} \cdots a_1^{n_1} a_2^{d_{e_2,2}} a_1^{p_1}. \end{aligned}$$

such that $P_1(x) = p_1$ and $P_s(x) = p_s$.

In total, there are $n_1 2^{n_2} 2^{n_3} \cdots 2^{n_k}$ classes. There are

$$D = \sum_{j=1}^{k-2} (n_1 \cdot (\prod_{r=2}^{j} (2^{n_r} - 1)) \cdot (2^{\sum_{q=j+2}^{k} n_q} - 1))$$

classes with both $p_t = \emptyset$ and $p_{t+1} \neq \emptyset$, $2 \leq t \leq k-1$. There are

$$E_1 = 1 + (2^{n_2-1} - 1) \cdot R_{1,1}$$

classes with both $p_1 = n_1 - 1$ and $0 \notin p_2$, which are not in D, where

$$\begin{aligned} R_{1,\mu} &= (1 + (2^{n_{\mu+2}} - 1) \cdot R_{1,\mu+1}) \text{ for } 1 \leq \mu \leq k-3; \\ R_{1,k-2} &= 2^{n_k}. \end{aligned}$$

There are

$$\begin{aligned} E_2 &= (n_1 - 1)2^{n_2-1}(1 + (2^{n_3-1} - 1) \cdot R_{2,1}) \\ &\quad + 2^{n_2-2}(1 + (2^{n_3-1} - 1) \cdot R_{2,1}) \end{aligned}$$

classes with both $n_2 - 1 \in p_2$ and $0 \notin p_3$, which are not in D, E_1, where

$$\begin{aligned} R_{2,\mu} &= (1 + (2^{n_{\mu+3}} - 1) \cdot R_{2,\mu+1}) \text{ for } 1 \leq \mu \leq k-4; \\ R_{2,k-3} &= 2^{n_k}. \end{aligned}$$

There are

$$\begin{aligned} E_3 &= (n_1 - 1)(2^{n_2-1} - 1)2^{n_3-1}(1 + (2^{n_4-1} - 1) \cdot R_{3,1}) \\ &\quad + (n_1 - 1)2^{n_2-1}2^{n_3-2}(1 + (2^{n_4-1} - 1) \cdot R_{3,1}) \\ &\quad + 2^{n_2-2}2^{n_3-1}(1 + (2^{n_4-1} - 1) \cdot R_{3,1}) \\ &\quad + 2^{n_2-2}2^{n_3-2}(1 + (2^{n_4-1} - 1) \cdot R_{3,1}) \end{aligned}$$

classes with both $n_3 - 1 \in p_3$ and $0 \notin p_4$, which are not in D, E_1, E_2, where

$$\begin{aligned} R_{3,\mu} &= (1 + (2^{n_{\mu+4}} - 1) \cdot R_{3,\mu+1}) \text{ for } 1 \leq \mu \leq k-5; \\ R_{3,k-4} &= 2^{n_k}. \end{aligned}$$

We omit the other similar classes until the hth group of such classes, $1 \leq h \leq k-1$. There are E_h classes with both $n_h - 1 \in p_h$ and $0 \notin p_{h+1}$, which are not in $D, E_1, E_2, \ldots, E_{h-1}$, where E_h is exactly the same as the one given in Theorem 5.41.

Thus, there are at least

$$n_1 2^{n_2+\ldots+n_k} - D - \sum_{h=1}^{k-1} E_h$$

distinct equivalence classes. *q.e.d.*

Before we investigate the upper bound on the state complexity of $L_1L_2\cdots L_2$, we first define an operation $\sqcup$ on R_1 and R_2 that are two classes of languages over Σ. Then

$$R_1 \sqcup R_2 = \{A \cup B \mid A \in R_1, B \in R_2\}.$$

We can easily see that $|R_1 \sqcup R_2| \leq |R_1| \cdot |R_2|$. The operation $\sqcup$ will be used in the proof of the following theorem.

Theorem 5.42 *Let A_i, $1 \leq i \leq k$ be k DFAs of n_i states, respectively, where A_i has f_i final states, $0 < f_i < n_i$. Then there exists a DFA of at most*

$$n_1 2^{n_2+\ldots+n_k} - D - \sum_{h=1}^{k-1} E_h$$

states that accepts $L(A_1)\cdots L(A_k)$, where

$$\begin{aligned}
D &= \sum_{j=1}^{k-2}(n_1 \cdot (\prod_{p=2}^{j}(2^{n_p}-1)) \cdot (2^{\sum_{q=j+2}^{k} n_q}-1));\\
E_h &= \sum_{v=0}^{2^{h-1}-1}((\prod_{y=1}^{h} G_{v,y}) \cdot (1+(2^{n_{h+1}-1}-1)\cdot R_{h,1}));\\
R_{h,t} &= (1+(2^{n_{t+h+1}}-1)\cdot R_{h,t+1}) \text{ for } 1 \leq t \leq k-h-2;\\
R_{h,k-h-1} &= 2^{n_k};
\end{aligned}$$

and for $w_z \in \{0,1\}$, $1 \leq z \leq h-1$ such that $w_1w_2w_3\cdots w_{h-1}$ is a binary number whose length is $h-1$ and value is v,

$$G_{v,1} = \begin{cases} n_1 - f_1, \text{ if } w_1 = 0 \text{ and } h \geq 2;\\ f_1, \text{ if } w_1 = 1 \text{ and } h \geq 2, \end{cases}$$

$$\text{for } 2 \leq x \leq h-1,\ G_{v,x} = \begin{cases} 2^{n_x-f_x}-1, \text{ if } w_{x-1}=0 \text{ and } w_x = 0;\\ (2^{f_x}-1)2^{n_x-f_x}, \text{ if } w_{x-1}=0 \text{ and } w_x=1;\\ 2^{n_x-f_x-1}, \text{ if } w_{x-1}=1 \text{ and } w_x=0;\\ (2^{f_x}-1)2^{n_x-f_x-1}, \text{ if } w_{x-1}=1 \text{ and } w_x=1, \end{cases}$$

$$G_{v,h} = \begin{cases} f_1, \text{ if } h=1;\\ (2^{f_h}-1)2^{n_h-f_h}, \text{ if } w_{h-1}=0 \text{ and } h \geq 2;\\ (2^{f_h}-1)2^{n_h-f_h-1}, \text{ if } w_{h-1}=1 \text{ and } h \geq 2. \end{cases}$$

Proof: Construct DFAs $A_i = (Q_i, \Sigma, \delta_i, 0, F_i)$, $1 \le i \le k$. Construct the DFA $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ such that

$$\begin{aligned}
Q_E &= Q_1 \times 2^{Q_2} \times 2^{Q_3} \times \cdots \times 2^{Q_k} - D' - \sum_{i=1}^{k-1} E'_i;\\
F_E &= \{\langle u_1, u_2, \ldots, u_k\rangle \in Q_E \mid u_k \cap F_k \neq \emptyset\};\\
q_0 &= \langle u_1, u_2, \ldots, u_k\rangle,
\end{aligned}$$

