Western University Scholarship@Western

Digitized Theses

Digitized Special Collections

2011

State Complexity of Combined Operations on Finite Languages

Jahedur Rahman Chowdhury

Follow this and additional works at: https://ir.lib.uwo.ca/digitizedtheses

Recommended Citation

Chowdhury, Jahedur Rahman, "State Complexity of Combined Operations on Finite Languages" (2011). *Digitized Theses*. 3508. https://ir.lib.uwo.ca/digitizedtheses/3508

This Thesis is brought to you for free and open access by the Digitized Special Collections at Scholarship@Western. It has been accepted for inclusion in Digitized Theses by an authorized administrator of Scholarship@Western. For more information, please contact wlswadmin@uwo.ca.

State Complexity of Combined Operations on Finite Languages

(Thesis format: Monograph)

by

Jahedur Rahman Chowdhury

Graduate Program in Computer Science

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science

1

r

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

©Jahedur Rahman Chowdhury 2011

Abstract

State complexity is a descriptive complexity measure for regular languages. It is a fundamental topic in automata and formal language theory. The state complexity of a regular language is the number of states in the minimal complete deterministic finite automaton accepting the language. During the last few decades, many publications have focused and studied the state complexity of many individual as well as combined operations on regular languages. Also, the state complexity of some basic operations on finite languages has been studied. But until now there has been no study on the state complexity of combined operations on finite languages.

In this thesis, we will first study the state complexity of the combined operation, star of union, on finite languages and give an exact bound. Then we will investigate the state complexity of star of catenation and show its approximation with a good ratio bound and finally, we will prove an upper bound for star of intersection.

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

Dedication

To my parents

Contents

Ce	ertifi	cate of Examination	ii	
Al	bstra	ct	iii	
Ac	cknov	wledgements	iv	
De	edica	tion	v	
Ta	ble o	of Contents	vi	
\mathbf{Li}	st of	Tables	x	
Li	st of	Figures	xi	
1	Intr	oduction	1	
	1.1	Automata and Formal Language Theory	1	
	1.2	State Complexity	2	
1.3 Motivation of State Complexity Study			3	
	1.4 Why Study State Complexity of Combined Operations on Finite Lan-			
		guages	4	
	1.5	State Complexity Approximation	4	
	1.6	Organization of the Thesis	5	
2	Bas	ic Definitions and Notations	7	
	2.1	Alphabets, Strings and Languages	7	

Contents

Ce	ertifi	cate of Examination	ii	
A	bstra	\mathbf{ct}	iii	
A	cknow	wledgements	iv	
De	edica	tion	v	
Ta	ble o	of Contents	vi	
\mathbf{Li}	st of	Tables	x	
Li	st of	Figures	xi	
1	Intr	oduction	1	
	1.1	Automata and Formal Language Theory	1	
	1.2	State Complexity	2	
1.3 Motivation of State Complexity Study		3		
	1.4 Why Study State Complexity of Combined Operations on Finite Lan-			
		guages	4	
	1.5	State Complexity Approximation	4	
	1.6	Organization of the Thesis	5	
2	Bas	ic Definitions and Notations	7	
	2.1	Alphabets, Strings and Languages	7	

		2.1.1	Alphabets	7
		2.1.2	Strings	7
		2.1.3	Languages	8
	2.2	Regul	ar Languages	10
		2.2.1	Regular Language	10
		2.2.2	Finite Language	10
	2.3	Determ	ministic Finite Automata	10
	2.4	Nonde	eterministic Finite Automata	13
	2.5	NFA t	o DFA Conversion	13
	2.6	Minim	nization of DFA	14
	2.7	Hoper	oft's Minimization Algorithm	15
3	Bac	kgroui	nd: State Complexity Research	16
	3.1	From	1950s to Early 1990s	16
	3.2	From	Early 1990s to 2005	17
	3.3	From	2006 to Now	18
4	Stat	te Con	nplexity: Previous Results	19
	4.1	State	Complexity of Basic Operations on Regular Languages	19
		4.1.1	State Complexity of Catenation	19
		4.1.2	State Complexity of Star	20
		4.1.3	State Complexity of Union and Intersection	20
		4.1.4	State Complexity of Reversal	21
		4.1.5	Summary Table	21
	4.2	State	Complexity of Basic Operations on Finite Languages	22
		4.2.1	State Complexity of Catenation	22
		4.2.2	State Complexity of Star	22
		4.2.3	State Complexity of Union and Intersection	23
		4.2.4	State Complexity of Reversal	23

		4.2.5	Summary Table	24
	4.3	State (Complexity of Combined Operations on Regular Languages	24
		4.3.1	State Complexity of Star of Union	24
		4.3.2	State Complexity of Star of Intersection	25
		4.3.3	State Complexity of Reversal of Union	25
		4.3.4	State Complexity of Reversal of Intersection	26
		4.3.5	State Complexity of Reversal of Catenation and Reversal of Star	26
		4.3.6	Summary Table	26
5	Stat	te Com	plexity of Star of Union Operation on Finite Languages	28
	5.1	Upper	Bound	29
	5.2	Lower	Bound: Worst-case Example	36
		5.2.1	Lower bound when $m \leq 3$ and $n \leq 3$	47
		5.2.2	Lower bound when $m = 2$ and $n > 3$ or $m > 3$ and $n = 2$	47
6	Stat	te Con	plexity of Star of Catenation Operation on Finite Lan-	
6	Stat gua	te Con ges	plexity of Star of Catenation Operation on Finite Lan-	48
6	Stat guas 6.1	te Con ges Upper	bound	48 48
6	Stat gua 6.1 6.2	te Con ges Upper Lower	bound	48 48 55
6	Stat gua 6.1 6.2 6.3	te Com ges Upper Lower Ratio	bound Star of Catenation Operation on Finite Lan- bound Bound: Worst-case Example Bound Bound	48 48 55 64
6	Stat gua 6.1 6.2 6.3 Stat	te Com ges Upper Lower Ratio	bound Star of Catenation Operation on Finite Lan- bound Bound: Worst-case Example Bound Bound Bou	48 48 55 64
6 7	Stat gua 6.1 6.2 6.3 Stat gua	te Con ges Upper Lower Ratio te Con ges	boundBound: Worst-case ExampleBoundBound	 48 48 55 64 66
6 7	Stat gua 6.1 6.2 6.3 Stat gua 7.1	te Com ges Upper Lower Ratio te Com ges Upper	applexity of Star of Catenation Operation on Finite Lan- bound	 48 48 55 64 66 66
6	Stat guay 6.1 6.2 6.3 Stat guay 7.1 7.2	te Com ges Upper Lower Ratio te Com ges Upper Lower	aplexity of Star of Catenation Operation on Finite Lan- bound	 48 48 55 64 66 66 80
6 7 8	Stat guay 6.1 6.2 6.3 Stat guay 7.1 7.2 Con	te Com ges Upper Lower Ratio te Com ges Upper Lower	applexity of Star of Catenation Operation on Finite Lan- bound	 48 48 55 64 66 66 80 84
6 7 8	Stat guas 6.1 6.2 6.3 Stat guas 7.1 7.2 Com 8.1	te Com ges Upper Lower Ratio te Com ges Upper Lower nclusion Conch	applexity of Star of Catenation Operation on Finite Lan- bound	 48 48 55 64 66 66 80 84 84

Bibliography

Curriculum Vitae

91

List of Tables

4.1	State complexity of basic operations on regular languages	21
4.2	State complexity of basic operations on finite languages	24
4.3	State complexity of combined operations on regular languages	27

List of Figures

5.1	DFA A of 6 states for Theorem 3 \ldots	37
5.2	DFA B of 6 states for Theorem 3 \ldots \ldots \ldots \ldots \ldots	38
6.1	DFA A of m states for Theorem 5	56
6.2	DFA B of n states for Theorem 5 $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$	56
7.1	DFA A of 6 states to get the lower bound for star-of-intersection	82
7.2	DFA B of 6 states to get the lower bound for star-of-intersection	83

Chapter 1 Introduction

1.1 Automata and Formal Language Theory

Automata theory is one of the oldest research topics in computer science. It includes the study and application of automata, such as Nondeterministic Finite Automata (NFA), Deterministic Finite Automata (DFA), Push Down Automata (PDA), Nondeterministic Push Down Automata (NPDA), etc. Research in this area started from 1930's. Nowadays, a large number of applications are using finite automata. Automata theory is helpful in the design and construction of various kinds of softwares [13].

Automata can be used as the representation of formal languages. Formal languages are useful in natural languages processing, compiler design, and programming languages, etc. The family of regular languages is a kind of formal language which is often used in various practical applications such as *vi*, *emacs*, and *Perl*. Furthermore, researchers developed a number of software libraries for manipulating formal language objects with the emphasis on regular languages [7].

1.2 State Complexity

State complexity is a descriptive complexity measure for regular languages. It is different from the time and space complexity but gives a lower bound for them. A large number of publications have focused on the state complexity issues [1, 2, 3, 6, 7, 8, 9, 14, 15, 16, 19, 23, 27, 30, 35, 36, 32]. It is a fundamental topic in automata and formal language theory. With respect to different automaton model, there are two kinds of state complexity, namely, deterministic state complexity and nondeterministic state complexity.

Deterministic state complexity

The deterministic state complexity of a regular language L, denoted by sc(L), is the number of states in the minimal complete deterministic finite automaton (DFA) accepting the language. In general, by state complexity we refer to deterministic state complexity. The state complexity of a class \mathcal{L} of regular languages, denoted $sc(\mathcal{L})$, is the supremum among all sc(L), $L \in \mathcal{L}$.

The state complexity of an operation on regular languages is the number of states in the minimal DFA that accepts the language resulting from that operation. For example, the state complexity of intersection between an m-state DFA language (i.e., a language accepted by an m-state complete DFA) and an n-state DFA language is mn; which means mn is the number of states in the minimal DFA that accepts the resulting language from intersection of m-state DFA and n-state DFA in the worst case [32].

Nondeterministic state complexity

The nondeterministic state complexity of a regular language L, denoted by nsc(L), is the number of states of a minimum-state nondeterministic finite automaton (NFA) accepting the language. The nondeterministic state complexity of an operation on regular languages is similarly defined.

1.3 Motivation of State Complexity Study

In the 60s and 70s, the number of states of finite automata used in applications was usually small. Finite automata were used in different applications then, such as switching theory, testing circuits, and pattern matching, etc. There was no strong motivation from the practice to study the state complexity issues in general then. In recent years, there have been many new applications of finite automata, e.g., in natural language processing, programming languages, software engineering, image generation and encoding, etc. Also, finite automata are becoming popular and attractive for solving problems in a wide range of computational domains, including speech and handwriting recognition, optical character recognition, encryption, image compression and indexing. There are some advantages of using finite automata in these applications:

- Firstly, they are easy to implement, e.g. a simple automaton can be represented by a matrix, which is useful in many cases.
- Secondly, they are fast to traverse, especially in the case of deterministic devices.
- Finally, they are mathematically elegant since they are closed under various useful operations, such as catenation, union, intersection, and Kleene star.

In spite of these advantages, almost all of these applications use finite automata with a large number of states, which results in large scale machines causing obstacles in terms of time and space complexity. For example, in natural language and speech processing the Bell Labs multilingual Text-To-Speech (TTS) system needs 26.6 mbytes for German, 30.0 for French, and 39.0 for Mandarin. The size of these modules becomes a serious hurdle when the system is to be imported into a special-purpose hardware

device with limited memory [17]. Sometimes, even the machines can be unmanageable. However, knowing the state complexities it is possible to design manageable and efficient machines as well as reduce the time and space complexity. So, the study of state complexity problems is strongly motivated by practical applications [33].

1.4 Why Study State Complexity of Combined Operations on Finite Languages

There are two types of regular languages: finite languages and infinite languages. Many applications of regular languages use essentially finite languages. State complexity of basic operations on finite languages has been studied in [2]. Later the state complexity of union and intersection of finite languages was studied in [7]. But until now there has been no study about state complexity of any combined operations on finite languages. There are no loops and backward transitions in finite automata accepting finite languages except in the sink state. Therefore, the state complexity of combined operations on finite languages will be different from, or rather less than, the state complexity of the same combined operations on regular languages. Moreover, similar to the regular languages, the state complexity of combined operations on finite languages is usually not the same as their mathematical composition of the state complexity of individual operations on finite languages. So, the state complexity of combined operations on finite languages should be studied individually.

1.5 State Complexity Approximation

The number of combined operations on regular languages is not limited as the number of individual operations on regular languages. There are two major problems concerning the state complexities of combined operations.

- 1. The state complexities of many combined operations are extremely difficult to compute.
- 2. A large proportion of results that have been obtained are pretty complex and impossible to comprehend.

Also there is no general algorithm for a given operation and a set of regular languages that can compute the state complexity of the operation on the set of regular languages. Therefore, a good approximation of state complexities of combined operations are useful in many automata applications [34].

An approximation of a state complexity is an estimate of the state complexity with a ratio bound clearly defined. The ratio bound gives a precise measurement on the quality of the estimate. A state complexity approximation is close to the exact state complexity and normally not equal to it. The ratio bound shows the error range of the approximation. In [34] the state complexity approximation of combined operations has been proposed and studied.

1.6 Organization of the Thesis

In this thesis, we will investigate the state complexity of the star of union operation on finite languages. We will also study the approximation of the state complexity of star of catenation and the state complexity of star of intersection.

This thesis consists of eight chapters. The first chapter gives an introduction and the second chapter describes some basic definitions and notations used in this thesis. Chapter 3 describes some backgrounds on state complexity research.

In Chapter 4, we will describe previous results of the state complexity of some basic operations on regular and finite languages. Also, we will describe earlier results of the state complexity of some combined operations on regular languages.

In Chapter 5, we will study the state complexity of star of union operation on finite languages. Then, in Chapter 6, we will study the state complexity of star of catenation operation on finite languages. We also study the state complexity of star of intersection operation on finite languages in Chapter 7. Finally, in Chapter 8, we will give a conclusion and describe future research of state complexity issues.

Chapter 2 Basic Definitions and Notations

We will describe some basic definitions and notations that will be used in this thesis. But for a detailed and complete background knowledge, the reader may refer to [13, 28, 31].

2.1 Alphabets, Strings and Languages

2.1.1 Alphabets

An alphabet is a finite, nonempty set of symbols, denoted by Σ . Some examples of alphabet are as follows.

- 1. $\Sigma = \{a, b, ..., z\}$, the set of all lower-case letters.
- 2. $\Sigma = \{a, .., z, A, ..., Z\}$, the set of all upper-case and lower-case letters.

2.1.2 Strings

A string or word is a finite sequence of symbols chosen from some alphabet Σ . For example, 0101 is a string from the binary alphabet $\Sigma = \{0, 1\}$. Σ^* is the set of all the strings whose symbols are chosen from Σ . The set of nonempty string from alphabet Σ is denoted by Σ^+ .

Length of a string

The length of a string x is the number of symbols in x. For any string $x \in \Sigma^*$, |x| denotes the length of x and $|x|_a$, for some $a \in \Sigma$, denotes the number of occurrences of a in x.

Empty string

The empty string is the string with zero occurrences of symbols. The empty string is denoted by ε . Therefore, $|\varepsilon| = 0$ and it is over any alphabet.

Catenation of strings

The catenation of two string x and y is denoted by xy and it is the string formed by the string x followed by the string y. So, |xy| = |x| + |y|.

Reversal of a string

Let $x = a_1 a_2 \dots a_{n-1} a_n$, $n \ge 0$ be a string over the alphabet Σ . The reversal of x, denoted by x^R , is the string $a_n a_{n-1} \dots a_2 a_1$.

2.1.3 Languages

A language L over an alphabet Σ is a set of strings which are chosen from Σ^* . The empty language is denoted by ϕ . The universal language over Σ , which is the language consisting of all strings over Σ , is Σ^* . For a language L, |L| denotes the cardinality of L, i.e., the number of strings in L. If an alphabet Σ contains only one letter, i.e. $|\Sigma| = 1$, then a language over Σ is referred to as a unary language. Similarly, if $|\Sigma| = 2$ then the language is referred to as a binary language.

Catenation of languages

The catenation of two languages $L_1, L_2 \subseteq \Sigma^*$ is denoted by L_1L_2 where

$$L_1L_2 = \{ w_1w_2 \mid w_1 \in L_1 \text{ and } w_2 \in L_2 \}.$$

Union of languages

The union of two languages $L_1, L_2 \subseteq \Sigma^*$, denoted by $L_1 \cup L_2$, consists of all the strings which are contained either in L_1 or L_2 , i.e.,

$$L_1 \cup L_2 = \{ w \mid w \in L_1 \text{ or } w \in L_2 \}.$$

Intersection of languages

The intersection of two languages $L_1, L_2 \subseteq \Sigma^*$, denoted by $L_1 \cap L_2$, consists of all the strings which are contained in both languages L_1 and L_2 , i.e.,

$$L_1 \cap L_2 = \{ w \mid w \in L_1 \text{ and } w \in L_2 \}.$$

Kleene star of a language

For an integer $n \ge 0$ and a language L, the n^{th} power of L, denoted by L^n , is defined by (i) $L^0 = \{\varepsilon\}$, and (ii) $L^n = L^{n-1}L$, for n > 0. The Kleene star of a language L is denoted by L^* where

$$L^* = \bigcup_{i=0}^{\infty} L^i.$$

Complement of a language

The complement of a language L with respect to a given alphabet Σ , denoted by $\Sigma^* - L$, consists of all the strings over the alphabet that are not in the language, i.e.,

$$\Sigma^* - L = \{ w \, | \, w \notin L \}.$$

Reversal of a language

The reversal of a language L, denoted by L^R , is the language consisting of the reversal

of all string in L, i.e.,

$$L^R = \{ w^R \, | \, w \in L \}$$

2.2 Regular Languages

2.2.1 Regular Language

Regular language is a type of formal language which are recognized by finite automata. Regular languages are closed under the operations of union, catenation and Kleene star. The set of regular languages over an alphabet Σ is defined recursively as follows:

- The empty language \emptyset is a regular language.
- The empty string language $\{\varepsilon\}$ is a regular language.
- For each $a \in \Sigma$, the singleton language $\{a\}$ is a regular language.
- If L_1 and L_2 are regular languages, then $L_1 \cup L_2$, L_1L_2 , and L_1^* are regular languages.
- No other languages over Σ are regular.

2.2.2 Finite Language

A language is finite if it consists of a finite number of strings, i.e., a finite language is a set of n strings for some nonnegative number n.

2.3 Deterministic Finite Automata

A deterministic finite automaton (DFA) can be defined as $A = (Q, \Sigma, \delta, q_0, F)$, where

• Q is the finite set of states,

- Σ is the finite alphabet,
- $\delta: Q \times \Sigma \longrightarrow Q$ is the transition function,
- $q_0 \in Q$ is the start state, and
- $F \subseteq Q$ is the set of final states.

The cardinality of Q, denoted by |Q|, is total number of states in DFA A. Similarly, the cardinality of F, denoted by |F|, is total number of final states in DFA A.

Complete DFA

A complete DFA is such a DFA where the transition function δ is defined for each of the state $q \in Q$ and each of the symbol $a \in \Sigma$, i.e., δ is a total function.

The transition function δ can be extended to $\delta^* : Q \times \Sigma^* \longrightarrow Q$, where Σ^* is the set of all strings over the alphabet Σ including the empty string ε , $\delta^*(q, \varepsilon) = q$ and $\delta^*(q, ax) = \delta^*(\delta(q, a), x)$ for $q \in Q$, $a \in \Sigma$, and $x \in \Sigma^*$. In the following, we will use δ instead of δ^* if there is no confusion.

A string $w \in \Sigma^*$ is accepted by the DFA A if the state $\delta(q_0, w)$ is a final state of DFA A. The language accepted by DFA A is denoted by L(A), where

$$L(A) = \{ w \in \Sigma^* \mid \delta(q_0, w) \in F \}.$$

Equivalent states

Given a DFA $A = (Q, \Sigma, \delta, q_0, F)$, two states $p, q \in Q$ are said to be equivalent, denoted by $p \equiv_A q$, if and only if for every $w \in \Sigma^*$,

$$\delta(p,w) \in F \Longleftrightarrow \delta(q,w) \in F.$$

Two states that are not equivalent are called distinguishable.

Equivalence relation

A binary relation R on a set S is a set of pairs of elements in S. If (a, b) is in R, then we denote the relation between a and b as aRb. A relation which is reflexive, symmetric, and transitive is called an equivalence relation. An important property of equivalence relation is that if R is an equivalence relation on the set S then we can divide S into k disjoint subsets, called equivalence classes, for some k between 1 and infinity, inclusive, such that aRb if and only if a and b are in the same subset. An equivalence relation \sim on strings of symbols from some alphabet Σ is said to be right invariant if $\forall x, y \in \Sigma^*$ with $x \sim y$ and $\forall w \in \Sigma^*$ we have $xw \sim yw$.

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 equivalence relation R be defined by: xRy if and only if for all $z \in \Sigma^*$, $xz \in L$ exactly when $yz \in L$. Then R is of finite index.

Minimal DFA

Myhill-Nerode theorem implies that there is a unique minimum-state DFA for each regular language. We determine the number of states in the minimal DFA based on the Myhill-Nerode equivalence relation. A DFA is called minimal if it has the number of equivalence classes equal to the number of states, i.e., if $L \subseteq \Sigma^*$ is a regular language then the number of states in the minimal DFA recognizing L is equal to the number of equivalence classes of the relation \sim_L . In another way, we can say that a DFA $A = (Q, \Sigma, \delta, q_0, F)$ is minimal if for every other automaton $A' = (Q', \Sigma, \delta', q'_0, F')$ such that L(A) = L(A'), we have $|Q| \leq |Q'|$ [2]. In particular, every regular language has a unique minimal DFA up to isomorphism.

2.4 Nondeterministic Finite Automata

A nondeterministic finite automaton (NFA) can be defined as $N = (Q, \Sigma, \delta, q_0, F)$, where

- Q is the finite set of states,
- Σ is the finite alphabet,
- δ: Q × (Σ ∪ {ε}) → 2^Q is the transition function (2^Q denotes the power set of Q),
- $q_0 \in Q$ is the start state, and
- $F \subseteq Q$ is the set of final states.

A string $w \in \Sigma^*$ is accepted by the NFA N if the set of states $\delta(q_0, w)$ contains an accepting state of the NFA N. The language accepted by NFA N is denoted by L(N), where

$$L(N) = \{ w \in \Sigma^* \, | \, \delta(q_0, w) \cap F \neq \phi \}.$$

2.5 NFA to DFA Conversion

Any NFA $M = (Q, \Sigma, \delta, q_0, F)$ can be converted to an equivalent DFA $M' = (2^Q, \Sigma, \delta', q'_0, F')$ using an algorithm known as the "subset construction" [24] in the following way:

- 1. Every state of the DFA M' is a subset of the state set Q.
- 2. The initial state of the DFA M' is $\{q_0\}$.

- 3. A state $R \in 2^Q$ is an accepting state of the DFA M' if it contains an accepting state of the NFA M.
- 4. The transition function δ' is defined by $\delta'(R, a) = \bigcup_{\forall r \in R} \delta(r, a)$ for any state $R \in 2^Q$ and any symbol $a \in \Sigma$.

The DFA M' need not be minimal since some of its states may be unreachable or equivalent [14].

2.6 Minimization of DFA

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA. We can minimize A as follows:

- 1. Eliminate all the states which cannot be reached from the initial state.
- 2. Partition the remaining states into the blocks of equivalent states by using table-filling algorithm [Algorithm 1].
- 3. Construct the minimum-state equivalent DFA B by using the blocks as its states; the set of final states of B is the set of blocks containing final states of A.

The running time of this algorithm is $O(n^2)$.

```
Algorithm 1: Table-filling Algorithm
   Data: Deterministic finite automaton
   Result: Minimal automaton
 1 Remove inaccessible states
2 Mark all pairs (p,q) where p \in F and q \notin F
 3 repeat
      forall the non-marked pairs (p,q) do
 \mathbf{4}
          forall the symbol a do
 5
             if the pair \delta(p, a), \delta(q, a) is marked then
 6
                 mark (p,q)
7
             end
8
          end
9
      end
10
11 until no new pairs are marked;
12 Construct the reduced automaton
```

2.7 Hopcroft's Minimization Algorithm

Hopcroft's DFA minimization algorithm is efficient and faster than previous algorithm. Also, it uses different partitioning technique to minimize the DFA. The running time of this algorithm is $O(n \log n)$ [12].

Chapter 3 Background: State Complexity Research

During the last few decades, many publications have been focused on the state complexities of different kinds of operations on regular languages. We can categorize the history of the state complexity research into three periods as follows.

3.1 From 1950s to Early 1990s

In 1959, Rabin and Scott [24] proved that the number of states in a DFA that is transformed from an *n*-state NFA is limited to 2^n . Later, in 1971, F. Moore proved that the bound is tight. In the 1960s, Arto Salomaa studied several state complexity issues [26]. In 1970, Maslov [20] studied the state complexities of several basic operations on regular languages which include union, concatenation, and star without rigorous proofs. The results obtained by him were almost accurate except that there were minor problems in his results. E. Leiss [18] studied succinct representation of regular languages in early 1980. J.C. Birget, in 1992, studied the state complexity of multiple intersection and union of regular languages [1]. Some other scattered results concerning state complexity have been obtained during this period of time.

3.2 From Early 1990s to 2005

In 1994, Yu, Zhang, and Salomaa systematically studied the state complexity problems of basic operations on regular languages over a general alphabet as well as over a one-letter alphabet [36]. They studied the state complexity of catenation, star, left quotient, right quotient, reversal, union and intersection operations.

Nicaud [22] was the first who investigated the average state complexity of operations on unary languages.

State complexity of some basic operations on finite languages was first studied in 2001 by Campeanu, Culik, Salomaa and Yu [2]. They have investigated the state complexity of star, concatenation and reversal operations on finite languages.

J. Shallit studied the state complexity of the intersection of regular languages over a one-letter alphabet in 2001 [30]. Pighizzini studied the state complexity of concatenation over a one-letter alphabet in the same year [23].

The state complexity of nondeterministic finite automata was studied by Holzer and Kutrib in 2002 [10, 11]. In 2005, Mera and Pighizzini investigated the nondeterministic state complexity of unary regular languages and that of their complements [21].

Domaratzki studied the state complexity on proportional removals of regular languages in 2002. In the same year, Campeanu, Salomaa and Yu obtained tight bound for the state complexity of shuffle of regular languages [3]. Jiraskova (one paper with Szabari) had several results on state complexity issues including the concatenation and complementation operations of finite automata [15, 16]. Many other results on state complexity have been obtained in this period.

3.3 From 2006 to Now

The state complexity of the standard combinations of basic operations has not been studied until early 2006. A. Salomaa, K. Salomaa and S. Yu studied the state complexity of two combined operations: star of union and star of intersection in 2007 [27]. Later in 2008, Gao, Salomaa, and Yu studied the state complexity of two other combined operations: star of catenation and star of reversal [6]. The state complexity of a number of other combinations of operations has been investigated until now, such as the state complexity of reversal of union, reversal of catenation etc.

Han, Salomaa, and Wood investigated nondeterministic state complexity of some basic operations for prefix-free regular languages in 2009 [8]. In the same year, the state complexity of some combined operations for prefix-free regular languages was also studied by Han, Salomaa, and Yu [9].

The state complexity of union and intersection of finite languages was studied in 2007 by Han and Salomaa [7].

The state complexity of combined operations and their estimations have been studied by Salomaa and Yu in 2007 [29]. Later in 2009, Esik, Gao, Lio and Yu studied the estimation of state complexity of some combined operations [5].

Chapter 4 State Complexity: Previous Results

4.1 State Complexity of Basic Operations on Regular Languages

4.1.1 State Complexity of Catenation

Let A be an m-state DFA and B be an n-state DFA defined on the same alphabet Σ , accepting the languages L(A) and L(B), respectively. Then there exists a DFA of $m2^n - k2^{n-1}$ states which accepts the language L(A)L(B), where k is the number of final states in DFA A such that 0 < k < m [36]. In [14] it was shown that the upper bound $m2^n - k2^{n-1}$ is tight for any integer k with 0 < k < m for three-letter alphabet.

In the case of $m \ge 1$ and n = 1, upper bound m was shown to be tight for a unary alphabet. In the case of m = 1 and $n \ge 2$, the worst case $2^n - 2^{n-1}$ was given by the catenation of two binary languages . Otherwise, the upper bound $m2^n - 2^{n-1}$ was proven to be tight for a three-letter alphabet [36]. For any integer n > 0, there exists a 2-state DFA language and an n-state DFA language such that any DFA accepting the catenation of the two languages needs at least 2^{n-1} states [25]. In [15], it was shown that the upper bound $m2^n - 2^{n-1}$ can also be reached by the catenation of two binary languages. However, in case of unary language, the optimal upper bound is mn that can be reached for any $m, n \ge 1$ such that m and n are relatively prime, i.e., (m, n) = 1 [36]. The unary case when m and n are not necessarily relatively prime was studied by Pighizzini and Shallit in [23, 30]. In this case, the tight bounds are given by the number of states in the noncyclic and in the cyclic parts of the resulting automata.

4.1.2 State Complexity of Star

If L(A) is the language accepted by a DFA A, then the (Kleene) star of L(A) is denoted by $(L(A))^*$. An example was given in [25], showing that any DFA accepting the star of an *n*-state DFA language needs at least 2^{n-1} states in some cases for n > 0. However, in [36], the above result has been improved and it was proven that $2^{n-1} + 2^{n-2}$ states are necessary in the worst case for a DFA to accept the star of an *n*-state DFA language for each n > 1. The proof was for binary languages. However, if the number of final states excluding the start state in an *n*-state DFA is k such that $k \ge 1$, then there exists a DFA of at most $2^{n-1} + 2^{n-k-1}$ states that accepts $(L(A))^*$. The optimal upper-bound for the number of states which is needed to accept the star of an *n*-state DFA language L over a one-letter alphabet is $(n-1)^2 + 1$ [36].

4.1.3 State Complexity of Union and Intersection

The state complexity of the union of an *m*-state DFA language L_1 and an *n*-state DFA language L_2 over a one-letter alphabet is mn where m and n are relatively prime. Similarly, the state complexity of the intersection of an *m*-state DFA language L_1 and an *n*-state DFA language L_2 over a one-letter alphabet is mn where (m, n) = 1 [36]. These bounds for union and intersection are also tight for languages over any arbitrary alphabet. However, these results are different when m and n are not relatively prime and also if we consider tail and cycle in the DFAs. Let A and A' be two unary DFAs and the transition diagram of A (resp. A') have a tail of size t and a cycle of size c(resp. t', c'). Then for all $t, t' \ge 0$ and $c, c' \ge 1$, the state complexity of intersection of L(A) and L(A') is max(t, t') + lcm(c, c'). The state complexity of union of L(A) of L(A') is also same, i.e., max(t, t') + lcm(c, c') and this upper bound is tight for both union and intersection [23, 30].

4.1.4 State Complexity of Reversal

Any DFA accepting the reversal of an *n*-state DFA language does not need more than 2^n states. In [4], a result on alternating finite automata implies that this upper bound can be reached in the case where *n* is in the form 2^k for some integer $k \ge 0$. The worst case state complexity 2^n can be reached by the reversal of a binary DFA language for all n > 0 [18]. However, in case of a unary regular language *L*, the state complexity of its reversal is *n* where *n* is number of states of a DFA accepting the language *L* [36].