$u_1 = 0$, $u_c = \{0\}$, $u_d = \emptyset$, $2 \le c \le i$, $i+1 \le c \le k$ when $0 \in F_1$ and $0 \notin F_i$, $2 \le i \le k$;

$$\begin{aligned}
\delta_E \;:\; & \delta_E(\langle u_1, u_2, \ldots, u_k\rangle, a) = \langle u'_1, u'_2, \ldots, u'_k\rangle, \text{ for } a \in \Sigma,\\
& u'_1 = \delta_{A_1}(u_1, a),\\
& u'_2 = \delta_{A_2}(u_2, a) \cup \{0\} \text{ if } u'_1 \in F_1,\\
& u'_2 = \delta_{A_2}(u_2, a) \text{ otherwise},\\
& u'_i = \delta_{A_i}(u_i, a) \cup \{0\} \text{ if } u'_{i-1} \cap F_{i-1} \neq \emptyset,\\
& u'_i = \delta_{A_i}(u_i, a) \text{ otherwise}, \quad 3 \le i \le k,
\end{aligned}$$

where

$$\begin{aligned}
D' &= \bigcup_{j=1}^{k-2}(Q_1 \times (\prod_{p=2}^{j}(2^{Q_p} - \{\emptyset\})) \times \{\emptyset\} \times (\prod_{g=j+2}^{k} 2^{Q_g} - \{\emptyset\}^{k-j-1});\\
E'_1 &= F_1 \times (\{\emptyset\}^{k-1} \cup (2^{Q_2 - \{0\}} - \{\emptyset\}) \times R'_{1,1});\\
R'_{1,t} &= (\{\emptyset\}^{k-t-1} \cup (2^{Q_{t+2}} - \{\emptyset\}) \times R'_{1,t+1}) \text{ for } 1 \le t \le k-3;\\
R'_{1,k-2} &= 2^{Q_k};\\
E'_2 &= (Q_1 - F_1) \times ((2^{F_2} - \{\emptyset\}) \sqcup 2^{Q_2 - F_2}) \times (\{\emptyset\}^{k-2} \cup (2^{Q_3 - 1} - \{0\}) \times R'_{2,1})\\
&\quad \cup F_1 \times ((2^{F_2} - \{\emptyset\}) \sqcup 2^{Q_2 - F_2 - \{0\}} \cup \{0\}) \times (\{\emptyset\}^{k-2} \cup (2^{Q_3 - \{0\}} - \{\emptyset\}) \times R'_{2,1});\\
R'_{2,t} &= (\{\emptyset\}^{k-t-2} \cup (2^{Q_{t+3}} - \{\emptyset\}) \times R'_{2,t+1}) \text{ for } 1 \le t \le k-4;\\
R'_{2,k-3} &= 2^{Q_k};
\end{aligned}$$

$$\ldots$$

$$\begin{aligned}
E'_h &= \bigcup_{v=0}^{2^{h-1}-1}((\prod_{y=1}^{h} G'_{v,y}) \times (\{\emptyset\}^{k-h} \cup (2^{Q_{h+1} - \{0\}} - \{\emptyset\}) \times R'_{h,1}));\\
R'_{h,t} &= (\{\emptyset\}^{k-t-h} \cup (2^{Q_{t+h+1}} - \{\emptyset\}) \times R'_{h,t+1}) \text{ for } 1 \le t \le k-h-2;\\
R'_{h,k-h-1} &= 2^{Q_k};
\end{aligned}$$

and for $w_z \in \{0,1\}$, $1 \leq z \leq h-1$ such that $w_1 w_2 w_3 \cdots w_{h-1}$ is a binary number whose length is $h-1$ and value is v,

$$G'_{v,1} = \begin{cases} Q_1 - F_1, \text{ if } w_1 = 0 \text{ and } h \geq 2; \\ F_1, \text{ if } w_1 = 1 \text{ and } h \geq 2, \end{cases}$$

$$\text{for } 2 \leq x \leq h-1,\ G'_{v,x} = \begin{cases} 2^{Q_x - F_x} - \{\emptyset\} \quad, \text{ if } w_{x-1} = 0 \text{ and } w_x = 0; \\ (2^{F_x} - \{\emptyset\} \quad) \sqcup 2^{Q_x - F_x}, \text{ if } w_{x-1} = 0 \text{ and } w_x = 1; \\ 2^{Q_x - F_x - \{0\}}, \text{ if } w_{x-1} = 1 \text{ and } w_x = 0; \\ (2^{F_x} - \{\emptyset\} \quad) \sqcup 2^{Q_x - F_x - \{0\}}, \text{ if } w_{x-1} = 1 \text{ and } w_x = 1, \end{cases}$$

$$G'_{v,h} = \begin{cases} F_1, \text{ if } h = 1; \\ (2^{F_h} - \{\emptyset\} \quad) \sqcup 2^{Q_h - F_h}, \text{ if } w_{h-1} = 0 \text{ and } h \geq 2; \\ (2^{F_h} - \{\emptyset\} \quad) \sqcup 2^{Q_h - F_h - \{0\}}, \text{ if } w_{h-1} = 1 \text{ and } h \geq 2. \end{cases}$$

Intuitively, Q_E is a set of k-tuples whose first component is a state in Q_1 and ith component is a subset of states in Q_i, $2 \leq i \leq k$.

Q_E does not contain those k-tuples whose ith component is $\emptyset$ and whose jth component is not $\emptyset$, when $1 < i < j \leq k$. D' is the set of them.

Q_E does not contain those k-tuples whose first component is an element of F_1 and second component is not $\emptyset$ (if it is $\emptyset$ then all the elements afterwards have to be $\emptyset$) and does not contain 0, either. E'_1 is the set consisting of them.

Q_E does not contain those k-tuples whose hth component contains one or more final states of A_h and whose $(h+1)$th component is not $\emptyset$ (if it is $\emptyset$ then all the elements afterwards have to be $\emptyset$) and does not contain 0, when $1 \leq h \leq k-1$, either. E'_h is the set consisting of them. Note that E'_h does not contain the k-tuples that belong to E'_j where $1 \leq j < i$.

Clearly, $L(E) = L(A_1) \cdots L(A_k)$. Let $|Q_{A_i}| = n_i$ and $|F_{A_i}| = f_i$, $1 \leq i \leq k$. Then E has the following number of states

$$n_1 2^{n_2 + \ldots + n_k} - D - \sum_{h=1}^{k-1} E_h.$$

q.e.d.

Note that when each A_i, $1 \leq i \leq k$, has one final state, this upper bound is exactly the same as the lower bound stated in Theorem 5.41. Thus, this bound is tight and is the state complexity of the catenation of k regular languages.

Chapter 6

Estimation and Approximation of State Complexity of Combined Operations

There are at least two problems concerning the exact state complexities of combined operations. Firstly, the exact state complexities of many combined operations are extremely difficult to compute. Secondly, a large proportion of results that have been obtained are rather complex and difficult to comprehend [32]. For example, the exact state complexity of the catenation of four regular languages accepted by m, n, p, q states, respectively, is

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

for $m, n, p \geq 2$, which is difficult to understand.

It is clear that good estimates and approximations of state complexities can be used in these two cases. We will first investigate estimation of state complexity of combined operations in the following.

6.1 Estimation of State Complexity of Combined Operations

In [97, 108], estimation based on nondeterministic state complexity was introduced. Briefly speaking, for a combined operation on regular languages, the method first estimates the nondeterministic state complexity of the combined operation using the composition of the nondeterministic state complexities of its component operations, and then converts it to an estimate of the deterministic state complexity [27]. For example, for $(L(A) \cup L(B))^*$ where A and B are DFAs of m states and n states, respectively, the nondeterministic state complexity of $L = L(A) \cup L(B)$ is $m+n+1$ and that of L^* is $m+n+2$, which is then converted to an estimation of the deterministic state complexity 2^{m+n+2}. Note that the nondeterministic state complexity $m+n+2$ is the direct mathematical composition of the two individual nondeterministic state complexities [27]. No optimization is made. Other nondeterministic state complexities for combined operations in this chapter are calculated in the same way.

It has been shown that this method can obtain good estimates for the combined operations: star of union, star of intersection, star of catenation, and star of reversal. Table 6.1 shows their actual state complexities and corresponding estimates.

Table 6.1: The exact state complexities of 4 combined operations and corresponding estimates

Operation	State Complexity	Estimate
$(L(A) \cup L(B))^*$	$2^{m+n-1} - 2^{m-1} - 2^{n-1} + 1$ [92]	2^{m+n+2} [27]
$(L(A) \cap L(B))^*$	$3 \cdot 2^{mn-2}$ [62]	2^{mn+1} [27]
$(L(A)L(B))^*$	$2^{m+n-1} + 2^{m+n-4} - 2^{m-1} - 2^{n-1} + m + 1$ [31]	2^{m+n+1} [27]
$(L(B)^R)^*$	2^n [31]	2^{n+2} [27]

However, this method clearly has its limitations. For example, we would obtain $2^{n_1+n_2+n_3+2}$ for the union of n_1-state, n_2-state, and n_3-state DFA languages using this method. However, the actual state complexity of this combined operation is $n_1 n_2 n_3$.

It seems that this method may work well for all combined operations with the final component operation having an exponential state complexity, e.g., star or reversal. Indeed, it works well when a combined operation is ended with the star operation. However, it does not work well in general for combined operations that are ended with reversal. For example, the state complexity of the reversal of the intersection of an m-state DFA language and an n-state DFA language is $2^{m+n} - 2^m - 2^n + 2$. However, we would obtain the estimate 2^{mn+1} using this method.

The following result was obtained in [97], where a regular operation expression is an expression built from occurrences of binary operations union and concatenation, occurrences of the unary operation star, and variables, where each variable occurs at most once in the expression, and $\mathrm{nsc}(f)$ denotes the nondeterministic state complexity of the operation f expressed by a regular operation expression.

Theorem 6.1 *Let f be an operation defined by a regular operation expression with k variables, and denote the number of states of the NFAs for the argument languages by $m_1, \ldots, m_k$. Then*

$$\mathrm{nsc}(f) \leq 1 + \sum_{i=1}^{k} m_i. \tag{6.1}$$

Using the above result, we easily obtain the following estimates [97].

Corollary 6.1 *Let f be an operation defined by a regular operation expression with k variables and denote the number of states of the NFAs for the argument languages by $m_1, \ldots, m_k$. Then the state complexity of f is no more than $2^{m_1+\cdots+m_k+1}$.*

We can see clearly that when the star operation is the final operation of f, the upper bound is almost tight [27].

6.2 Approximation of State Complexity of Combined Operations

Although an estimation of the state complexity of a combined operation is simpler and more convenient to use, it does not show how close it is to the exact state complexity. To solve this problem, we study approximation of state complexity [32].

The idea of approximation of state complexity is from the notion of approximation algorithms which was formalized in early 1970's by David S. Johnson et al. [34, 64, 65]. Many polynomial-time approximation algorithms have been designed for quite a large number of NP-complete problems, which include the well-known traveling-salesman problem, the set-covering problem, and the subset-sum problem. Obtaining an optimal solution for an NP-complete problem is considered intractable. Near optimal solutions are often good enough in practice. Assuming that the problem is a maximization or a minimization problem, an approximation algorithm is said to have a ratio bound of $\rho(n)$ if for any input of size n, the cost C of the solution produced by the algorithm is within a factor of $\rho(n)$ of the cost C^* of an optimal solution [19]:

$$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \leq \rho(n).$$

The concept of approximation of state complexity is in many ways similar to that of approximation algorithms. An approximation of state complexity is close to the exact state complexity and normally not equal to it. The ratio bound shows the error range of the approximation. In addition to the property of having a small ratio bound in general, we also consider that an approximation of state complexity should be in a simple and intuitive form.

In spite of the similarities, there are fundamental differences between an approximation of state complexity and an approximation algorithm. The efforts in the area of approximation algorithms are in finding polynomial algorithms for NP-complete problems such that the results of the algorithms approximate the optimal results. In comparison, the efforts in approximation of state complexity are in searching directly for estimates of state complexities such that they satisfy certain ratio bounds. The aim of designing an approximation algorithm is to transform an intractable problem into one that is easier to compute where the result is acceptable although not optimal. In comparison, an approximation of state complexity may have two different effects: (1) it gives a reasonable estimation of certain state complexity, with some bound, the exact value of which is difficult or impossible to compute; or (2) it gives a simpler and more comprehensible formula that approximates a known state complexity.

Let ξ be a combined operation on k regular languages. Assume that the state complexity of ξ is θ. We say that α is an approximation of the state complexity of the operation ξ with ratio bound ρ if, for any sufficiently large positive integers $n_1, \ldots, n_k$, which are the numbers of states of the DFAs that accept the argument languages of the operation, respectively,

$$\max\left(\frac{\alpha(n_1, \ldots, n_k)}{\theta(n_1, \ldots, n_k)}, \frac{\theta(n_1, \ldots, n_k)}{\alpha(n_1, \ldots, n_k)}\right) \leq \rho(n_1, \ldots, n_k).$$

Note that in many cases, ρ is a constant.

Here are some examples. Consider the estimates of state complexities of the four combined operations listed in the previous table to be approximations of state complexities. Then we can easily get their ratio bounds in Table 6.2 by comparing them with the exact state complexities. In the above cases,

Table 6.2: The ratio bounds of the approximations of the state complexities of 4 combined operations [32]

Operation	Ratio bound of the approximation
$(L(A) \cup L(B))^*$	≈ 8
$(L(A) \cap L(B))^*$	$8/3$
$(L(A)L(B))^*$	4
$(L(B)^R)^*$	4

although the exact state complexities have been obtained, the approximation results with small ratio bounds are good enough for practical purposes, and they clearly have the advantage of being more intuitive and simpler in formulation.

In rest of this section, we consider two combined operations: (1) star of left quotient and (2) left quotient of star. For each of the combined operations, we do not have the exact state complexity; however, an approximation with a good ratio bound is obtained in our paper [32]. In the following, we assume that all languages are over an alphabet of at least two letters.

Theorem 6.2 *Let R be a language accepted by an n-state DFA M, $n > 0$, and L be an arbitrary language. Then there exists a DFA of at most 2^n states that accepts $(L \backslash R)^*$.*

Proof: Let $M = (Q, \Sigma, \delta, s, F)$ be a complete DFA of n states and $R = L(M)$. For each $q \in Q$, denote by $L(M_q)$ the set $\{w \in \Sigma^* \mid \delta(s, w) = q\}$. We construct an NFA M' with multiple initial states to accept $(L \backslash R)^+$ as follows. M' is the same as M except that the initial state s of M is replaced by the set of initial states $S = \{q \mid L(M_q) \cap L \neq \emptyset\}$ and ε-transitions are added from each final state to the states in S. By performing the subset construction, we can construct a DFA A' of no more than $2^n - 1$ states that is equivalent to M'.