4.1.5 Summary Table

We assume that L_1 is accepted by an *m*-state DFA and L_2 is accepted by an *n*-state DFA and m, n > 1. Table 4.1 shows the summary of the state complexity of some of the basic operations on regular languages.

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

Table 4.1: State complexity of basic operations on regular languages

4.2 State Complexity of Basic Operations on Finite Languages

4.2.1 State Complexity of Catenation

Here the following notation is used:

$$\binom{n}{\leq i} = \sum_{j=0}^{i} \binom{n-2}{j}$$

Let $A_1 = (Q_1, \Sigma, \delta_1, q_{0,1}, F_1)$ be a DFA accepting the finite language $L(A_1)$ and $A_2 = (Q_2, \Sigma, \delta_2, q_{0,2}, F_2)$ be a DFA accepting the finite language $L(A_2)$ where $|\Sigma| = k$, $|Q_1| = m$, $|Q_2| = n$, and $|F_1| = t$. Then there exists a DFA $A = (Q, \Sigma, \delta, q_0, F)$ such that $L(A) = L(A_1)L(A_2)$ and

$$|Q| \leq \sum_{i=0}^{m-2} \min\left\{k^{i}, \left(\begin{array}{c}n-2\\\leq i\end{array}\right), \left(\begin{array}{c}n-2\\\leq t-1\end{array}\right)\right\} + \min\left\{k^{m-1}, \left(\begin{array}{c}n-2\\\leq t\end{array}\right)\right\}.$$

If t > 0 is a constant then there exists a DFA $A = (Q, \Sigma, \delta, q_0, F)$ of $O(mn^{t-1} + n^t)$ states such that $L(A) = L(A_1)L(A_2)$. For k = 2 and $m+1 \ge n+2$, the upper bound is $(m-n+3)2^{n-2}-1$. This upper bound is reachable and the proof was given for a binary alphabet in [2].

4.2.2 State Complexity of Star

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA accepting the finite language L(A), where $q_0 \notin F$, $|F| = t \ge 2$, and $|Q| = n \ge 4$. Then there exists a DFA of at most $2^{n-3} + 2^{n-t-2}$ states that accepts $(L(A))^*$. If t = 1, then the DFA accepting $(L(A))^*$ needs at most n-1 states. Otherwise an upper bound is $2^{n-3} + 2^{n-4}$. This upper bound is reachable and the proof was shown for three-letter alphabet in [2].

4.2.3 State Complexity of Union and Intersection

Let A be an m-state DFA and B be an n-state DFA accepting the finite languages L(A) and L(B), respectively. Then the DFA accepting the language $L(A) \cup L(B)$ needs at most mn - (m + n) states. This upper bound is reachable if the size of the alphabet can depend on m and n.

Let A be an m-state DFA and B be an n-state DFA accepting the finite languages L(A) and L(B), respectively. Then there exists a DFA of at most mn - 3(m+n) + 12 states which accepts the language $L(A) \cap L(B)$. This upper bound is reachable if $|\Sigma|$ depends on m and n.

If $|\Sigma|$ is fixed and *m* and *n* are arbitrarily large, then it was shown that the upper bounds for both union and intersection operation cannot be reached [7].

4.2.4 State Complexity of Reversal

Let $A = (Q_A, \Sigma, \delta_A, q_{0,A}, F_A)$ be a DFA accepting a finite language L, where $|Q| = n \ge 3$ and $|\Sigma| = k \ge 2$. Also, let t be the smallest integer such that $2^{n-1-t} \le k^t$. Then there exists a DFA $B = (Q_B, \Sigma, \delta_B, q_{0,B}, F_B)$, where

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

that accepts the language L^R , i.e., the reversal of L.

In case when $|\Sigma| = 2$, the DFA accepting L^R needs at most $3 \cdot 2^{p-1} - 1$ states if n = 2p or $2^p - 1$ states if n = 2p - 1. This bound is reachable and the proof was

shown for a binary alphabet in [2].

4.2.5 Summary Table

We assume that L_1 is accepted by an *m*-state DFA A_1 and L_2 is accepted by an *n*-state DFA A_2 and m, n > 1. We use *t* to denote the number of final states in A_1 . Table 4.2 shows the summary of the state complexity of some of the basic operations on finite languages.

Operation	$ \Sigma = 1$	$ \Sigma > 1$
$L_1 \cup L_2$	max(m, n), for (m, n) = 1	O(mn)
$L_1 \cap L_2$	min(m, n), for $(m, n) = 1$	O(mn)
$\Sigma^* - L_1$	m	m
L_1L_2	m + n - 1, for $(m, n) = 1$	$O(mn^{t-1} + n^t)$
L_1^R	m	$2^{m/2}$, for $ \Sigma = 2$
L_1^*	$m^2 - 7m + 13$, for $m > 4$	$2^{m-3} + 2^{m-4}$, for $m \ge 4$

Table 4.2: State complexity of basic operations on finite languages

4.3 State Complexity of Combined Operations on Regular Languages

4.3.1 State Complexity of Star of Union

If A_1 is a complete DFA with m_1 states accepting the language L_1 and A_2 is a complete DFA of m_2 states accepting the language L_2 , then the state complexity of star of union of these two regular languages, i.e. $sc((L_1 \cup L_2)^*)$ is $2^{m_1+m_2-1} - 2^{m_1-1} - 2^{m_2-1} + 1$. This upper bound is tight for three-letter alphabet when $m_1, m_2 \ge 3$ and four-letter alphabet when one or both of m_1 and m_2 can be equal to two. It remains an open question whether this upper bound can be reached by regular languages over a twoletter alphabet [27].

4.3.2 State Complexity of Star of Intersection

If L_1 is accepted by a DFA with m_1 states and L_2 is accepted by a DFA with m_2 states, then from [36] we know that the state complexity of $(L_1 \cap L_2)^*$ is at most $2^{m_1m_2-1} + 2^{m_1m_2-2}$. However, in [27] it was shown that the state complexity of star of intersection may become at least reasonably close to the composition of the worst-case state complexities of the individual operations. In general case, i.e. when $m_1, m_2 \geq 3$, $sc((L_1 \cap L_2)^*)$ reaches at least $2^{m_1(m_2-2)}$ over a five-letter alphabet. But in case of eight-letter alphabet, the state complexity becomes $2^{m_1(m_2-2)} + 2^{m_2(m_1-2)} - 2^{m_1m_2-2(m_1+m_2+1)}$. By increasing the alphabet size this lower bound could be further increased. By using large size of alphabet it can be verified that when $m_1, m_2 \leq 3$, the star-of-intersection can reach exactly $2^{m_1m_2-1} + 2^{m_1m_2-2}$.

4.3.3 State Complexity of Reversal of Union

Let L_1 and L_2 be two regular languages recognized by two DFAs of size m and n respectively. Then the reversal of union of these two languages is $(L_1 \cup L_2)^R$. From [36], we know that the state complexity of the union of an m-state DFA language and an n-state DFA language is mn. We also know that the state complexity of the reversal of an n-state DFA language is 2^n . Therefore, from the mathematical composition of the state complexity of union and the state complexity of reversal we get $sc((L_1 \cup L_2)^R)$ is 2^{mn} . However, the state complexity of $(L_1 \cup L_2)^R$ is less than the mathematical composition of the state complexity of individual operations. In [19], it has been shown that the upper bound of the minimal DFA accepting $(L_1 \cup L_2)^R$ is $2^{m+n} - 2^m - 2^n + 2$. This upper bound can be reached and a worst case example was given for a three-letter alphabet in [19].
4.3.4 State Complexity of Reversal of Intersection

Similar to the reversal of union, the mathematical composition of the individual state complexity of reversal of intersection of an *m*-state DFA language and an *n*-state DFA language is 2^{mn} . However, the actual state complexity of the reversal of intersection is the same as that of reversal of union, i.e., $sc((L_1 \cap L_2)^R)$ is $2^{m+n} - 2^m - 2^n + 2$. This can be obtained by the following observation:

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

where \overline{L} denotes the complement of L, and that the state complexity of the complement of an *n*-state DFA language is n [19].

4.3.5 State Complexity of Reversal of Catenation and Reversal of Star

Let L_1 and L_2 be an *m*-state DFA language and an *n*-state DFA language, respectively. Since $(L_1L_2)^R = L_2^R L_1^R$, then the state complexity of reversal of catenation is bounded by $2^{(n+2^m)} - 2^{(2^m-1)}$. However, in [19], it was proved that the actual upper bound is much better which is $3 \cdot 2^{m+n-2} - 2^n + 1$, i.e., $3 \cdot 2^{m+n-2} - 2^n + 1$ states are sufficient for a DFA to accept the language $(L_1L_2)^R$ where $m, n \ge 1$.

Let L be a language accepted by a DFA with *n*-state, then the state complexity of the reversal of the star operation on L, i.e. $sc((L^*)^R)$, is exactly 2^n , for any n > 0[19].

4.3.6 Summary Table

We assume that L_1 is accepted by an *m*-state DFA and L_2 is accepted by an *n*-state DFA and $m, n \ge 3$. Table 4.3 shows the summary of the state complexity of some of

Operation	State Complexity
$(L_1 \cup L_2)^*$	$2^{m+n-1} - 2^{m-1} - 2^{n-1} + 1$
$(L_1 \cap L_2)^*$	$2^{mn-1} + 2^{mn-2}$
$(L_1 \cup L_2)^R$	$2^{m+n} - 2^m - 2^n + 2$
$(L_1 \cap L_2)^R$	$2^{m+n} - 2^m - 2^n + 2$
$(L_1L_2)^R$	$3 \cdot 2^{m+n-2} - 2^n + 1$
$(L_1^*)^R$	2^m

the combined operations on regular languages.

Table 4.3: State complexity of combined operations on regular languages

Chapter 5 State Complexity of Star of Union Operation on Finite Languages

In this chapter, we will study the state complexity of the combined operation, star of union, on finite languages. By star of union combined operation, we mean that two languages are combined using union operation first and then we perform the (Kleene) star operation on the resulting language from the union operation. In case of general regular languages, it has been shown in [27] that if A is a complete DFA with m states accepting the language L(A) and B is a complete DFA of n states accepting the language L(A) and B is a complete DFA of n states accepting the language L(B), then the state complexity of star of union of these two regular languages, i.e., $sc((L(A) \cup L(B))^*)$, is $2^{m+n-1} - 2^{m-1} - 2^{n-1} + 1$. However, in this chapter we will show that in the case where DFA A and B accept finite languages rather than general regular languages, the corresponding bound is exactly 2^{m+n-5} .

Let A be an m-state complete DFA accepting the finite language L(A) and B be an n-state DFA accepting the finite language L(B). In [7] it was shown that the upper bound for union operation on L(A) and L(B) is mn - (m + n). Also, in [2] it was shown that the tight upper bound for star operation on L(B) is $2^{n-3} + 2^{n-4}$. Therefore, the mathematical composition of the state complexity of star of union will be $2^{mn-m-n-3} + 2^{mn-m-n-4}$. But we will show that the actual state complexity of star of union operation on finite languages is not the same as the mathematical composition of the state complexity of individual operations.

5.1 Upper Bound

In the following subsection, we will prove an upper bound and then give a worst-case example that reaches the upper bound in the next subsection.

Theorem 1. Let $A = (Q_A, \Sigma, \delta_A, q_0, F_A)$ be an *m*-state complete DFA accepting the finite language L(A) and $B = (Q_B, \Sigma, \delta_B, p_0, F_B)$ be an *n*-state complete DFA accepting the finite language L(B) such that $|F_A - \{q_0\}| = k_a$ and $|F_B - \{p_0\}| = k_b$. Then there exists a DFA of $2^{m+n-5} - 2^{m-3} - 2^{n-3} + 2^{m-k_a-2} + 2^{n-k_b-2}$ states that accepts the language $(L(A) \cup L(B))^*$.

Proof. Let $A = (Q_A, \Sigma, \delta_A, q_0, F_A)$ be a DFA where $|Q_A| = m$ and $|F_A - \{q_0\}| = k_a$ and $B = (Q_B, \Sigma, \delta_B, p_0, F_B)$ be a DFA where $|Q_B| = n$ and $|F_B - \{p_0\}| = k_b$. Here, we assume that q_{m-1} and p_{n-1} are the only sink states of DFA A and B, respectively, without loss of generality. We will denote $F_A - \{q_0\}$ by F'_A and $F_B - \{p_0\}$ by F'_B in the following. For star-of-union combined operation we first perform the union operation on A and B. Since there are no loops in the initial states of A and B, we can simply merge the initial states q_0 and p_0 and make this merged state (q_0, p_0) also as a final state. Then we add ε -transition from each of the final states of A and B to the merged initial state (q_0, p_0) . That is how we get our NFA $N = (Q_N, \Sigma, \delta_N, (q_0, p_0), F_N)$ where

$$Q_N = (Q_A - \{q_0\}) \cup (Q_B - \{p_0\}) \cup \{(q_0, p_0)\},$$

$$F_N = \{(q_0, p_0)\} \cup (F_A - \{q_0\}) \cup (F_B - \{p_0\}),$$

and the transition function, for each $q \in Q_N$ and for all $u \in \Sigma$, is

$$\delta_{N}(q,u) = \begin{cases} \delta_{A}(q,u) & \text{if } q \in (Q_{A} - \{q_{0}\}) \text{ and } \delta_{A}(q,u) \notin F_{A} \\ \delta_{A}(q,u) \cup \{(q_{0},p_{0})\} & \text{if } q \in (Q_{A} - \{q_{0}\}) \text{ and } \delta_{A}(q,u) \in F_{A} \\ \delta_{B}(q,u) & \text{if } q \in (Q_{B} - \{p_{0}\}) \text{ and } \delta_{B}(q,u) \notin F_{B} \\ \delta_{B}(q,u) \cup \{(q_{0},p_{0})\} & \text{if } q \in (Q_{B} - \{p_{0}\}) \text{ and } \delta_{B}(q,u) \in F_{B} \\ \delta_{A}(q_{0},u) \cup \delta_{B}(p_{0},u) & \text{if } q = \{(q_{0},p_{0})\}. \end{cases}$$

Now by subset construction we convert our NFA N to a DFA and then minimized the DFA to get the DFA $C = (Q_C, \Sigma, \delta_C, \{(q_0, p_0)\}, F_C)$. We calculate the number of states in C, $|Q_C|$, by applying the following rules.

<u>Rule 1</u>: After subset construction in the DFA we will have at most the set of states Q_1 where $Q_1 = \{P_A \cup P_B | P_A \subseteq Q_A, P_B \subseteq Q_B \text{ and } P_A \cup P_B \neq \phi\}$. The cardinality of Q_1 is

$$|Q_1| = 2^{m+n} - 1. (5.1)$$

Now we will find out the states which are not reachable and exclude those states from the above number of states. Also, we will find out the equivalent set of states and merge them.