Note that $\emptyset$ is not a state of A'. From the DFA A', we construct a new DFA A by just adding a new initial state that is also a final state and the transitions from this new state that are the same as the transitions from the original initial state of A'. It is easy to see that $L(A) = (L\backslash R)^*$ and A has 2^n states. *q.e.d.*

This result gives an upper bound for the state complexity of the combined operation: star of left quotient. It means that for any n-state DFA language R, $n > 0$, and an arbitrary language L, the state complexity of the star of the left quotient of R by L is no more than 2^n.

Theorem 6.3 *For any integer $n \geq 2$, there exists a DFA M of n states and a language L such that any DFA that accepts $(L\backslash L(M))^*$ needs at least $2^{n-1} + 2^{n-2}$ states.*

Proof: For $n = 2$, it is clear that $R = \{w \in \{a, b\}^* \mid |w|_a \text{ is odd}\}$ is accepted by a two-state DFA, and

$$(\{\varepsilon\}\backslash R)^* = R^* = \{\varepsilon\} \cup \{w \in \{a, b\}^* \mid |w|_a \geq 1\}$$

cannot be accepted by a DFA with less than three states.

For $n > 2$, let $M = (Q, \{a, b\}, \delta, 0, \{n-1\})$ where $Q = \{0, 1, \ldots, n-1\}$ and

$$\begin{aligned}
\delta(i, a) &= (i+1) \bmod n,\ i = 0, 1, \ldots, n-1, \\
\delta(0, a) &= 0, \\
\delta(j, b) &= (j+1) \bmod n,\ j = 1, \ldots, n-1.
\end{aligned}$$

M is the same as the witness DFA for the state complexity of star operation on regular languages. The transition diagram of M is shown in Figure 3.4. It has been proved in [111] that the minimal DFA that accepts $L(M)^*$ has $2^{n-1} + 2^{n-2}$ states. Let $L = \{\varepsilon\}$. Then $(L\backslash L(M))^* = L(M)^*$. So, any DFA that accepts $(L\backslash L(M))^*$ needs at least $2^{n-1} + 2^{n-2}$ states. *q.e.d.*

This result gives a lower bound for the state complexity of star of left quotient. Clearly, the lower bound does not coincide with the upper bound. We still do not know the exact state complexity for this combined operation, and it could be difficult to obtain. However, we can easily obtain a good state complexity approximation for the operation. Let 2^n be the approximation. Then the ratio bound is

$$\frac{2^n}{2^{n-1}+2^{n-2}} = \frac{4}{3} \quad .$$

Next we consider the combined operation: left quotient of star.

Theorem 6.4 *Let R be a language accepted by an n-state DFA M and L be an arbitrary language. Then there exists a DFA of at most $2^{n+1}-1$ states that accepts $L \backslash R^*$.*

Proof: Let $M = (Q, \Sigma, \delta, s, F)$ be a complete DFA of n states and $R = L(M)$. Then we can easily construct an $(n+1)$-state NFA $M' = (Q \cup \{s'\}, \Sigma, \delta', s', F \cup \{s'\})$ such that $L(M') = R^*$ by adding a new initial state s' and transitions $\delta'(s', \varepsilon) = s$ and $\delta'(f, \varepsilon) = s'$ for each final state $f \in F$. For each $q \in Q \cup \{s'\}$, we denote by $L(M_q)$ the set $\{w \in \Sigma^* \mid q \in \delta'(s', w)\}$. We construct an NFA N with multiple initial states to accept $L \backslash L(M') = L \backslash R^*$ as follows. N is the same as M' except that the initial state s' of M' is replaced by the set of initial states $S = \{q \mid L(M_p) \cap L \neq \emptyset\}$. By performing the subset construction, we can verify that there exists a DFA A of no more than $2^{n+1}-1$ states that is equivalent to N. Note that $\emptyset$ is not a state of A. It is easy to see that

$$L(A) = L(N) = L \backslash L(M') = L \backslash R^*.$$

So, $2^{n+1}-1$ is an upper bound on the state complexity of the left quotient of star operation. *q.e.d.*

Theorem 6.5 *For any integer $n \geq 2$, there exist a DFA M of n states and a language L such that any DFA that accepts $L \backslash L(M)^*$ needs at least $2^{n-1}+2^{n-2}$ states.*

Proof: For $n = 2$, we still use $R = \{w \in \{a,b\}^* \mid |w|_a \text{ is odd}\}$ which is accepted by a two-state DFA. $\{\varepsilon\} \backslash R^* = R^*$ cannot be accepted by a DFA with less than three states.

Again we use DFA M shown in Figure 3.4 for any integer $n > 2$. As stated before, it has been proved that the minimal DFA that accepts $L(M)^*$ has $2^{n-1}+2^{n-2}$ states. So any DFA that accepts $L \backslash L(M)^*$ needs at least $2^{n-1}+2^{n-2}$ states. *q.e.d.*

For the combined operation: left quotient of star, we choose 2^{n+1} to be an approximation of its state complexity. Then the ratio bound can be calculated easily as follows:

$$\frac{2^{n+1}}{2^{n-1}+2^{n-2}} = \frac{8}{3} \quad .$$

Chapter 7

Conclusion and Future Work

In this thesis, some recent results of the study of state complexity are summarized and our new research results on state complexity of combined operations are presented and proved.

7.1 Summary of Results

Assume that there are k regular languages over the same alphabet, where $k \geq 3$. The language L_i is one of them and accepted by an n_i-state DFA $A_i = (Q_i, \Sigma, \delta_i, s_i, F_i)$, $1 \leq i \leq k$. The exact state complexities of the 12 combined operations investigated in this thesis are listed in Table 7.1.

The state complexities of most of these combined operations are smaller than the compositions of the state complexities of individual operations that form the combinations. Only the state complexities of $L_1(L_2 \cap L_3)$ and combined Boolean operations are the same as the compositions of the state complexities of the component operations.

The reason for this difference is that the result of the first operation is not among the worst cases of the second operation. Thus, the state complexity of a combined operation does not necessarily equal the composition of the state complexities of individual operations that form the combination. Figure 7.1 shows this situation clearly.

Note that in the proofs of the lower bounds of the state complexities of combined Boolean operations and $L_1L_2\cdots L_k$, alphabets of size dependant on k are used. It remains open whether the same results can be obtained with witness regular languages over fixed alphabets.

Table 7.1: The exact state complexities of the 12 combined operations investigated in this thesis

Operation	State complexity
$L_1^* L_2$	$5 \cdot 2^{n_1+n_2-3} - 2^{n_1-1} - 2^{n_2} + 1$ [12]
$L_1 L_2^*$	$n_1 \frac{3}{4} 2^{n_2} - 2^{n_2-2}$ [10]
$L_1^R L_2$	$3 \cdot 2^{n_1+n_2-2}$ [12]
$L_1 L_2^R$	$n_1 2^{n_2} - 2^{n_2-1} - n_1 + 1$ [10]
$L_1(L_2 \cup L_3)$	$(n_1 - 1)(2^{n_2+n_3} - 2^{n_2} - 2^{n_3} + 2) + 2^{n_2+n_3-2}$ [11]
$L_1(L_2 \cap L_3)$	$n_1 2^{n_2 n_3} - 2^{n_2 n_3 - 1}$ [11]
$L_1^* \cup L_2$	$\frac{3}{4} 2^{n_1} \cdot n_2 - n_2 + 1$ [33]
$L_1^* \cap L_2$	$\frac{3}{4} 2^{n_1} \cdot n_2 - n_2 + 1$ [33]
$L_1^R \cup L_2$	$2^{n_1} \cdot n_2 - n_2 + 1$ [33]
$L_1^R \cap L_2$	$2^{n_1} \cdot n_2 - n_2 + 1$ [33]
Combined Boolean operations on $L_1, L_2, \ldots, L_k$	$n_1 n_2 \cdots n_k$ [27]
$L_1 L_2 L_3$	$n_1 2^{n_2+n_3} - 2^{n_2+n_3-1} - (n_1 - 1) 2^{n_2+n_3-2}$ $-2^{n_2+n_3-3} - (n_1 - 1)(2^{n_3} - 1)$ [27]

In this thesis, we have also discussed estimation and approximation of state complexity. We have reviewed the estimation method based on nondeterministic state complexity and pointed out that this method may work well for all combined operations with the final component operation having an exponential state complexity. The new concept of approximation of state complexity further advances the idea of estimation of state complexity by including the ratio bound. The ratio bound gives a precise and intuitive measurement on the "quality" of the estimation. We have given the approximations of the state complexities of 6 combined operations on regular languages which are shown in Table 7.2.

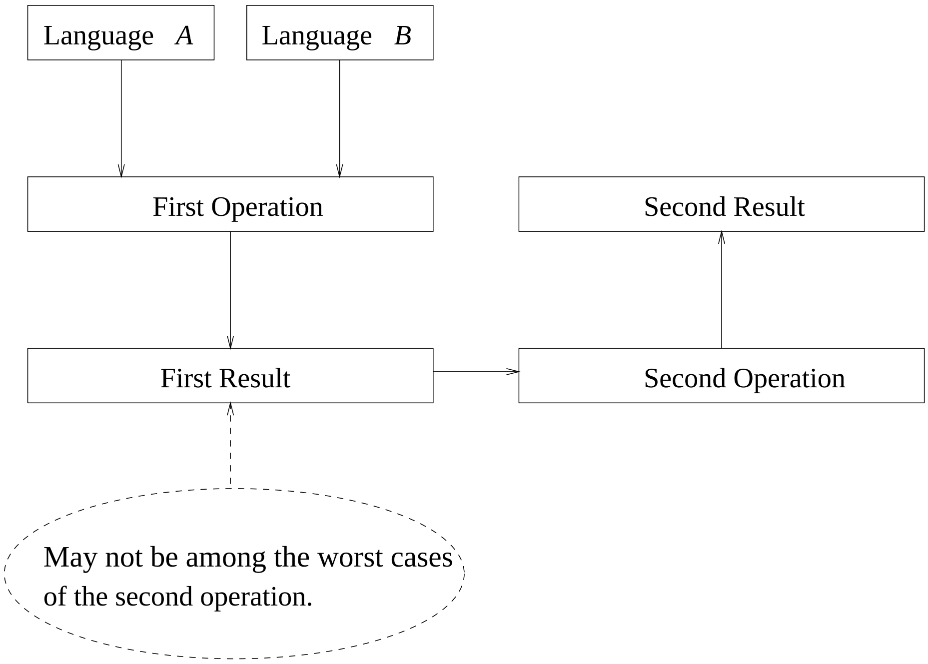

Figure 7.1: The reason for the difference in state complexity

Table 7.2: The approximations of the state complexities of 6 combined operations [32]

Operation	Approximation of state complexity	Ratio bound
$(L_1 \cup L_2)^*$	2^{m+n+2}	≈ 8
$(L_1 \cap L_2)^*$	2^{mn+1}	$\frac{8}{3}$
$(L_1 L_2)^*$	2^{m+n+1}	4
$(L_1^R)^*$	2^{n+2}	4
$(L \backslash L_1)^*$	2^{n_1}	$\frac{4}{3}$
$L \backslash L_1^*$	2^{n_1+1}	$\frac{8}{3}$

7.2 List of Contributions

I am the main contributor in the research on the state complexities of eight combined combined operations among the twelve summarized in Section 7.1, which are shown as follows:

1. the state complexity of $L_1^R L_2$ [12];

2. the state complexity of $L_1L_2^R$ [10];

3. the state complexity of $L_1(L_2 \cap L_3)$ [11];

4. the state complexity of $L_1^* \cup L_2$ [33];

5. the state complexity of $L_1^* \cap L_2$ [33];

6. the state complexity of $L_1^R \cup L_2$ [33];

7. the state complexity of $L_1^R \cap L_2$ [33];

8. the state complexity of multiple catenations [27];

For all the eight combined operations, I first find the upper bounds on their state complexities. Then I do hundreds of experiments to find their worst-case examples that attain the upper bounds, and finally, I prove their state complexities theoretically.

The state complexities of the other four combined operations listed in the following are obtained through the teamwork of me and the co-authors of the papers in which these results are presented.

9. the state complexity of $L_1^*L_2$ [12];

10. the state complexity of $L_1L_2^*$ [10];

11. the state complexity of $L_1(L_2 \cup L_3)$ [11];

12. the state complexity of combined Boolean operations [27].

For these results, my contributions are mainly in obtaining their upper bounds and finding the corresponding worst-case examples through experiments.

I am also the main contributor in the study on the approximations of the state complexities of $(L\backslash L_1)^*$ and $L\backslash L_1^*$ [32]. By finding and proving their upper bounds and lower bounds, I obtain their approximations and ratio bounds. The other results on estimation and approximation of state complexity are obtained through teamwork [27, 32].

7.3 Future Work

There are still many interesting combined operations that have not yet been studied. The compositions may not necessarily be restricted to two operations. The compositions of three or more individual operations will be much more complex. The sequence in which the individual operations are performed can also be changed when they form compositions. In this way, the state complexity will change, too.

There might also be some more general rules on the relationship between the state complexity of a combined operation and the state complexities of its individual component operations. Many further problems in this direction are to be solved in the near future.

REFERENCES

[1] A. Aho, J. Hopcroft and J. Ullman: The Design and Analysis of Computer Algorithms, Addison-Wesley, 1974

[2] M. Amos: Theoretical and Experimental DNA Computation (Natural Computing Series), Springer, 2005

[3] F. Bassino, L. Giambruno and C. Nicaud: The average state complexity of the star of a finite set of words is linear, in: *Proceedings of the 12th Developments in Language Theory (DLT 2008)*, Springer, LNCS 5257, 2008, 134–145

[4] J. Birget: Intersection and union of regular languages and state complexity, *Information Processing Letters*, vol. 43 no. 4 (1992) 185–190

[5] J. Birget: State-complexity of finite-state devices, state compressibility and incompressibility, *Mathematical Systems Theory*, vol. 26 no. 3 (1993) 237–269

[6] J. Birget: Partial orders on words, minimal elements of regular languages and state complexity, *Theoretical Computer Science*, vol. 119 no. 2 (1993) 267–291

[7] J. Birget: The state complexity of $\overline{\Sigma^*\overline{L}}$ and its connection with temporal logic, *Information Processing Letters*, vol. 58 no. 4 (1996) 185–188

[8] J. Brzozowski: Quotient complexity of regular languages, in: *Proceedings of Descriptional Complexity of Formal Systems 11th Workshop (DCFS 2009)*, Magdeburg, Germany, 2009, 17–28

[9] J. Brzozowski, S. Konstantinidis: State-complexity hierarchies of uniform languages of alphabet-size length, in: *Theoretical Computer Science*, vol. 410 no. 35 (2009) 3223–3235