<u>Rule 2</u>: A set of states in the DFA which contains any of the sink states of A and B or both the sink states of A and B is equivalent to the set of states that contains no sink states but all other states remaining the same because transitions from sink state does not change the state. That is, if Q' is a set of states such that $\phi \neq Q' = P_A \cup P_B$, for $P_A \subseteq Q_A - \{q_{m-1}\}$ and $P_B \subseteq Q_B - \{p_{n-1}\}$, then

$$Q' \equiv Q' \cup \{q_{m-1}\} \equiv Q' \cup \{p_{n-1}\} \equiv Q' \cup \{q_{m-1}, p_{n-1}\}.$$

So, we can exclude $3 \cdot (2^{m+n-2}-1)$ such states. However, we cannot merge the set of states of the form $P'_A \cup \{p_{n-1}\}$ with P'_A and the set of states of the form $P'_B \cup \{q_{m-1}\}$ with P'_B , where $\phi \neq P'_A \subseteq Q_A - F'_A - \{q_0, q_{m-1}\}$ and $\phi \neq P'_B \subseteq Q_B - F'_B - \{p_0, p_{n-1}\}$, since P'_A and P'_B are not reachable as we must have at least one state from both A and B. So, we have to include $(2^{m-k_a-2}-1) + (2^{n-k_b-2}-1)$ such states. Therefore, we get the set of states Q_2 to be excluded and the cardinality of Q_2 is

$$|Q_2| = 3 \cdot (2^{m+n-2} - 1) - (2^{m-k_a-2} - 1) - (2^{n-k_b-2} - 1).$$
(5.2)

<u>Rule 3</u>: Every set of states in the DFA must contain states from both A and B. So, the set of states which contain states only from either A or B is not reachable. We have such set of states P'_A and P'_B where $P'_A = \{P \mid \phi \neq P \subseteq Q_A - \{q_{m-1}\}\}$ and $P'_B = \{R \mid \phi \neq R \subseteq Q_B - \{p_{n-1}\}\}$. Therefore, $|P'_A| = 2^{m-1} - 1$ and $|P'_B| = 2^{n-1} - 1$. Also, we can merge the singleton sink states in the DFA, i.e., $\{q_{m-1}\}$ and $\{p_{n-1}\}$, with $\{q_{m-1}, p_{n-1}\}$. So, we get the set of states Q_3 that should be excluded where

$$|Q_3| = 2^{m-1} + 2^{n-1}. (5.3)$$

<u>Rule 4</u>: We mentioned before that the initial state of A, q_0 , and the initial state of B, p_0 , can be merged because there are no loops in these states as DFA A and Bonly accept finite languages. So, the set of states which contain either q_0 or p_0 but not both does not exist. We can say, every set of states of the form $\{q_0\} \cup P'$ and $\{p_0\} \cup P'$, where

$$\phi \neq P' = P'_A \cup P'_B,$$
$$P'_A \subseteq Q_A - \{q_0, q_{m-1}\},$$

and

$$P'_B \subseteq Q_B - \{p_0, p_{n-1}\},\$$

is not reachable. That is, we should exclude $2 \cdot (2^{m+n-4} - 1)$ such states. But, in rule 3, we already excluded the set of states of the form $\{q_0\} \cup S'_A$ and $\{p_0\} \cup S'_B$ where

$$\phi \neq S'_A \subseteq Q_A - \{q_0, q_{m-1}\}$$

and

$$\phi \neq S'_B \subseteq Q_B - \{p_0, p_{n-1}\}.$$

That is, we already excluded $(2^{m-2}-1) + (2^{n-2}-1)$ states. Finally, we get the set of states Q_4 that should be excluded where

$$|Q_4| = 2 \cdot (2^{m+n-4} - 1) - (2^{m-2} - 1) - (2^{n-2} - 1).$$
(5.4)

<u>Rule 5</u>: Each set of states of the form $\{(q_0, p_0)\} \cup P_A$, where $\phi \neq P_A \subseteq Q_A - \{q_0, q_{m-1}\}$, is not reachable. Similarly, each set of states of the form $\{(q_0, p_0)\} \cup P_B$, where $\phi \neq P_B \subseteq Q_B - \{p_0, p_{n-1}\}$, is not reachable. We can easily verify that transition from any set of states which reaches the set $\{(q_0, p_0)\} \cup P_A$ must also add at least one state from $Q_B - \{p_0\}$. Similarly, transition from any set of states which reaches the set $\{(q_0, p_0)\} \cup P_A$ must also add at least the set $\{(q_0, p_0)\} \cup P_B$ must also add at least one state from $Q_A - \{q_0\}$. So, we will exclude the set of states Q_5 where

$$Q_5 = \{ P \cup R \mid P = \{ (q_0, p_0) \} \cup P_A, R = \{ (q_0, p_0) \} \cup P_B, \\ \phi \neq P_A \subseteq Q_A - \{ q_0, q_{m-1} \} \text{ and } \phi \neq P_B \subseteq Q_B - \{ p_0, p_{n-1} \} \}.$$

Therefore, the cardinality of Q_5 is

$$|Q_5| = (2^{m-2} - 1) + (2^{n-2} - 1).$$
(5.5)

<u>Rule 6</u>: To perform star operation, we added ε -transition from each of the final states of A and B to the initial state (q_0, p_0) in the NFA. Therefore, any set of states in the DFA which contains any of the final states of A and B, but doesn't contain the initial state (q_0, p_0) doesn't exist. So, we have such set of states of the form $(P'_A \cup P'_B) \cup P''$ where

$$P'_{A} \subseteq Q_{A} - F'_{A} - \{q_{0}, q_{m-1}\},$$

$$P'_{B} \subseteq Q_{B} - F'_{B} - \{p_{0}, p_{n-1}\},$$

$$\phi \neq P'' = P''_{A} \cup P''_{B},$$

$$P''_{A} \subseteq F'_{A}, \text{ and}$$

$$P''_{B} \subseteq F'_{B}.$$

That is, we should exclude $(2^{m+n-k_a-k_b-4})(2^{k_a+k_b}-1)$ such states. But, in rule 3, we already excluded the set of states of the form $S'_A \cup S''_A$ and $S'_B \cup S''_B$ where

$$S'_{A} \subseteq Q_{A} - F'_{A} - \{q_{0}, q_{m-1}\},$$

$$S'_{B} \subseteq Q_{B} - F'_{B} - \{p_{0}, p_{n-1}\},$$

$$\phi \neq S''_{A} \subseteq F'_{A}, \text{ and}$$

$$\phi \neq S''_{B} \subseteq F'_{B}.$$

That is, we already excluded $(2^{m-k_a-2})(2^{k_a}-1) + (2^{n-k_b-2})(2^{k_b}-1)$ states. Finally, we should exclude the set of states Q_6 where

$$|Q_6| = (2^{m+n-k_a-k_b-4})(2^{k_a+k_b}-1) - (2^{m-k_a-2})(2^{k_a}-1) - (2^{n-k_b-2})(2^{k_b}-1).$$
(5.6)

<u>Rule 7</u>: Since after star-of-union operation we made the merged initial state (q_0, p_0) also as a final state in the NFA, any set of nonfinal states that contains (q_0, p_0) doesn't exist. Therefore, every set of states of the form $\{(q_0, p_0)\} \cup P_A \cup P_B$,

where

$$\phi \neq P_A \subseteq Q_A - F'_A - \{q_0, q_{m-1}\}$$
and
$$\phi \neq P_B \subseteq Q_B - F'_B - \{p_0, p_{n-1}\},$$

is not reachable. So, we should exclude the set of states Q_7 where

$$Q_7 = \{P \mid P = \{(q_0, p_0)\} \cup P_A \cup P_B, \phi \neq P_A \subseteq Q_A - F'_A - \{q_0, q_{m-1}\} \text{ and} \\ \phi \neq P_B \subseteq Q_B - F'_B - \{p_0, p_{n-1}\}\}.$$

Therefore, the cardinality of Q_7 is

$$|Q_7| = (2^{m-k_a-2} - 1)(2^{n-k_b-2} - 1).$$
(5.7)

<u>Rule 8</u>: Any set of states that contains either q_{m-2} or p_{n-2} or both can be merged into one set because all the transitions from q_{m-2} and p_{n-2} go to the sink state q_{n-1} and p_{n-1} , respectively. That is, if P' is the set of states such that $\phi \neq P' = P'_A \cup P'_B$, for $P'_A \subseteq Q_A - \{q_0, q_{m-2}, q_{m-1}\}$ and $P'_B \subseteq Q_B - \{p_0, p_{n-2}, p_{n-1}\}$, then

$$\{(q_0, p_0)\} \cup P' \cup \{q_{m-2}\} \equiv \{(q_0, p_0)\} \cup P' \cup \{p_{n-2}\} \equiv \{(q_0, p_0)\} \cup P' \cup \{q_{m-2}, p_{n-2}\}.$$

Therefore, we have to exclude $2 \cdot (2^{m+n-6}-1)$ such states. But, in rule 7, we already excluded the set of states of the form $\{(q_0, p_0)\} \cup P''_A \cup \{q_{m-2}\}$ and $\{(q_0, p_0)\} \cup P''_B \cup \{p_{n-2}\}$ where

$$\phi \neq P''_A \subseteq Q_A - \{q_0, q_{m-2}, q_{m-1}\} \text{ and}$$

$$\phi \neq P''_B \subseteq Q_B - \{p_0, p_{n-2}, p_{n-1}\}.$$

That is, we should include $(2^{m-3}-1) + (2^{n-3}-1)$ states. Thus, we get the set of

states Q_8 to be excluded where

$$|Q_8| = 2 \cdot (2^{m+n-6} - 1) - (2^{m-3} - 1) - (2^{n-3} - 1).$$
(5.8)

<u>Rule 9</u>: The states $\{(q_0, p_0)\}$ and $\{(q_0, p_0), q_{m-2}, p_{n-2}\}$ are equivalent because the next transition from $\{q_{m-2}, p_{n-2}\}$ is always $\{q_{n-1}, p_{n-1}\}$. Let the next transition from $\{(q_0, p_0)\}$ be Q'. Therefore, the next transition from $\{(q_0, p_0), q_{m-2}, p_{n-2}\}$ will be $Q' \cup \{q_{m-1}, p_{n-1}\}$. But, in rule 2, we proved that

$$Q' \equiv Q' \cup \{q_{m-1}, p_{n-1}\}.$$

So, we can exclude 1 state, i.e.,

$$|Q_9| = 1. (5.9)$$

Κ

Finally, subtracting total number of states in 5.2 to 5.9 from 5.1, we get the upper bound, i.e.,

$$|Q_C| = |Q_1| - (|Q_2| + |Q_3| + \dots + |Q_9|).$$

After calculating we get the following result

$$|Q_C| = 2^{m+n-5} - 2^{m-3} - 2^{n-3} + 2^{m-k_a-2} + 2^{n-k_b-2}.$$
 (5.10)

We can see that as the value of k_a and k_b increases, less number of states in $2^{m-k_a-2} + 2^{n-k_b-2}$ is added to $2^{m+n-5} - 2^{m-3} - 2^{n-3}$. The upper bound 5.10 reaches the worst-case when $k_a = 1$ and $k_b = 1$, and we get

$$|Q_C| = 2^{m+n-5}. (5.11)$$

Theorem 2. Let A be an m-state complete DFA accepting the finite language L(A)

and B be an n-state complete DFA accepting the finite language L(B). Then $(L(A) \cup L(B))^*$ is accepted by a complete DFA of no more than 2^{m+n-5} states.

5.2 Lower Bound: Worst-case Example

In the following subsection, we will show a worst-case example that reaches the upper bound.

Theorem 3. Let A and B be two DFAs of m states and n states accepting the finite languages L(A) and L(B), respectively. Then the DFA accepting the language $(L(A) \cup L(B))^*$ needs at least 2^{m+n-5} states in the worst case, when the size of the alphabet $|\Sigma| = m + n - 3$ and either $m \ge 3, n > 3$ or $m > 3, n \ge 3$.

Proof. Let m and n be positive numbers such that either $m \ge 3, n > 3$ or $m > 3, n \ge 3$ and

$$\Sigma = \{c, d, e\} \cup \{a_k, b_l | 1 \le k \le m - 3 \text{ and } 1 \le l \le n - 3\}.$$

Let $A = (Q_A, \Sigma, \delta_A, q_0, F_A)$ where $Q_A = \{q_0, q_1, ..., q_{m-1}\}, F_A = \{q_{m-2}\}$ and δ_A be defined as follows:

$$\begin{split} \delta_A(q_i,c) &= \begin{cases} q_{i+1} & \text{for } 0 \leq i \leq m-2 \\ q_{m-1} & \text{otherwise} \end{cases} \\ \delta_A(q_i,d) &= q_{m-1} & \text{for } 0 \leq i \leq m-1 \\ \delta_A(q_i,e) &= \begin{cases} q_{i+1} & \text{for } 0 \leq i \leq m-2 \\ q_{m-1} & \text{otherwise} \end{cases} \\ \delta_A(q_i,a_k) &= q_{i+k+1} & \text{for } 1 \leq k \leq m-3 \text{ and } 0 \leq i \leq m-k-2 \\ \delta_A(q_i,b_l) &= \begin{cases} q_{i+1} & \text{for } 1 \leq l \leq n-3 \text{ and } 0 \leq i \leq m-2 \\ q_{m-1} & \text{for } 1 \leq l \leq n-3 \text{ and } i = m-1. \end{cases} \end{split}$$

All other transitions that are not mentioned above go to the sink state q_{m-1} . DFA A is shown in the figure 5.1 where m = 6 and q_5 is the sink state. In the figure we omitted some transitions to the sink state for simplicity.



Figure 5.1: DFA A of 6 states for Theorem 3

Let $B = (Q_B, \Sigma, \delta_B, p_0, F_B)$ where $Q_B = \{p_0, p_1, ..., p_{n-1}\}, F_B = \{p_{n-2}\}$ and δ_B be defined as follows:

$$\begin{split} \delta_B(p_i, c) &= \begin{cases} p_{i+1} & \text{for } 0 \le i \le n-2 \\ p_{n-1} & \text{otherwise} \end{cases} \\ \delta_B(p_i, d) &= \begin{cases} p_{i+1} & \text{for } 0 \le i \le n-2 \\ p_{n-1} & \text{otherwise} \end{cases} \\ \delta_B(p_i, e) &= p_{n-1} & \text{for } 0 \le i \le n-1 \end{cases} \\ \delta_B(p_i, a_k) &= \begin{cases} p_{i+1} & \text{for } 1 \le k \le m-3 \text{ and } 0 \le i \le n-2 \\ p_{n-1} & \text{for } 1 \le k \le m-3 \text{ and } i = n-1 \end{cases} \\ \delta_B(p_i, b_l) &= p_{i+l+1} & \text{for } 1 \le l \le n-3 \text{ and } 0 \le i \le n-l-2. \end{split}$$

All other transitions that are not mentioned above go to the sink state p_{n-1} . DFA *B* is shown in the figure 5.2 where n = 6 and p_5 is the sink state. In the figure we omitted some transitions to the sink state for simplicity.



Figure 5.2: DFA B of 6 states for Theorem 3

Let $C = (Q, \Sigma, \delta, \{(q_0, p_0)\}, F)$ be the DFA constructed from A and B to accept the language $(L(A) \cup L(B))^*$. Here, $\{(q_0, p_0)\}$ is the singleton initial state. We define

$$Q = \mathcal{P} \cup \mathcal{R}$$

where \mathcal{P} is the set of nonfinal states and \mathcal{R} is the set of final states. Here, $\{(q_0, p_0)\} \notin \mathcal{P}$ and $\{(q_0, p_0)\} \in \mathcal{R}$. We define the transition function δ , for all $u \in \Sigma$, as follows:

$$\delta(P_A \cup P_B, u) = \begin{cases} \delta_A(P_A, u) \cup \delta_B(P_B, u) & \text{if } \delta_A(P_A, u) \cap F_A \neq \phi \\ & \text{and } \delta_B(P_B, u) \cap F_B \neq \phi \\ \delta_A(P_A, u) \cup \delta_B(P_B, u) \cup \{(q_0, p_0)\} & \text{otherwise} \end{cases}$$

where $P_A \subseteq Q_A$ and $P_B \subseteq Q_B$ such that $P_A \cup P_B \in Q$.