[10] B. Cui, Y. Gao, L. Kari and S. Yu: State complexity of catenation combined with star and reversal, in: *Proceedings of Descriptional Complexity of Formal Systems 12th Workshop (DCFS 2010)*, Saskatoon, Canada, 2010, 74–85

[11] B. Cui, Y. Gao, L. Kari and S. Yu: State complexity of catenation combined with union and intersection, in: *Proceedings of 15th International Conference on Implementation and Application of Automata (CIAA 2010)*, Winnipeg, Canada, 2010, 89–98

[12] B. Cui, Y. Gao, L. Kari and S. Yu: State complexity of two combined operations: reversal-catenation and star-catenation, *Computing Research Repository* (2010) arXiv:1006.4646v1

[13] C. Câmpeanu, K. Culik, K. Salomaa and S. Yu: State complexity of basic operations on finite languages, in: *Proceedings of 4th International Workshop on Implementing Automata (WIA 1999)*, Springer, LNCS 2214, 2001, 60–70

[14] C. Câmpeanu, W. Ho: The maximum state complexity for finite languages, *Journal of Automata, Languages and Combinatorics*, vol. 9 no. 2-3 (2004) 189–202

[15] C. Câmpeanu, A. Paun: Tight bounds for NFA to DFCA transformations for binary alphabets, in: *Proceedings of 9th International Conference on Implementation and Application of Automata (CIAA 2004)*, Springer, LNCS 3317, 2004, 306–307

[16] C. Câmpeanu, 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, 53–73

[17] C. Câmpeanu, K. Salomaa and S. Yu: Tight lower bound for the state complexity of shuffle of regular languages, *Journal of Automata, Languages and Combinatorics*, vol. 7 no. 3 (2002) 303–310

[18] E. Charlier, N. Rampersad, M. Rigo and L. Waxweiler: State complexity of testing divisibility, in: *Proceedings of Descriptional Complexity of*

Formal Systems 12th Workshop (DCFS 2010), Saskatoon, Canada, 2010, 62–73

[19] T. Cormen, C. Leiserson and R. Rivest: Introduction to Algorithms, MIT Press and McGraw-Hill, 1990

[20] C. Calude, K. Salomaa and T. Roblot: Finite state complexity and the size of transducers, in: *Proceedings of Descriptional Complexity of Formal Systems 12th Workshop (DCFS 2010)*, Saskatoon, Canada, 2010, 50–61

[21] M. Daley, M. Domaratzki, K. Salomaa: State complexity of orthogonal catenation, in: *Proceedings of Descriptional Complexity of Formal Systems 10th Workshop (DCFS 2008)*, Charlottetown, Canada, 2008, 134–144

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

[23] M. Domaratzki, A. Okhotin: State complexity of power, *Theoretical Computer Science*, vol. 410 no. 24-25 (2009) 2377–2392

[24] M. Domaratzki, K. Salomaa: State complexity of shuffle on trajectories, *Journal of Automata, Languages and Combinatorics*, vol. 9 no. 2-3 (2004) 217–232

[25] M. Domaratzki, K. Salomaa: Lower bounds for the transition complexity of NFAs, in: *Proceedings of 31st International Symposium on Mathematical Foundations of Computer Science (MFCS 2006)*, Springer, LNCS 4162, 2007, 315–326

[26] K. Ellul: Descriptional complexity measures of regular languages, Master's Thesis, University of Waterloo, Canada, 2002

[27] Z. Ésik, Y. Gao, G. Liu and S. Yu: Estimation of state complexity of combined operations, *Theoretical Computer Science*, vol. 410 no. 35 (2009) 3272–3280

[28] Y. Gao: State complexity of combined operations on regular languages, Master's Thesis, University of Western Ontario, Canada, 2006

[29] Y. Gao, Z. Ésik, G. Liu and S. Yu: State complexity of intersection, union and catenation on k regular languages, Technical Report 715, Department of Computer Science, the University of Western Ontario, 2008

[30] Y. Gao, K. Salomaa and S. Yu: Transition complexity of incomplete DFAs, in: *Proceedings of Descriptional Complexity of Formal Systems 12th Workshop (DCFS 2010)*, Saskatoon, Canada, 2010, 123–134

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

[32] Y. Gao, S. Yu: State complexity approximation, in: *Proceedings of Descriptional Complexity of Formal Systems 11th Workshop (DCFS 2009)*, Magdeburg, Germany, 2009, 163-174

[33] Y. Gao, S. Yu: State complexity of union and intersection combined with star and reversal, *Computing Research Repository* (2010) arXiv:1006.3755v1

[34] M. Garey, R. Graham and J. Ullman: Worst-case analysis of memory allocation algorithms, in: *Proceedings of 4th Annual ACM Symposium on the Theory of Computing*, 1972, 143–150

[35] H. Gruber, M. Holzer: Finding lower bounds for nondeterministic state complexity is hard, in: *Proceedings of the 10th Developments in Language Theory (DLT 2006)*, Springer, LNCS 4036, 2006, 363–374

[36] H. Gruber, M. Holzer: Inapproximability of nondeterministic state and transition complexity assuming $P \neq NP$, in: *Proceedings of the 11th Developments in Language Theory (DLT 2007)*, Springer, LNCS 4588, 2007, 205–216

[37] H. Gruber, M. Holzer: On the average state and transition complexity of finite languages, *Theoretical Computer Science*, vol. 387 no. 2 (2007) 155–166

[38] J. Gruska, A. Monti, M. Napoli and D. Parente: State complexity of SBTA languages, in: *Proceedings of 2nd Latin American Symposium Valparaiso (LATIN 1995)*, Springer, LNCS 911, 1995, 346–357

[39] Y. Han, K. Salomaa: State complexity of union and intersection of finite languages, in: *Proceedings of the 11th Developments in Language Theory (DLT 2007)*, Springer, LNCS 4588, 2007, 217–228

[40] Y. Han, K. Salomaa: State complexity of union and intersection of finite languages, *International Journal of Foundations of Computer Science*, vol. 19 no. 3 (2008) 581–595

[41] Y. Han, K. Salomaa: State complexity of basic operations on suffix-free regular languages, *Theoretical Computer Science*, vol. 410 no. 27-29 (2009) 2537–2548

[42] Y. Han, K. Salomaa: Nondeterministic state complexity of nested word automata, *Theoretical Computer Science*, vol. 410 no. 30-32 (2009) 2961–2971

[43] Y. Han, K. Salomaa: Nondeterministic state complexity for suffix-free regular languages, in: *Proceedings of Descriptional Complexity of Formal Systems 12th Workshop (DCFS 2010)*, Saskatoon, Canada, 2010, 227–235

[44] Y. Han, K. Salomaa and D. Wood: Nondeterministic state complexity of basic operations for prefix-free regular languages, *Fundamenta Informaticae*, vol. 90 no. 1-2 (2009) 93–106

[45] Y. Han, K. Salomaa and S. Yu: State complexity of combined operations for prefix-free regular languages, in: *Proceedings of 3rd International Conference on Language and Automata Theory and Applications (LATA 2009)*, Springer-Verlag, LNCS 5457, 2009, 398–409

[46] M. Holzer, B. König: Regular languages, sizes of syntactic monoids, graph colouring, state complexity results, and how these topics are related to each other (Column: Formal Language Theory), *Bulletin of the EATCS*, vol. 83 (2004) 139–155

[47] M. Holzer, M. Kutrib: State complexity of basic operations on nondeterministic finite automata, in: *Proceedings of 7th International Conference on Implementation and Application of Automata (CIAA 2002)*, Springer, LNCS 2608, 2003, 148–157

[48] M. Holzer, M. Kutrib: Unary language operations and their nondeterministic state complexity, in: *Proceedings of the 6th Developments in Language Theory (DLT 2002)*, Kyoto, Japan, 2002, Springer LNCS 2450, 162–172

[49] M. Holzer, M. Kutrib: Nondeterministic finite automata-recent results on the descriptional and computational complexity, in: *Proceedings of 13th International Conference on Implementation and Application of Automata (CIAA 2008)*, Springer, LNCS 5148, 2008, 1–16

[50] M. Holzer, K. Salomaa and S. Yu: On the state complexity of K-entry deterministic finite automata, *Journal of Automata, Languages and Combinatorics*, vol. 6 (2001) 453–466

[51] J. Hopcroft: An $n \log n$ algorithm for minimizing states in a finite automaton, *The Theory of Machines and Computations*, Academic Press, 1971, 189–196

[52] J. Hopcroft, R. Motwani and J. Ullman: Introduction to Automata theory, Languages, and Computation, 2nd ed., Addison-Wesley, 2001

[53] J. Hopcroft, J. Ullman: Introduction to Automata Theory, Languages, and Computation, 1st ed., Addison-Wesley, 1979

[54] L. Ilie, R. Solis-Oba and S. Yu: Reducing the size of NFAs by using equivalences and preorders, in: *Proceedings of 16th Annual Symposium on Combinatorial Pattern Matching (CPM 2005)*, Springer, LNCS 3537, 2005, 306–307

[55] 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*, vol. 237 (2002) 485–494

[56] J. Jirásek, G. Jirásková, A. Szabari: State complexity of concatenation and complementation of regular languages, *International Journal of Foundations of Computer Science*, vol. 16 (2005) 511–529

[57] G. Jirásková: State complexity of some operations on regular languages, in: *Proceedings of Descriptional Complexity of Formal Systems 5th Workshop (DCFS 2003)*, Budapest, Hungary, 2003, 114–125

[58] G. Jirásková: State complexity of some operations on binary regular languages, *Theoretical Computer Science*, vol. 330 (2005) 287–298

[59] G. Jirásková: On the state complexity of complements, stars, and reversals of regular languages, in: *Proceedings of the 12th Developments in Language Theory (DLT 2008)*, Springer, LNCS 5257, 2008, 431–442

[60] G. Jirásková: Concatenation of regular languages and descriptional complexity, in: *Proceedings of 4th International Computer Science Symposium in Russia (CSR 2009)*, Springer, LNCS 5675, 2009, 203–214

[61] G. Jirásková, A. Okhotin: State complexity of cyclic shift, in: *Proceedings of Descriptional Complexity of Formal Systems 7th Workshop (DCFS 2005)*, Como, Italy, 2005, 182–193

[62] G. Jirásková, A. Okhotin: On the state complexity of star of union and star of intersection, *Turku Center for Computer Science TUCS Technical Report* 825, 2007

[63] G. Jirásková, A. Okhotin: On the state complexity of operations on two-way finite automata, in: *Proceedings of the 12th Developments in Language Theory (DLT 2008)*, Springer, LNCS 5257, 2008, 443–454

[64] D. Johnson: Fast allocation algorithms, in: *Proceedings of 13th Annual IEEE Symposium on Switching and Automata Theory*, 1972, 144–154

[65] D. Johnson: Near-optimal bin packing algorithms, PhD Dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1973

[66] B. Krawetz, J. Lawrence and J. Shallit: State complexity and the monoid of transformations of a finite set, *International Journal of Foundations of Computer Science*, vol. 16 no. 3 (2005) 547–563

[67] D. Krieger, A. Miller, N. Rampersad, B. Ravikumar and J. Shallit: Decimations of languages and state complexity, *Theoretical Computer Science*, vol. 410 no. 24-25 (2008) 2401–2409

[68] J. Lee, J. Shallit: Regular expressions: enumeration and state complexity, in: *Proceedings of 9th International Conference on Implementation*

and Application of Automata (CIAA 2004), Springer, LNCS 3317, 2005, 2–22

[69] E. Leiss: Succinct representation of regular languages by boolean automata II, *Theoretical Computer Science*, vol. 13 (1981) 323–330

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

[71] H. Lu, S. Yu: Are statecharts finite automata? in: *Proceedings of 14th International Conference on Implementation and Application of Automata (CIAA 2009)*, Springer, LNCS 5642, 2009, 258–261

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

[73] B. Mirkin: On dual automata, *Cybernetics and Systems Analysis*, vol. 2 no. 1 (1966) 6–9

[74] M. Mohri: On some applications of finite-state automata theory to natural language processing, *Natural Language Engineering*, vol. 2 (1996) 61–80

[75] D. Morales, F. Almeida, C. Rodriguez, J. Roda, I. Coloma and A. Delgado: Parallel dynamic programming and automata theory, *Parallel Computing*, vol. 26 (2000) 113–134

[76] C. Nicaud: Average state complexity of operations on unary automata, *Symposium on Mathematical Foundations of Computer Science* (M. Kutylowski, L. Pacholski, and T. Wierzbicki, Eds.), LNCS 1672, 1999, 231–240

[77] A. Okhotin: State complexity of linear conjunctive grammars, *Journal of Automata, Languages and Combinatorics*, vol. 9 no. 2-3 (2004) 365–381

[78] F. Pereira, M. Riley: Speech recognition by composition of weighted finite automata, in: *Finite-State Language Processing*, MIT Press, 1996, pp. 431–453

[79] X. Piao, K. Salomaa: Operational state complexity of nested word automata, *Theoretical Computer Science*, vol. 410 no. 35 (2009) 3290–3302

[80] X. Piao, K. Salomaa: Operational state complexity of deterministic unranked tree automata, in: *Proceedings of Descriptional Complexity of Formal Systems 12th Workshop (DCFS 2010)*, Saskatoon, Canada, 2010, 181–192

[81] X. Piao, K. Salomaa: Transformations between different types of unranked bottom-up tree automata, in: *Proceedings of Descriptional Complexity of Formal Systems 12th Workshop (DCFS 2010)*, Saskatoon, Canada, 2010, 193–204

[82] Giovanni Pighizzini: Unary language concatenation and its state complexity, in: *Proceedings of 5th International Conference on Implementation and Application of Automata (CIAA 2000)*, Springer, LNCS 2088, 2001, 252–262

[83] G. Pighizzini, J. Shallit: Unary language operations, state complexity and Jacobsthal's function, *International Journal of Foundations of Computer Science*, vol. 13 (2002) 145–159

[84] E. Pribavkina, E. Rodaro: State complexity of prefix, suffix, bifix and infix operators on regular languages, in: *Proceedings of the 14th Developments in Language Theory (DLT 2010)*, Springer, LNCS 6224, 2010, 252–262

[85] D. Qiu: Research report: state complexity of operations on two-way quantum finite automata, *Computing Research Repository*, (2008) abs/0807.0476

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

[87] N. Rampersad, N. Santean, J. Shallit and B. Ravikumar: State complexity of unique rational operations, *Theoretical Computer Science*, vol. 410 no. 24-25 (2008) 2431–2441

[88] D. Raymond, D. Wood and S. Yu (Eds.): Automata Implementation, *Proceedings of 1st International Workshop on Implementing Automata (WIA 1996)*, LNCS 1260, 1996