Nonfinal states: We can calculate nonfinal states as follows: 1) $P_1 = \{P_A \cup P_B \mid \phi \neq P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}\}$ and $\phi \neq P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}\}$. So,

$$|P_1| = (2^{m-3} - 1)(2^{n-3} - 1).$$
(5.12)

2)
$$P_2 = \{P'_A \cup P'_B \mid P'_A = P_A \cup \{p_{n-1}\}, P'_B = \{q_{m-1}\} \cup P_B, \phi \neq P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}, \text{ and } \phi \neq P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}\}.$$
 So,

$$|P_2| = (2^{m-3} - 1) + (2^{n-3} - 1).$$
(5.13)

3) Sink state $P_3 = \{q_{m-1}, p_{n-1}\}$. So,

$$|P_3| = 1. (5.14)$$

Therefore,

$$|\mathcal{P}| = |P_1| + |P_2| + |P_3| = 2^{m+n-6}.$$
(5.15)

<u>Final states</u>: We calculate final states as follows: 1) $R_1 = \{\{(q_0, p_0)\} \cup R \cup \{q_{m-2}, p_{n-2}\} \mid \phi \neq R = P_A \cup P_B,$ $P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}, \text{ and } P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}\}.$ So,

$$|R_1| = 2^{m+n-6} - 1. (5.16)$$

2) Final state $R_2 = \{(q_0, p_0)\}$ which is also initial state. So,

$$|R_2| = 1. (5.17)$$

Therefore,

$$|\mathcal{R}| = |R_1| + |R_2| = 2^{m+n-6}.$$
(5.18)

Finally,

$$|Q| = |\mathcal{P}| + |\mathcal{R}| = 2^{m+n-5}.$$
(5.19)

Now, we will prove that every state in Q mentioned above is reachable from the initial state $\{(q_0, p_0)\}$ and pairwise inequivalent.

Claim 1. Every set of states in the form $P_A \cup \{p_{n-1}\}$ and $\{q_{m-1}\} \cup P_B$, where $\phi \neq P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}$ and $\phi \neq P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}$, is reachable from the initial state $\{(q_0, p_0)\}$.

Proof. i) Each set of states $\{q_i, p_{n-1}\}$, where $1 \le i \le m-3$, is obviously reachable from the initial state $\{(q_0, p_0)\}$. We know that DFA A has transition for letter e as $\delta_A(q_i, e) = q_{i+1}, \ 0 \le i \le m-2$, and DFA B has transition for letter e as $\delta_B(p_j, e) =$ $p_{n-1}, \ 0 \le j \le n-1$. So, after taking *i*-times input of e, the initial state of DFA C reaches the state $\{q_i, p_{n-1}\}$, i.e., $\delta(\{(q_0, p_0)\}, e^i) = \{q_i, p_{n-1}\}$.

Similarly, each set of states $\{q_{m-1}, p_j\}$, where $1 \leq j \leq n-3$, is reachable from the initial state $\{(q_0, p_0)\}$. We know that DFA *B* has transition for letter *d* as $\delta_B(p_j, d) = p_{j+1}, 0 \leq j \leq n-2$, and DFA *A* has transition for letter *d* as $\delta_A(q_i, d) = q_{m-1}, 0 \leq i \leq m-1$. So, after taking *j*-times input of *d*, the initial state of DFA *C* reaches the state $\{q_{m-1}, p_j\}$, i.e., $\delta(\{(q_0, p_0)\}, d^j) = \{q_{m-1}, p_j\}$.

ii) Every set of states in the form $P_A \cup \{p_{n-1}\}$ is also reachable, where $\phi \neq P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}$.

Let $P_A = \{q_{i_1}, q_{i_2}, ..., q_{i_{k+1}}\}, 1 \le i_1 < i_2 < ... < i_{k+1} \le m-3$. We will prove above claim by induction process.

<u>Base case</u>: If $|P_A| = 1$, then $P_A = \{q_t\}, 1 \le t \le m - 3$. But we already proved in (i) that $\{q_t, p_{n-1}\}, 1 \le t \le m - 3$, is reachable from $\{(q_0, p_0)\}$.

Induction: Let $X_A = \{q_{i_1}, q_{i_2}, ..., q_{i_k}\}, 1 \leq i_1 < i_2 < ... < i_k < m-3$ and $X_A \cup \{p_{n-1}\}$ be reachable from the initial state $\{(q_0, p_0)\}$. We have to prove that when $P_A = \{q_{i_1}, q_{i_2}, ..., q_{i_{k+1}}\} = X_A \cup \{q_{i_{k+1}}\}$ above claim is also true, i.e., $X_A \cup \{q_{i_{k+1}}\} \cup \{p_{n-1}\}$ is also reachable from the initial state $\{(q_0, p_0)\}$.

It is easy to see that if $X_A \cup \{p_{n-1}\}$ is reachable from $\{(q_0, p_0)\}$ then $\{(q_0, p_0)\} \cup X_A \cup \{q_{m-2}, p_{n-2}\}$ is also reachable from $\{(q_0, p_0)\}$. We have $\delta_B(p_{n-1}, e) = p_{n-1}$ and $\delta_A(q_i, e) = q_{i+1}$. Therefore, after consuming some e, each state in X_A is shifted forward until it reaches final state q_{m-2} , which will add $\{q_0, p_0\} \cup \{p_{n-2}\}$ to the set. Now, by b_{n-3} -transition q_1 will be added to the current set and element from the end of X_A will be removed. At the same time, the transition function δ_B will add $\{(q_0, p_0)\} \cup \{q_{m-2}, p_{n-2}\}$ to the current set for b_{n-3} -input. After some repetition of b_{n-3} -transition states of X_A will be restored while $\{(q_0, p_0)\} \cup \{q_{m-2}, p_{n-2}\}$ still be added in the current set. Now, using e-transition we can reach $X_A \cup \{q_{i_{k+1}}\} \cup \{p_{n-1}\}$, i.e., $P_A \cup \{p_{n-1}\}$.

iii) Every set of states in the form $\{q_{m-1}\} \cup P_B$ is also reachable, where $\phi \neq P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}$. This can be proved in the same way as above (ii). Only we have to consider *d*-transition instead of *e*-transition and a_{m-3} -transition instead of b_{n-3} -transition.

Claim 2. Each set of states of the form $P_A \cup P_B$, where $\phi \neq P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}$ and $\phi \neq P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}$, is reachable from the initial state $\{(q_0, p_0)\}$.

Proof. We consider the following cases.

Case 1: $|P_A| = 1$ and $|P_B| = 1$, Case 2: $|P_A| \ge 1$ and $|P_B| = 1$, Case 3: $|P_A| = 1$ and $|P_B| \ge 1$, Case 4: $|P_A| > 1$ and $|P_B| > 1$.

<u>Case 1</u>: Every set of states $\{q_i, p_j\}$, for $1 \le i \le m-3$ and $1 \le j \le n-3$, is reachable. If i = j, then from initial state $\{(q_0, p_0)\}$, considering only transition labeled by c, we can reach any set of states $\{q_i, p_j\}$. If i > j, then the set of states $\{q_i, p_j\}$ can be reached from initial state $\{(q_0, p_0)\}$ by transition labeled by only c and a_k ,

 $1 \leq k \leq m-3$. If i < j, then the set of states $\{q_i, p_j\}$ can be reached from initial state $\{(q_0, p_0)\}$ by transition labeled by only c and b_l , $1 \leq l \leq n-3$.

<u>Case 2</u>: In claim 1(ii), we already proved that if $P_A = \{q_{i_1}, q_{i_2}, ..., q_{i_{k+1}}\}, 1 \le i_1 < i_2 < ... < i_{k+1} \le m-3$, then $P_A \cup \{p_{n-1}\}$ is reachable from the initial state $\{(q_0, p_0)\}$. So, if $X_A = \{q_{i_1}, q_{i_2}, ..., q_{i_k}\}, 1 \le i_1 < i_2 < ... < i_k < m-3$, then $X_A \cup \{p_{n-1}\}$ is also reachable. Also, in claim 1(ii), we proved that if $X_A \cup \{p_{n-1}\}$ is reachable from $\{(q_0, p_0)\}$ then $\{(q_0, p_0)\} \cup X_A \cup \{q_{m-2}, p_{n-2}\}$ is also reachable from $\{(q_0, p_0)\}$.

Let $P_B = \{p_j\}$ where $1 \leq j \leq n-3$. Now, if we take *c*-transition then from $\{(q_0, p_0)\} \cup X_A \cup \{q_{m-2}, p_{n-2}\}$ we will reach $P_A \cup \{p_1\}$. If we take b_j -transition instead of *c*, then we will reach $P_A \cup \{p_{j+1}\}, 1 \leq j < n-3$. Thus we can reach any set of state $P_A \cup \{p_j\}, 1 \leq j \leq n-3$.

<u>Case 3</u>: Case 3 is similar to case 2 and thus can be proved in the same way.

<u>Case 4</u>: If m = n, then we can reach any set of state $\{q_i, p_i\}$ from $\{(q_0, p_0)\}$ by only *c*- transition, i.e., $\delta(\{(q_0, p_0)\}, c^i) = \{q_i, p_i\}$. So, we can reach $\{q_{m-4}, p_{n-4}\}$. If m > n, then we can use a_k -transition along with *c*-transition to reach the state $\{q_{m-4}, p_{n-4}\}$. On the other hand, if m < n, then we can use b_l -transition with *c*-transition to reach the state $\{q_{m-4}, p_{n-4}\}$. We can see that

$$\delta(\{q_{m-4}, p_{n-4}\}, b_1) = \{(q_0, p_0), q_{m-3}, q_{m-2}, p_{n-2}\},$$

$$\delta(\{(q_0, p_0), q_{m-3}, q_{m-2}, p_{n-2}\}, c) = \{(q_0, p_0), q_1, p_1, q_{m-2}, p_{n-2}\},$$

$$\delta(\{(q_0, p_0), q_1, p_1, q_{m-2}, p_{n-2}\}, c) = \{q_1, q_2, p_1, p_2, q_{m-1}, p_{n-1}\}.$$

Similarly, we can reach any set of states $P_A \cup P_B$ where $\phi \neq P_A \subseteq \{q_1, q_2, ..., q_{m-3}\}$ and $\phi \neq P_B \subseteq \{p_1, p_2, ..., p_{n-3}\}$.

Claim 3. The sink state $\{q_{m-1}, p_{n-1}\}$ is reachable from the initial state $\{(q_0, p_0)\}$.

Proof. It can be easily verified that we can reach $\{q_{m-1}, p_{n-1}\}$ from $\{(q_0, p_0)\}$ either using d- or e-transitions or any other transitions. We can see

$$\delta(\{(q_0, p_0)\}, d^{n-3}b_1) = \{q_{m-1}, p_{n-1}\}$$

and

$$\delta(\{(q_0, p_0)\}, e^{m-3}a_1) = \{q_{m-1}, p_{n-1}\}.$$

Claim 4. Each set of states of the form $\{(q_0, p_0)\} \cup R \cup \{q_{m-2}, p_{n-2}\}$, where $\phi \neq R = P_A \cup P_B$, $P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}$, and $P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}$, is reachable from the initial state $\{(q_0, p_0)\}$.

Proof. i) Each set of states $\{(q_0, p_0)\} \cup \{q_i\} \cup \{q_{m-2}, p_{n-2}\}, 1 \le i \le m-3$, is reachable from $\{(q_0, p_0)\}$. We have

$$\delta(\{(q_0, p_0)\}, b_{n-3}) = \{(q_0, p_0)\} \cup \{q_1\} \cup \{q_{m-2}, p_{n-2}\}.$$

Again, if we use c-transitions and/or a_k -transitions we can reach any set of states $\{q_t, p_{n-3}\}, 1 \leq t \leq m-4$. Then by next c-transition we will reach $\{(q_0, p_0)\} \cup \{q_{t+1}\} \cup \{q_{m-2}, p_{n-2}\}, 2 \leq t+1 \leq m-3$. Similarly, each set of states $\{(q_0, p_0)\} \cup \{p_j\} \cup \{q_{m-2}, p_{n-2}\}, 1 \leq j \leq n-3$, is reachable from $\{(q_0, p_0)\}$. We have

$$\delta(\{(q_0, p_0)\}, a_{m-3}) = \{(q_0, p_0)\} \cup \{p_1\} \cup \{q_{m-2}, p_{n-2}\}$$

Again, if we use c-transitions and/or b_l -transitions we can reach any set of states $\{q_{m-3}, p_t\}, 1 \leq t \leq n-4$. Then by next c-transition we will reach $\{(q_0, p_0)\} \cup \{p_{t+1}\} \cup \{q_{m-2}, p_{n-2}\}, 2 \leq t+1 \leq n-3$.

ii) Every set of states $\{(q_0, p_0)\} \cup \{q_i, p_j\} \cup \{q_{m-2}, p_{n-2}\}$, where $1 \le i \le m-3$ and

 $1 \leq j \leq n-3$, is reachable.

We proved above that the set of states $\{(q_0, p_0)\} \cup \{q_{m-3}\} \cup \{q_{m-2}, p_{n-2}\}$ is reachable. Using *c*-transition we can reach $\{(q_0, p_0)\} \cup \{q_1, p_1\} \cup \{q_{m-2}, p_{n-2}\}$ from $\{(q_0, p_0)\} \cup \{q_{m-3}\} \cup \{q_{m-2}, p_{n-2}\}$. In claim 2 we proved that any set of states $\{q_i, p_j, p_{n-3}\}$ is reachable. Now, we have

$$\delta(\{q_i, p_j, p_{n-3}\}, c) = \{(q_0, p_0)\} \cup \{q_{i+1}, p_{j+1}\} \cup \{q_{m-2}, p_{n-2}\},\$$

 $1 \leq i \leq m-4$ and $1 \leq j \leq n-4$. Therefore, each set of states of the form $\{(q_0, p_0)\} \cup \{q_i, p_j\} \cup \{q_{m-2}, p_{n-2}\}$, where $1 \leq i \leq m-3$ and $1 \leq j \leq n-3$, is reachable.

iii) Each set of state $\{(q_0, p_0)\} \cup (P_A \cup P_B) \cup \{q_{m-2}, p_{n-2}\}$, where $\phi \neq P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}$ and $\phi \neq P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}$, is reachable from the initial state $\{(q_0, p_0)\}$. We consider the following cases.

Case 1: $|P_A| \ge 1$ and $|P_B| = 1$, Case 2: $|P_A| = 1$ and $|P_B| \ge 1$, Case 3: $|P_A| > 1$ and $|P_B| > 1$.

<u>Case 1</u>: Let $P_A = \{q_{i_1}, q_{i_2}, \dots, q_{i_{k+1}}\}, 1 \leq i_1 < i_2 < \dots < i_{k+1} \leq m-3$ and $P_B = \{p_j\}, 1 \leq j \leq n-3.$

<u>Base case</u>: If $|P_A| = 1$, then $P_A = \{q_i\}$ where $1 \le i \le m - 3$. Therefore, the set of states $\{(q_0, p_0)\} \cup \{q_i, p_j\} \cup \{q_{m-2}, p_{n-2}\}$ is reachable from $\{(q_0, p_0)\}$, which we already proved above.

Induction: Let $X_A = \{q_{i_1}, q_{i_2}, ..., q_{i_k}\}, 1 \le i_1 < i_2 < ... < i_k < m - 3$. Also, let the set of states $\{(q_0, p_0)\} \cup X_A \cup \{p_j, q_{m-2}, p_{n-2}\}$ is reachable from $\{(q_0, p_0)\}$. Using only

c-transition from $\{(q_0, p_0)\} \cup X_A \cup \{p_{n-3}, q_{m-2}, p_{n-2}\}$ we can reach

$$\{(q_0, p_0)\} \cup X_A \cup \{q_{i_{k+1}}, p_1, q_{m-2}, p_{n-2}\},\$$

i.e.,

$$\{(q_0, p_0)\} \cup P_A \cup \{p_1, q_{m-2}, p_{n-2}\}.$$

Now, if $\{(q_0, p_0)\} \cup P_A \cup \{p_1, q_{m-2}, p_{n-2}\}$ is reachable from $\{(q_0, p_0)\}$, then $\{(q_0, p_0)\} \cup X_A \cup \{q_{m-3}, p_1, q_{m-2}, p_{n-2}\}$ is also reachable. So, by *c*-transition we can reach the state

$$\{(q_0, p_0)\} \cup X_A \cup \{q_{i_{k+1}}, p_2, q_{m-2}, p_{n-2}\},\$$

i.e.,

$$\{(q_0, p_0)\} \cup P_A \cup \{p_2, q_{m-2}, p_{n-2}\}.$$

If we use b_l -transition instead of *c*-transition, we can reach any set of states of the form $\{(q_0, p_0)\} \cup P_A \cup \{p_j, q_{m-2}, p_{n-2}\}, 3 \le j \le n-3.$

<u>Case 2</u>: Case 2 is symmetric to case 1 and thus can be proved in the same way.

<u>Case 3</u>: We already proved that $\{(q_0, p_0)\} \cup P_A \cup \{p_j, q_{m-2}, p_{n-2}\}$, for $\phi \neq P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}$ and $1 \leq j \leq n-3$, is reachable. Therefore, the set of states $\{(q_0, p_0), q_1, q_{m-3}, p_1, q_{m-2}, p_{n-2}\}$ is also reachable. Now, if we use *c*-transition we get

$$\delta(\{(q_0, p_0), q_1, q_{m-3}, p_1, q_{m-2}, p_{n-2}\}, c) = \{(q_0, p_0), q_1, q_2, p_1, p_2, q_{m-2}, p_{n-2}\}.$$

Similarly, we can reach any set of states $\{(q_0, p_0)\} \cup (P_A \cup P_B) \cup \{q_{m-2}, p_{n-2}\}$, for $\phi \neq P_A \subseteq \{q_1, q_2, ..., q_{m-3}\}$ and $\phi \neq P_B \subseteq \{p_1, p_2, ..., p_{n-3}\}$.

Thus we proved that all the states in DFA C are reachable.

Claim 5. All the states in the DFA C are pairwise inequivalent.

Proof. Let $P = P_1 \cup P_2$ and $R = R_1 \cup R_2$ be distinct states of Q, where $P_1, R_1 \subseteq (Q_A - \{q_0\}) \cup \{(q_0, p_0)\}$ and $P_2, R_2 \subseteq Q_B - \{p_0\}$. Consider the case where $P_1 \neq R_1$. We assume that there exists an element $q \in P_1 - R_1$, without loss of generality.

i) If $q = \{(q_0, p_0)\}$, then R does not contain any of the final states, since any set of states containing any final state must contain $\{(q_0, p_0)\}$ in the set. Since, $\{(q_0, p_0)\}$ is a final state of C, it implies that P is an accepting state of C and R is not an accepting state of C, and consequently P and R are inequivalent.

ii) If $q = q_{m-1}$, then we have $P_1 = \{q_{m-1}\}$ and $P_2 = \phi$ and in that case, on any input P_1 remains the same and never reaches the final state, but R_1 can reach q_{m-2} . So, P and R are inequivalent.

ii) If $q = q_i$, $1 \le i \le m-3$, then we claim that $\delta(P, c^{m-i-2}) \in F$ and $\delta(R, c^{m-i-2}) \notin F$. We can verify that δ_A transition function of DFA A takes the state q_i to q_{m-2} on input c^{m-i-2} which adds $\{(q_0, p_0)\}$ to the current set.

On the other hand, we can see that while the transition function δ_A is taking input c^{m-i-2} , the set R_1 can reach the state q_{m-2} which adds $\{(q_0, p_0)\}$ to the current set before consuming all the input c. So, the remaining input c will shift the state q_{m-2} to q_{n-1} and $\{(q_0, p_0)\}$ to q_j where $q_j < q_i$. Otherwise, the set R_1 can reach the state up to q_{m-3} , which implies that $\delta(P, c^{m-i-2}) \in F$ and $\delta(R, c^{m-i-2}) \notin F$ and thus P and R are inequivalent.

The other case $P_2 \neq R_2$ can be proved in the similar way. Thus, we proved that all the states in Q are pairwise inequivalent.

5.2.1 Lower bound when $m \le 3$ and $n \le 3$

The lower bound in 5.19 is true when $m \ge 3$ and n > 3 or, when m > 3 and $n \ge 3$. But, in the case when $m \le 3$ and $n \le 3$, we have only 1 state in the minimal DFA which is $\{(q_0, p_0)\}$.

5.2.2 Lower bound when m = 2 and n > 3 or m > 3 and n = 2

Let A be an m- state complete DFA accepting the finite language L(A) and B be an n- state complete DFA accepting the finite language L(B). In case when m = 2 and n > 3, then $L(A) = \phi$ and $(L(A) \cup L(B))^* = (L(B))^*$. Therefore, $sc(L(A) \cup L(B))^*$ becomes $sc(L(B))^*$ which is the state complexity of star operation on L(B). Similarly, when m > 3 and n = 3, then $L(B) = \phi$ and $(L(A) \cup L(B))^* = (L(A))^*$. Therefore, $sc(L(A) \cup L(B))^*$ becomes $sc(L(A))^*$ which is the state complexity of star operation on L(A).

Chapter 6 State Complexity of Star of Catenation Operation on Finite Languages

In this chapter, we will study the state complexity of the star of catenation operation on finite languages. By star of catenation combined operation we mean that first we combine two languages by catenation and then we do the (Kleene) star operation on the resulting language from the catenation.

6.1 Upper bound

Let $A = (Q_A, \Sigma, \delta_A, q_0, F_A)$ be a complete DFA accepting the finite language L(A), where $|Q_A| = m$, and $|F_A| = k_a$ and $B = (Q_B, \Sigma, \delta_B, p_0, F_B)$ be a complete DFA accepting the finite language L(B), where $|Q_B| = n$, and $|F_B| = k_b$. Here, we assume that in DFA A, q_{m-1} is the only sink state and in DFA B, p_{n-1} is the only sink state, without loss of generality.

First, we construct an NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ by catenating DFA A and B by adding ε -transition from each of the final states of A, $f_a \in F_A$, to the initial state of B, p_0 , and adding ε -transition from each of the final states of B, $f_b \in F_B$, to the initial state of A, q_0 . Since there is no loop in the state q_0 , we simply make this initial

÷.,

state as a final state. Here, $Q_N = Q_A \cup Q_B$, $F_N = F_B \cup \{q_0\}$, and the transition function δ , for each $q \in Q_N$ and for all $u \in \Sigma$, is

$$\delta_N(q,u) = \begin{cases} \delta_A(q,u) & \text{if } q \in Q_A \text{ and } \delta_A(q,u) \notin F_A \\\\ \delta_A(q,u) \cup \{p_0\} & \text{if } q \in Q_A \text{ and } \delta_A(q,u) \in F_A \\\\ \delta_B(q,u) & \text{if } q \in Q_B \text{ and } \delta_B(q,u) \notin F_B \\\\ \delta_B(q,u) \cup \{q_0\} & \text{if } q \in Q_B \text{ and } \delta_B(q,u) \in F_B. \end{cases}$$

Then we convert NFA N to a DFA by subset construction and minimize it and thus get DFA $C = (Q, \Sigma, \delta, \{q_0\}, F)$ which accepts the language $(L(A)L(B))^*$. We calculate the upper bound by applying the following rules.

<u>Rule 1</u>: After subset construction from NFA to DFA, we have at most Q_1 set of states where $Q_1 = \{P_1 \cup P_2 \mid P_1 \subseteq Q_A, P_2 \subseteq Q_B \text{ and } P_1 \cup P_2 \neq \phi\}$ and the cardinality of Q_1 is

$$|Q_1| = 2^{m+n} - 1. (6.1)$$

Now, we will find out the states which are not reachable and then exclude those states from Q_1 . Also we will find out the equivalent set of states and merge them.

<u>Rule 2</u>: A set of states in the DFA which contains any of the sink states of A and B or both the sink states of A and B is equivalent to the set of states that contains no sink states but all other states remaining the same because transitions from sink state does not change the state. That is, if P' is a set of states such that $\phi \neq P' = P_A \cup P_B$, for $P_A \subseteq Q_A - \{q_{m-1}\}$ and $P_B \subseteq Q_B - \{p_{n-1}\}$, then

$$P' \equiv P' \cup \{q_{m-1}\} \equiv P' \cup \{p_{n-1}\} \equiv P' \cup \{q_{m-1}, p_{n-1}\}.$$

So, we can exclude the set of states Q_2 which consists of all the sets $P' \cup \{q_{m-1}\}$,

 $P' \cup \{p_{n-1}\}$, and $P' \cup \{q_{m-1}, p_{n-1}\}$. Therefore, the cardinality of Q_2 is

$$|Q_2| = 3 \cdot (2^{m+n-2} - 1). \tag{6.2}$$

<u>Rule 3</u>: The states $\{q_{m-1}\}$ and $\{p_{n-1}\}$ can be merged with $\{q_{m-1}, p_{n-1}\}$. So, we can exclude

$$|Q_3| = 2. (6.3)$$

<u>Rule 4</u>: Any set of nonfinal states that contains q_0 doesn't exist, since q_0 is a final state in the NFA, it cannot be in the set of nonfinal states in the DFA. We have such set of states of the form $\{q_0\} \cup P$ where $\phi \neq P = P_A \cup P_B$, $P_A \subseteq Q_A - \{q_0, q_{m-1}\}$ and $P_B \subseteq Q_B - F_B - \{p_{n-1}\}$. So, we get the set of states Q_4 to be excluded where

$$Q_4 = \{\{q_0\} \cup P \mid \phi \neq P = P_A \cup P_B, P_A \subseteq Q_A - \{q_0, q_{m-1}\} \text{ and } P_B \subseteq Q_B - F_B - \{p_{n-1}\}\}.$$

Therefore, we get

$$|Q_4| = 2^{m+n-k_b-3} - 1. (6.4)$$

<u>Rule 5</u>: Any set of states which contains any of the final states of B but doesn't contain q_0 doesn't exist, since there is ε -transition from each of the final states of B to the initial state of A, q_0 . We have such set of states of the form $P_A \cup P_B \cup P'_B$ where

$$P_A \subseteq Q_A - \{q_0, q_{m-1}\},$$

 $P_B \subseteq Q_B - F_B - \{p_{n-1}\},$

and

$$\phi \neq P'_B \subseteq F_B.$$

So, we have to exclude the set of states Q_5 where

$$Q_{5} = \{ P_{A} \cup P_{B} \cup P'_{B} \mid P_{A} \subseteq Q_{A} - \{ q_{0}, q_{m-1} \},\$$
$$P_{B} \subseteq Q_{B} - F_{B} - \{ p_{n-1} \} \text{ and } \phi \neq P'_{B} \subseteq F_{B} \}.$$

Therefore, we have

$$|Q_5| = (2^{m+n-k_b-3})(2^{k_b}-1).$$
(6.5)

<u>Rule 6</u>: The states $\{q_0\}$ and $\{q_0, p_{n-2}\}$ are equivalent because all the transitions from p_{n-2} go to the sink state of B, p_{n-1} . Let the next transition from $\{q_0\}$ be Q'. Therefore, the next transition from $\{q_0, p_{n-2}\}$ will be $Q' \cup \{p_{n-1}\}$. But, in rule 2, we proved that

$$Q' \equiv Q' \cup \{p_{n-1}\}.$$

So, we can exclude 1 state, i.e.,

$$|Q_6| = 1. (6.6)$$

<u>Rule 7</u>: Any set of states that contains any of the final states of A must contain the initial state of B, p_0 , because there is ε -transition from each of the final states of A to the initial state of B in the NFA N. So, each set of states that contains any of the final states of A but does not contain p_0 is not reachable. That is, we have such set of states of the form $P'_A \cup P''_A \cup P'_B$ where

$$P'_A \subseteq Q_A - F_A - \{q_{m-1}\},$$
$$\phi \neq P''_A \subseteq F_A$$

and

$$P'_B \subseteq Q_B - \{p_0, p_{n-1}\}.$$

So, we should exclude $(2^{m+n-k_a-3})(2^{k_a}-1)$ such states, if $q_0 \notin F_A$. However, if

 $q_0 \in F_A$, then we already merged $\{q_0\}$ with $\{q_0, p_{n-2}\}$ in rule 6. Therefore, in that case, we should exclude $(2^{m+n-k_a-3})(2^{k_a}-1)-1$ such states. But, in rule 4, we already excluded the set of states of the form $\{q_0\} \cup R'$ where

$$\begin{cases} R' = R'_A \cup R''_A \cup R'_B & \text{if } q_0 \notin F_A \\ \phi \neq R' = R'_A \cup R''_A \cup R'_B & \text{otherwise,} \end{cases}$$
$$R'_A \subseteq \begin{cases} Q_A - F_A - \{q_0, q_{m-1}\} & \text{if } q_0 \notin F_A \\ Q_A - F_A - \{q_{m-1}\} & \text{otherwise,} \end{cases}$$
$$\begin{cases} \phi \neq R''_A \subseteq F_A & \text{if } q_0 \notin F_A \\ R''_A \subseteq F_A - \{q_0\} & \text{otherwise,} \end{cases}$$

and

$$R'_B \subseteq \begin{cases} Q_B - F_B - \{p_0, p_{n-1}\} & \text{if } p_0 \notin F_B \\ Q_B - F_B - \{p_{n-1}\} & \text{otherwise.} \end{cases}$$

So, we have to include

(i) if $q_0 \notin F_A$ and $p_0 \notin F_B$, then $(2^{m+n-k_a-k_b-4})(2^{k_a}-1)$ states, (ii) if $q_0 \notin F_A$ and $p_0 \in F_B$, then $(2^{m+n-k_a-k_b-3})(2^{k_a}-1)$ states, (iii) if $q_0 \in F_A$ and $p_0 \notin F_B$, then $(2^{m+n-k_a-k_b-3})(2^{k_a-1})-1$ states, (iv) if $q_0 \in F_A$ and $p_0 \in F_B$, then $(2^{m+n-k_a-k_b-2})(2^{k_a-1})-1$ states.

Again, in rule 5, we already excluded the set of states of the form $S'_A \cup S'_A \cup S'_B \cup S''_B$ where

$$S'_{A} \subseteq \begin{cases} Q_{A} - F_{A} - \{q_{0}, q_{m-1}\} & \text{if } q_{0} \notin F_{A} \\ Q_{A} - F_{A} - \{q_{m-1}\} & \text{otherwise,} \end{cases}$$

$$\phi \neq S_A'' \subseteq \begin{cases} F_A & \text{if } q_0 \notin F_A \\ F_A - \{q_0\} & \text{otherwise,} \end{cases}$$
$$S_B' \subseteq \begin{cases} Q_B - F_B - \{p_0, p_{n-1}\} & \text{if } p_0 \notin F_B \\ Q_B - F_B - \{p_{n-1}\} & \text{otherwise,} \end{cases}$$

and

$$\phi \neq S'_B \subseteq \begin{cases} F_B & \text{if } p_0 \notin F_B \\ F_B - \{p_0\} & \text{otherwise.} \end{cases}$$

So, we should include

(i) if $q_0 \notin F_A$ and $p_0 \notin F_B$, then $(2^{m+n-k_a-k_b-4})(2^{k_a}-1)(2^{k_b}-1)$ states, (ii) if $q_0 \notin F_A$ and $p_0 \in F_B$, then $(2^{m+n-k_a-k_b-3})(2^{k_a}-1)(2^{k_b-1}-1)$ states, (iii) if $q_0 \in F_A$ and $p_0 \notin F_B$, then $(2^{m+n-k_a-k_b-3})(2^{k_a-1})(2^{k_b}-1)$ states, (iv) if $q_0 \in F_A$ and $p_0 \in F_B$, then $(2^{m+n-k_a-k_b-2})(2^{k_a-1})(2^{k_b-1}-1)$ states.