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

[90] A. Salomaa: Theory of Automata, Pergamon, New York, 1969

[91] A. Salomaa: Formal Languages, Academic Press, New York, 1973

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

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

[94] K. Salomaa: State complexity of nested word automata, in: *Proceedings of 3rd International Conference on Language and Automata Theory and Applications (LATA 2009)*, Springer, LNCS 5457, 2009, 59–70

[95] K. Salomaa, S. Yu: NFA to DFA transformation for finite languages over arbitrary alphabets, *Journal of Automata, Languages and Combinatorics*, vol. 2 no. 3 (1997) 177–186

[96] K. Salomaa, P. Schofield: State complexity of additive weighted finite automata, *International Journal of Foundations of Computer Science*, vol. 18 no. 6 (2007) 1407–1416

[97] K. Salomaa, S. Yu: On the state complexity of combined operations and their estimation, *International Journal of Foundations of Computer Science*, vol. 18 no. 4 (2007) 683–698

[98] M. Scott: Programming Language Pragmatics, Elsevier, New York, 2006

[99] J. Shallit: State complexity and Jacobsthal's function, in: *Proceedings of 5th International Conference on Implementation and Application of Automata (CIAA 2000)*, Springer, LNCS 2088, 2001, 272–278

[100] J. Shallit: New directions in state complexity, in: *Proceedings of Descriptional Complexity of Formal Systems 8th Workshop (DCFS 2006)*, Las Cruces, USA, 2006, 41–45

[101] J. Shallit: A Second Course in Formal Languages and Automata Theory, Cambridge University Press, USA, 2009

[102] A. Turing: On computable numbers, with an application to the entscheidungsproblem, in: *Proceedings of the London Mathematical Society*, Ser. 2 vol. 42, 1937, 230–265

[103] D. Wood, S. Yu (Eds.): Automata implementation, in: *Proceedings of 1st International Workshop on Implementing Automata (WIA 1996)*, LNCS 1260, 1996

[104] S. Yu: Regular languages, in: *Handbook of Formal Languages*, vol. 1,(G. Rozenberg, A. Salomaa, Eds.), Springer, 1997, pp. 41–110

[105] S. Yu: State complexity of regular languages, *Journal of Automata, Languages and Combinatorics*, vol. 6 no. 2 (2001) 221–234

[106] S. Yu: State complexity of finite and infinite regular languages, in: *Bulletin of the EATCS*, no. 76 (2002) 142–152

[107] S. Yu: State complexity: recent results and open problems, *Fundamenta Informaticae*, vol. 64 no. 1-4 (2005) 471–480

[108] S. Yu: On the state complexity of combined operations, in: *Proceedings of 11th International Conference on Implementation and Application of Automata (CIAA 2006)*, Springer, LNCS 4094, 2006, 11–22

[109] S. Yu: Lecture notes for CS 3331a Foundations of Computer Science, The University of Western Ontario, 2010

[110] S. Yu, Q. Zhuang: On the state complexity of intersection of regular languages, *ACM SIGACT News*, vol. 22 no. 3 (1991) 52–54

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

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

VITA

NAME:	Yuan Gao
PLACE OF BIRTH:	Liaoning, P.R.China
YEAR OF BIRTH:	1983
POST SECONDARY EDUCATION AND DEGREES:	The University of Western Ontario London, Canada 2005-2006 M.Sc.
	Northeastern University Liaoning, P.R.China 2001-2005 B.Sc.
HONOURS AND AWARDS:	Western Graduate Thesis Research Awards Fund 2010
	1st place - University of Western Ontario Research in Computer Science (UWORCS) 2010
	Western Graduate Thesis Research Awards Fund 2009
	1st place - University of Western Ontario Research in Computer Science (UWORCS) 2007
	Western Graduate Research Scholarships Computer Science 2005-2010

Excellent Theses Award of Northeastern University
2005

The Second Prize in China Undergraduate Mathematical Contest in Modeling
2004

RELATED WORK EXPERIENCE:

Research Assistant / Teaching Assistant
The University of Western Ontario
2005-2006

EDITED PROCEEDINGS:

Proceedings of 14th International Conference of Developments in Language Theory,
with H. Lu, S. Seki and S. Yu,
Springer, LNCS 6224, 2010

PUBLICATIONS:

B. Cui, Y. Gao, L. Kari and S. Yu:
State complexity of two combined operations: catenation-Star and catenation-reversal,
International Journal of Foundations of Computer Science, submitted by invitation for publication

B. Cui, Y. Gao, L. Kari and S. Yu:
State complexity of two combined operations: reversal-catenation and star-catenation,
Computing Research Repository (2010)
arXiv:1006.4646v1

Y. Gao, S. Yu:
State complexity of union and intersection combined with star and reversal,
Computing Research Repository (2010)
arXiv:1006.3755v1

Y. Gao, K. Salomaa and S. Yu:
Transition complexity of incomplete DFAs,
in: *Proceedings of Descriptional Complexity of Formal Systems 12th Workshop (DCFS 2010)*, Saskatoon, Canada, 2010, 123-134

B. Cui, Y. Gao, L. Kari and S. Yu:
State complexity of catenation combined with star and reversal,
in: *Proceedings of Descriptional Complexity of Formal Systems 12th Workshop (DCFS 2010)*, Saskatoon, Canada, 2010, 74-85

B. Cui, Y. Gao, L. Kari and S. Yu:
State complexity of catenation combined with union and intersection,
in: *Proceedings of the 15th International Conference on Implementation and Application of Automata (CIAA 2010)*, Winnipeg, Canada, 2010, 89-98

Y. Gao, S. Yu:
State complexity approximation of combined operations, submitted to *Journal of Computer and System Sciences* (2009)

Y. Gao, S. Yu:
State complexity approximation,
in: *Proceedings of Descriptional Complexity of Formal Systems 11th Workshop (DCFS 2009)*, Magdeburg, Germany, 2009, 163-174

Z. Ésik, Y. Gao, G. Liu and S. Yu:
Estimation of state complexity of combined operations,
Theoretical Computer Science, vol. 410 no. 35 (2009) 3272-3280

Z. Ésik, Y. Gao, G. Liu and S. Yu:
Estimation of state complexity of combined operations (preliminary version),
Proceedings of Descriptional Complexity of Formal Systems 10th Workshop (DCFS 2008), Charlottetown, Canada, 2008, 168-181

Y. Gao, K. Salomaa and S. Yu:
The state complexity of two combined operations: star of catenation and star of reversal,
Fundamenta Informaticae, vol. 83 no. 1-2 (2008) 75-89

Y. Gao, Z. Ésik, G. Liu and S. Yu:
State complexity of intersection, union and catenation on k regular languages,
Technical Report 715, Department of Computer Science, the University of Western Ontario, 2008

Y. Gao, K. Salomaa and S. Yu:
State complexity of star of catenation and reversal,
Proceedings of Descriptional Complexity of Formal Systems 8th Workshop (DCFS 2006), Las Cruces, USA, 2006, 153-164

Y. Gao, G. Yu, F. Li and Y. Yu:
A data stream operator scheduling algorithm for multiple goals,
Journal of Huazhong University of Science And Technology (Nature Science) vol. 33 no. s1 (2005) 232-234

Y. Gao, G. Yu, F. Li and Y. Yu:
A data stream operator scheduling algorithm for multiple goals (preliminary version),
Proceedings of the 2nd China National Computer Conference, Wuhan, China, 2005