Therefore, we actually have to exclude the set of states Q_7 where

$$|Q_{7}| = \begin{cases} 2^{m+n-4} - 2^{m+n-k_{a}-4} & \text{if } q_{0} \notin F_{A} \text{ and } p_{0} \notin F_{B} \\ 2^{m+n-4} - 2^{m+n-k_{a}-4} & \text{if } q_{0} \notin F_{A} \text{ and } p_{0} \in F_{B} \\ 2^{m+n-4} - 2^{m+n-k_{a}-k_{b}-3} & \text{if } q_{0} \in F_{A} \text{ and } p_{0} \notin F_{B} \\ 2^{m+n-3} - 2^{m+n-k_{a}-k_{b}-2} & \text{if } q_{0} \in F_{A} \text{ and } p_{0} \in F_{B} \end{cases}$$

$$(6.7)$$

Finally, subtracting total number of states in 6.2 to 6.7 from 6.1, we get the upper bound, i.e.,

$$|Q| = |Q_1| - (|Q_2| + |Q_3| + \dots + |Q_7|).$$

After calculating we get the following results:

.

(a) if $q_0 \notin F_A$, then

$$|Q| = 2^{m+n-4} + 2^{m+n-k_a-4}, (6.8)$$

(b) if $q_0 \in F_A$ and $p_0 \notin F_B$, then

$$|Q| = 2^{m+n-4} + 2^{m+n-k_a-k_b-3}, (6.9)$$

(c) if $q_0 \in F_A$ and $p_0 \in F_B$, then

$$|Q| = 2^{m+n-k_a-k_b-2}. (6.10)$$

We can easily eliminate the third option since it returns smaller number of states than the other two options. If $k_a = 1$ and $k_b = 1$, then for catenation operation we can simply merge the final state of A with the initial state of B, p_0 , and remove the sink state of A, q_{m-1} , by shifting all the transitions to this sink state to the sink state of B, p_{n-1} . Then for star operation we can merge the final state of B with the initial state of A, q_0 , and remove the sink state of B, p_{n-1} . Therefore, in that case we will have a DFA of at most m + n - 2 states accepting the language $(L(A)L(B))^*$.

However from 6.8 and 6.9, we can see that as the value k_a and k_b increases |Q| decreases. So, to get the upper bound we must have either $k_a = 2$ or $k_a = 1$ and $k_b = 2$. Hence, we get the upper bound

$$|Q| = 2^{m+n-4} + 2^{m+n-6} = 5 \cdot 2^{m+n-6}.$$
(6.11)

Thus we get the following theorem.

Theorem 4. Let A be an m-state complete DFA accepting the finite language L(A)and B be an n-state complete DFA accepting the finite language L(B). Then $5 \cdot 2^{m+n-6}$ states are sufficient for a DFA to accept the language $(L(A)L(B))^*$.

6.2 Lower Bound: Worst-case Example

Theorem 5. Let A and B be two DFAs of m states and n states accepting the finite languages L(A) and L(B), respectively, and $m, n \ge 4$. Then the DFA accepting the language $(L(A)L(B))^*$ needs at least $3 \cdot 2^{m+n-6}$ states in the worst case.

Proof. Let $m \ge 4$ and $n \ge 4$ be positive numbers and $\Sigma = \{a, b, c\}$. Let $A = (Q_A, \Sigma, \delta_A, q_0, F_A)$ where $Q_A = \{q_0, q_1, ..., q_{m-1}\}, F_A = \{q_{m-3}, q_{m-2}\}$, and δ_A be defined as follows:

$$\delta_{A}(q_{i}, a) = \begin{cases} q_{i+1} & \text{for } 0 \leq i \leq m-2 \\ q_{m-1} & \text{otherwise} \end{cases}$$

$$\delta_{A}(q_{i}, b) = \begin{cases} q_{i+1} & \text{for } 0 \leq i \leq m-3 \\ q_{m-2} & \text{for } i = m-4 \\ q_{m-1} & \text{otherwise} \end{cases}$$

$$\delta(q_{i}, c) = \begin{cases} q_{i+1} & \text{for } 0 \leq i \leq m-2 \text{ and } i \neq m-4 \\ q_{m-1} & \text{otherwise.} \end{cases}$$

DFA A is shown in the figure 6.1 where q_{m-1} is the sink state.

Let $B = (Q_B, \Sigma, \delta_B, p_0, F_B)$ where $Q_B = \{p_0, p_1, ..., p_{n-1}\}, F_B = \{p_{n-2}\}$, and δ_B be defined as follows:



Figure 6.1: DFA A of m states for Theorem 5

$$\delta_B(p_i, a) = \begin{cases} p_{i+1} & \text{for } 0 \le i \le n-2\\ p_{n-1} & \text{otherwise} \end{cases}$$
$$\delta_B(p_i, b) = \begin{cases} p_{i+1} & \text{for } 1 \le i \le n-2\\ p_{n-1} & \text{otherwise} \end{cases}$$
$$\delta_B(p_i, c) = \begin{cases} p_{i+1} & \text{for } 0 \le i \le n-2\\ p_{n-1} & \text{otherwise.} \end{cases}$$

DFA B is shown in the figure 6.2 where p_{n-1} is the sink state.



Figure 6.2: DFA B of n states for Theorem 5

Let $C = (Q, \Sigma, \delta, \{q_0\}, F)$ be the DFA constructed from A and B to accept the

language $(L(A)L(B))^*$. We define

$$Q = \mathcal{P} \cup \mathcal{R}$$

where \mathcal{P} is the set of nonfinal states and \mathcal{R} is the set of final states. Here, $\{q_0\} \notin \mathcal{P}$ and $\{q_0\} \in \mathcal{R}$. We define the transition function δ , for all $u \in \Sigma$, as follows:

$$\delta(P_A \cup P_B, u) = \begin{cases} \delta_A(P_A, u) \cup \delta_B(P_B, u) & \text{if } \delta_A(P_A, u) \cap F_A \neq \phi \\ & \text{and } \delta_B(P_B, u) \cap F_B \neq \phi \\ \delta_A(P_A, u) \cup \{p_0\} \cup \delta_B(P_B, u) & \text{if } \delta_A(P_A, u) \cap F_A = \phi \\ & \text{and } \delta_B(P_B, u) \cap F_B \neq \phi \\ & \text{and } \delta_B(P_B, u) \cap F_A \neq \phi \\ & \text{and } \delta_B(P_B, u) \cap F_B = \phi \\ & \{q_0\} \cup \delta_A(P_A, u) \cup \{p_0\} \cup \delta_B(P_B, u) & \text{otherwise} \end{cases}$$

where $P_A \subseteq Q_A$ and $P_B \subseteq Q_B$ such that $P_A \cup P_B \in Q$.

Nonfinal states: We can calculate nonfinal states as follows:

1) $P_1 = \{P_A \cup P_B \mid P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}, P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\} \text{ and } P_A \cup P_B \neq \phi\}$. So,

$$|P_1| = 2^{m+n-7} - 1. (6.12)$$

2)
$$P_2 = \{P_A \cup \{q_{m-3}, p_0\} \cup P_B \mid P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\} \text{ and } P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}\}.$$
 So,

$$|P_2| = 2^{m+n-7}. (6.13)$$

3)
$$P_3 = \{P_A \cup \{q_{m-2}, p_0\} \cup P_B \mid P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\} \text{ and }$$

$$P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}\}.$$
 So,

$$|P_3| = 2^{m+n-7}. (6.14)$$

4) Sink state $P_4 = \{q_{m-1}, p_{n-1}\}$. So,

$$|P_4| = 1. (6.15)$$

Therefore,

$$|\mathcal{P}| = |P_1| + |P_2| + |P_3| + |P_4| = 2^{m+n-6} + 2^{m+n-7}.$$
(6.16)

Final states: We calculate final states as follows: 1) $R_1 = \{\{q_0\} \cup P_A \cup P_B \cup \{p_{n-2}\} \mid P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}\}$ and $P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}\}$. So,

$$|R_1| = 2^{m+n-7}. (6.17)$$

2) $R_2 = \{\{q_0\} \cup P_A \cup \{q_{m-3}, p_0\} \cup P_B \cup \{p_{n-2}\} \mid P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\} \text{ and } P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}\}.$ So,

$$|R_2| = 2^{m+n-7}. (6.18)$$

3)
$$R_3 = \{\{q_0\} \cup P_A \cup \{q_{m-2}, p_0\} \cup P_B \cup \{p_{n-2}\} \mid P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\} \text{ and } P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}\}$$
. So,

$$|R_3| = 2^{m+n-7}. (6.19)$$

Therefore,

$$|\mathcal{R}| = |R_1| + |R_2| + |R_3| = 2^{m+n-6} + 2^{m+n-7}.$$
(6.20)

59

Finally,

$$|Q| = |\mathcal{P}| + |\mathcal{R}| = 2^{m+n-5} + 2^{m+n-6} = 3 \cdot 2^{m+n-6}.$$
(6.21)

Now, we will prove that every state in Q mentioned above is reachable from the initial state $\{q_0\}$ and pairwise inequivalent.

Claim 6. Every set of states in the form $\phi \neq P_A \cup P_B$, where $P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}$ and $P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}$, is reachable from the initial state $\{q_0\}$.

Proof. i) Every set of states $\{q_i\}$, $1 \leq i \leq m-4$, and $\{p_j\}$, $1 \leq j \leq n-3$, is obviously reachable from the initial state $\{q_0\}$. Using only *a*-transition we can reach any of the state $\{q_i\}$, $1 \leq i \leq m-4$, since $\delta_A(q_0, a^i) = q_i$, $1 \leq i \leq m-4$. Next, we have $\delta(\{q_{m-4}\}, b) = \{q_{m-2}, p_0\}$ and then $\delta(\{q_{m-2}, p_0\}, a) = \{p_1\}$. Now, using only *a*-transition we can reach any of the state $\{p_j\}$, $2 \leq j \leq n-3$, from $\{p_1\}$, since $\delta_B(p_1, a^j) = p_j$, $2 \leq j \leq n-3$.

ii) Let the set of states $X_A = \{q_{i_1}, q_{i_2}, ..., q_{i_k}\}, 1 \leq i_1 < i_2 < ... < i_k < m - 4$, be reachable from $\{q_0\}$. We will prove by induction process that $P_A = X_A \cup \{q_{i_{k+1}}\}$ is also reachable from $\{q_0\}$.

<u>Base case</u>: If k = 1 then $|P_A| = 1$ and $P_A = \{q_i\}, 1 \le i \le m - 4$. We already proved that $\{q_i\}$ is reachable from $\{q_0\}$.

Induction: If X_A is reachable from $\{q_0\}$ then using only *a*-transition we can shift each state in X_A until we reach the state q_{m-4} . Next *a*-transition will add q_{m-3} and p_0 to the set, and then next *a*-transition will add q_{m-2} , p_0 , and p_1 , while removing state from front of X_A . After some repetition we will reach the state set $\{q_0\} \cup X_A \cup \{p_{n-2}\}$. Then by next *a*-transition we will reach P_A . iii) Let the set of states $X_B = \{p_{i_1}, p_{i_2}, ..., p_{i_k}\}, 1 \leq i_1 < i_2 < ... < i_k < n-3$, be reachable from $\{q_0\}$. We will prove by induction process that $P_B = X_B \cup \{p_{i_{k+1}}\}$ is also reachable from $\{q_0\}$.

<u>Base case</u>: If k = 1, then $|P_B| = 1$ and $P_B = \{p_j\}, 1 \le j \le n-3$. We already proved that $\{p_j\}$ is reachable from $\{q_0\}$.

Induction: If X_B is reachable from $\{q_0\}$ then using only *a*-transition we can shift each state in X_B until we reach the state p_{n-3} . Next *a*-transition will add p_{n-2} and q_0 to the set while removing state from front of X_B . After some repetition we will reach the state set $\{q_{m-2}, p_0\} \cup X_B$. Then by next *a*-transition we will reach P_B .

iv) Now, we will prove the set of states of the form $P_A \cup P_B$, where $\phi \neq P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}$ and $\phi \neq P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}$, is reachable. There are three cases.

Case 1: $|P_A| \ge 1$ and $|P_B| = 1$ Case 2: $|P_A| = 1$ and $|P_B| \ge 1$ Case 3: $|P_A| \ge 1$ and $|P_B| \ge 1$

<u>Case 1</u>: Let us assume $P_A = \{q_{i_1}, q_{i_2}, ..., q_{i_{k+1}}\}, 1 \le i_1 < i_2 < ... < i_{k+1} \le m-4$, and $P_B = \{p_j\}, 1 \le j \le n-3$.

Let $X_A = \{q_{i_1}, q_{i_2}, ..., q_{i_k}\}, 1 \leq i_1 < i_2 < ... < i_k < m-4$, and $X_A \cup \{p_j\}$ is reachable from $\{q_0\}$. If we use *a*-transitions then each state in X_A as well as the state p_j will be shifted to the next state. If any of the state in X_A reaches q_{m-4} or q_{m-3} , then the next transition will add p_0 to the set. Similarly, if p_j reaches p_{n-3} , then the next transition will add q_0 to the set. Therefore, after some repetition we will have the set of states as $\{q_0\} \cup X_A \cup \{p_{j-1}\}$. In some cases, we may have $\{q_{m-2}, p_0, p_{j-1}\},$ $j-1 \neq 0$, in the same set. We can easily eliminate $\{q_{m-2}, p_0\}$ if we use *b*-transition instead of *a*-transition so that finally only one p_j exists in the set. By next *a*-transition we will reach $X_A \cup \{q_{i_{k+1}}\} \cup \{p_j\}$, i.e., $P_A \cup \{p_j\}$. If we reach the set of states as $\{q_0\} \cup X_A \cup \{q_{m-2}, p_0, p_{j-1}\}$, then by next *b*-transition we will reach $X_A \cup \{q_{i_{k+1}}\} \cup \{p_j\}$, i.e., $P_A \cup \{p_j\}$.

<u>Case 2</u>: Let $P_B = \{p_{j_1}, p_{j_2}, ..., p_{j_{k+1}}\}, 1 \le j_1 < j_2 < ... < j_{k+1} \le n-3$, and $P_A = \{q_i\}, 1 \le i \le m-4$.

Let $X_B = \{p_{j_1}, p_{j_2}, ..., p_{j_k}\}, 1 \leq j_1 < j_2 < ... < j_k < n-3$, and $\{q_i\} \cup X_B$ is reachable from $\{q_0\}$. If we use *a*-transitions then each state in X_B as well as the state q_i will be shifted to the next state. If any of the state in X_B reaches p_{n-2} , then the next transition will add q_0 to the set. Similarly, if q_i reaches q_{m-4} or q_{m-3} , then the next transition will add p_0 to the set. Therefore, after some repetition we will have the set of states as $\{q_{i-1}, q_{m-2}, p_0\} \cup X_B$. By next *a*-transition we will reach $\{q_i\} \cup X_B \cup \{p_{j_{k+1}}\}$, i.e., $\{q_i\} \cup P_B$.

<u>Case 3</u>: When any state $q \in P_A$ reaches either q_{m-3} or q_{m-2} , p_0 is added to the current set. So, from $X'_A \cup \{q_{m-3}, p_0\} \cup X_B$ we can reach $X''_A \cup P_B$, where $X'_A, X''_A \subseteq P_A$ and $X_B \subset P_B$. Similarly, when any state $p \in P_B$ reaches p_{n-2} , q_0 is added to the current set. So, from $X_A \cup X'_B \cup \{q_0, p_{n-2}\}$ we can reach $P_A \cup X''_B$, where $X'_B, X''_B \subseteq P_B$ and $X_A \subset P_A$. Thus we can reach any set of states of the form $P_A \cup P_B$, where $\phi \neq P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}$ and $\phi \neq P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}$.

Claim 7. Every set of states in the form $P_A \cup \{q_{m-3}, p_0\} \cup P_B$ and $P_A \cup \{q_{m-2}, p_0\} \cup P_B$, where $P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}$ and $P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}$, is reachable from the initial state $\{q_0\}$.

Proof. In claim 6, we proved that $P_A \cup P_B$ is reachable from $\{q_0\}$ where

 $P_A = \{q_{i_1}, q_{i_2}, \dots, q_{i_{k+1}}\}, \ 1 \le i_1 < i_2 < \dots < i_{k+1} \le m-4, \text{ and} \\ P_B = \{p_{j_1}, p_{j_2}, \dots, p_{j_{l+1}}\}, \ 1 \le j_1 < j_2 < \dots < j_{l+1} \le n-3.$

Therefore, $\{q_0\} \cup X_A \cup X_B \cup \{p_{n-2}\}$ is also reachable where

 $X_A = \{q_{i_1}, q_{i_2}, \dots, q_{i_k}\}, 1 \le i_1 < i_2 < \dots < i_k < m - 4, \text{ and}$
$$P_B = \{p_{j_1}, p_{j_2}, \dots, p_{j_{l+1}}\}, \ 1 \le j_1 < j_2 < \dots < j_{l+1} \le n-3.$$

Now, if we use a-transition then q_{m-3} and p_0 will be added to the set, i.e.,

$$\delta(\{q_0\} \cup X_A \cup X_B \cup \{p_{n-2}\}, a) = P_A \cup \{q_{m-3}, p_0\} \cup P_B$$

But if we use b-transition then q_{m-2} and p_0 will be added to the set, i.e.,

$$\delta(\{q_0\} \cup X_A \cup X_B \cup \{p_{n-2}\}, b) = P_A \cup \{q_{m-2}, p_0\} \cup P_B.$$

Claim 8. The state $\{q_{m-1}, p_{n-1}\}$ is reachable from $\{q_0\}$.

Proof. We have
$$\delta(q_0, b^{m-2}) = \{q_{m-2}, p_0\}$$
 and $\delta(\{q_{m-2}, p_0\}, b) = \{q_{m-1}, p_{n-1}\}$.

Claim 9. Every set of states in the form $\{q_0\} \cup P_A \cup P_B \cup \{p_{n-2}\}$, where $P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}$ and $P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}$, is reachable from the initial state $\{q_0\}$.

Proof. We proved earlier that $P_A \cup P_B$ is reachable. Therefore, $P'_A \cup P'_B \cup \{p_{n-3}\}$ is also reachable where P'_A is the set of states from P_A making each state shifted 1 step backward, and P'_B is the set of states from P_B making each state shifted 1 step backward. Then using *a*-transition from $P'_A \cup P'_B \cup \{p_{n-3}\}$ we will reach $\{q_0\} \cup P_A \cup P_B \cup \{p_{n-2}\}.$

Claim 10. Every set of states in the form $\{q_0\} \cup P_A \cup \{q_{m-3}, p_0\} \cup P_B \cup \{p_{n-2}\}$ and $\{q_0\} \cup P_A \cup \{q_{m-2}, p_0\} \cup P_B \cup \{p_{n-2}\}$, where $P_A \subseteq Q_A - F_A - \{q_0, q_{m-1}\}$, and $P_B \subseteq Q_B - F_B - \{p_0, p_{n-1}\}$, is reachable from the initial state $\{q_0\}$.

Proof. We can prove this claim using the claim 9. We can reach the set $\{q_0\} \cup P_A \cup \{q_{m-3}, p_0\} \cup P_B \cup \{p_{n-2}\}$ using a-transition from $P'_A \cup \{q_{m-4}\} \cup P'_B \cup \{p_{n-3}\}$ and the set $\{q_0\} \cup P_A \cup \{q_{m-2}, p_0\} \cup P_B \cup \{p_{n-2}\}$ using b-transition from $P'_A \cup \{q_{m-4}\} \cup P'_B \cup \{p_{n-3}\}$, where P'_A and P'_B are defined as in claim 9.

Proof. Let $P = P_1 \cup P_2$ and $R = R_1 \cup R_2$ be distinct states of Q, where $P_1, R_1 \subseteq Q_A$ and $P_2, R_2 \subseteq Q_B$. Consider the case where $P_1 \neq R_1$. We assume that there exists an element $q \in P_1 - R_1$, without loss of generality.

i) If $q = q_0$, then P_2 must contain p_{n-2} and R_2 must not contain p_{n-2} , since any set of states containing the final state of B, p_{n-2} , must contain q_0 in the set. Since, q_0 is a final state of C, it implies that P is an accepting state of C and R is not an accepting state of C, and consequently P and R are inequivalent.

ii) If $q = q_{m-1}$, then we have $P_1 = \{q_{m-1}\}$ and $P_2 = \phi$. Therefore, $R_1 = \phi$ and $R_2 = \{p_{n-1}\}$, since there is only set of states in DFA C which contains q_{m-1} and p_{n-1} , i.e., $\{q_{m-1}, p_{n-1}\}$. That is, in this case P and R are inequivalent.

iii) If $q = q_i$, $1 \le i \le m - 4$, then we claim that $\delta(P, b^{m-i-4}ba^{n-2}) \in F$ and $\delta(R, b^{m-i-4}ba^{n-2}) \notin F$. We can verify that δ_A transition function of DFA A takes the state q_i to q_{m-4} on input b^{m-i-4} . Then next b-transition takes q_{m-4} to q_{m-2} which adds p_0 to the current set. Next a^{n-2} input takes p_0 to the final state of B, adding q_0 to the current set.

On the other hand, we can see that while the transition function δ_A is taking input b^{m-i-4} , two cases may occur:

1) The set R_1 can reach the state q_{m-4} before reading all the input *b*. The next input *b* will take the state q_{m-4} to q_{m-2} and add p_0 to the set. However, there will still one or more than one *b*-input remaining. So, the remaining input *b* will take p_0 to the sink state of *B*, p_{n-1} , and the input a^{n-2} will keep the state p_{n-2} unchanged.

2) Otherwise, the set R_1 can reach the state up to q_{m-5} while reading input b^{m-i-4} . The next input b will take q_{m-5} to q_{m-4} . Now we will have only a^{n-2} input remaining. If we read one a input q_{m-4} will be shifted to q_{m-3} adding p_0 to the set. Still we have a^{n-3} input. It is easy to see that p_0 will reach p_{n-3} consuming a^{n-3} input. That is p_0 will never reach p_{n-2} , the final state of B.

Therefore, in both cases P and R are inequivalent.

iv) If $q = q_{m-3}$ or $q = q_{m-2}$, then we claim that $\delta(P, ab^{n-3}) \in F$ but $\delta(R, ab^{n-3}) \notin F$. It is easy to see that if q_{m-3} or q_{m-2} is in P_1 , then P_2 must contain p_0 . Therefore, R does not contain p_0 . Now on input a, p_0 will reach p_1 and then input b^{n-3} will take p_1 to the final state of B, p_{n-2} , adding q_0 to the current set.

On the other hand, on input a, if R_1 reaches $\{q_{m-3}, p_0\}$ or $\{q_{m-2}, p_0\}$, then input b^{n-3} will take p_0 to the sink state p_{n-1} and stay there. In the mean time, other states in R_2 will simply pass the state p_{n-2} . Again, on input a, if R_1 does not reach $\{q_{m-3}, p_0\}$ or $\{q_{m-2}, p_0\}$, then it will take all or some of b^{n-3} input to reach the state. However, reading the remaining input b, R will never reach p_{n-2} . That is, P and R are inequivalent.

The other case $P_2 \neq R_2$ can be proved in the similar way. Thus, we proved that all the states in Q are pairwise inequivalent.

6.3 Ratio Bound

We have an upper bound $5 \cdot 2^{m+n-6}$ and a lower bound $3 \cdot 2^{m+n-6}$. We can see that the lower bound does not coincide with the upper bound. We didn't get the exact state complexity of star of catenation yet. However, we obtained the approximation of the state complexity for this combined operation. We consider the upper bound, i.e., $5 \cdot 2^{m+n-6}$, as the approximation of the state complexity of star of catenation and the ratio bound for this approximation is

$$\frac{5 \cdot 2^{m+n-6}}{3 \cdot 2^{m+n-6}} = \frac{5}{3} = 1.666.$$

Ň

Chapter 7 State Complexity of Star of Intersection Operation on Finite Languages

In this chapter, we will study the state complexity of the star of intersection operation on finite languages. By star of intersection combined operation we mean that first we combine two languages by intersection operation and then we do the (Kleene) star operation on the resulting language from the intersection.

7.1 Upper Bound

In the following subsection, we will prove an upper bound for star of intersection combined operation.

Lemma 1. Given two minimal DFAs A and B accepting the finite languages L(A)and L(B), respectively, mn - 3m - 3n + 12 states are sufficient for the intersection of L(A) and L(B), where m and n are the numbers of states of A and B, respectively.

Proof. For proof the reader may refer to [7].

Let $A = (Q_A, \Sigma, \delta_A, q_0, F_A)$ be a complete DFA where $|Q_A| = m$ and $|F_A - \{q_0\}| = k_a$ and $B = (Q_B, \Sigma, \delta_B, p_0, F_B)$ be a complete DFA where $|Q_B| = n$ and $|F_B - \{p_0\}| = k_a$

۲,

 k_b . Here, we assume that q_{m-1} and p_{n-1} are the only sink states of DFA A and B, respectively, without loss of generality. We will denote $F_A - \{q_0\}$ by F'_A and $F_B - \{p_0\}$ by F'_B . For star-of-intersection combined operation we follow the following steps.

<u>Step 1</u>: We construct a DFA $M = (Q_M, \Sigma, \delta_M, (q_0, p_0), F_M)$ by taking the Cartesian product of states of A and B, where $Q_M = Q_A \times Q_B$, $F_M = F_A \times F_B$, and the transition function, δ_M , for all $q \in Q_A$ and $p \in Q_B$ and $a \in \Sigma$, is

$$\delta_M((q, p), a) = (\delta_A(q, a), \delta_B(p, a)).$$

So, we have $L(M) = L(A) \cap L(B)$ and $|Q_M| = mn$.

In the following when there is no danger of confusion, we will denote a singleton set $\{(i, j)\}$ by (i, j) where $(i, j) \in Q_A \times Q_B$.

Step 2: Next, we remove all the non-reachable states and merge all the equivalent states in M to get the minimal DFA $D = (Q_D, \Sigma, \delta_D, (q_0, p_0), F_D)$ where $Q_D = \{q \mid q \in Q_M\}$, $F_D = \{f \mid f \in F_M\}$, and $\delta_D = \delta_M$. So, we have $|Q_D| = mn - 3m - 3n + 12$ by Lemma 1.

Step 3: Now, to perform the star operation we add ε -transition from each of the final states of D, $f \in F_D$, to the initial state of D, (q_0, p_0) and make the initial state (q_0, p_0) also as a final state and thus get the NFA $N = (Q_N, \Sigma, \delta_N, (q_0, p_0), F_N)$ where $Q_N = Q_D$,

$$F_N = \begin{cases} F_D & \text{if } (q_0, p_0) \in F_D \\ F_D \cup (q_0, p_0) & \text{otherwise,} \end{cases}$$

and the transition function, δ_N , for all $q \in Q_N$ and $a \in \Sigma$, is

$$\delta_N(q,a) = \begin{cases} \delta_D(q,a) & \text{if } \delta_D(q,a) \notin F_D\\ \delta_D(q,a) \cup (q_0, p_0) & \text{otherwise.} \end{cases}$$

Step 4: Finally, we convert NFA N to a DFA by subset construction and then minimize the DFA to get our minimal DFA $C = (Q, \Sigma, \delta, (q_0, p_0), F)$ which accepts the language $(L(A) \cap L(B))^*$. Here, $Q = \{Q' \mid \phi \neq Q' \subseteq Q_N\}, F = \{X \subseteq Q_N \mid X \cap F_N \neq \phi\}$, and the transition function δ , for $Y \subseteq Q$ and for all $u \in \Sigma$, is

$$\delta(Y, u) = \begin{cases} \{\delta_N(y, u) \mid y \in Q_N\} & \text{if } \delta_N(y, u) \notin F_N \\ \{\delta_N(y, u) \mid y \in Q_N\} \cup (q_0, p_0) & \text{otherwise.} \end{cases}$$

Now, we calculate the upper bound by applying the following rules.

<u>Rule 1</u>: By Lemma 1, we have $|Q_N| = mn - 3m - 3n + 12$ states in step 3 in the NFA N. So, after subset construction from NFA N, we will have at most the set of states Q_1 , where $Q_1 = \{Q' \mid \phi \neq Q' \subseteq Q_N\}$, in the DFA. The cardinality of Q_1 is

$$|Q_1| = 2^{mn-3m-3n+12} - 1. (7.1)$$

Now we will find out the states which are not reachable and exclude those states from the above number of states. Also, we will find out the equivalent set of states and merge them.

<u>Rule 2</u>: A set of states in the DFA which contains the sink state (q_{m-1}, p_{n-1}) is equivalent to the set of states that contains no sink states but all other states remaining the same because transitions from the sink state does not change the state. That is, if Q' is a set of states such that $\phi \neq Q' \subseteq Q_N - \{(q_{m-1}, p_{n-1})\}$ then

$$Q' \equiv Q' \cup \{(q_{m-1}, p_{n-1})\}.$$

So, we can exclude the set of states $Q_2 = \{P \mid P = Q' \cup \{(q_{m-1}, p_{n-1})\}$ and $\phi \neq Q' \subseteq Q_N - \{(q_{m-1}, p_{n-1})\}\}$ and thus

$$|Q_2| = 2^{mn - 3m - 3n + 11} - 1. (7.2)$$

<u>Rule</u> <u>3</u>: Since after star-of-intersection operation we made the initial state (q_0, p_0) also as a final state, then any set of nonfinal states in the DFA C which contains the initial state (q_0, p_0) doesn't exist.

We have $|F'_A| = k_a$ and $|F'_B| = k_b$. So, we have $k_a \times k_b$ final states, excluding (q_0, p_0) if $(q_0, p_0) \in F_M$, in the DFA M. But according to the Lemma 5 of [7], a state (q_i, p_{n-2}) , for $1 \le i \le m-3$, is equivalent to the final state (q_{m-2}, p_{n-2}) if q_i is a final state of A. Similarly, a state (q_{m-2}, p_j) , for $1 \le j \le n-3$, is equivalent to the final state (q_{m-2}, p_{n-2}) if p_j is a final state of B. So, if $q \in F'_A - \{q_{m-2}\}$, then

$$(q, p_{n-2}) \equiv (q_{m-2}, p_{n-2}),$$

and if $p \in F'_B - \{p_{n-2}\}$, then

$$(q_{m-2}, p) \equiv (q_{m-2}, p_{n-2}).$$

Therefore, we already merged $k_a + k_b - 2$ final states in DFA D. So, in DFA D as well as in NFA N, we have $k_a k_b - k_a - k_b + 2$ final states excluding (q_0, p_0) , if $(q_0, p_0) \in F_D$. Thus we have $mn - 3m - 3n - k_a k_b + k_a + k_b + 8$ nonfinal states in NFA N excluding the sink state (q_{n-1}, p_{n-1}) . So, we have to exclude the set of states Q_3 in the DFA where

$$|Q_3| = 2^{mn-3m-3n-k_ak_b+k_a+k_b+8}.$$
(7.3)

<u>Rule 4</u>: Any set of final states in the DFA that doesn't contain the initial state (q_0, p_0) does not exist, since there is an ε -transition from each of the final states of NFA N to the initial state (q_0, p_0) . So, we have to exclude the set of states Q_4 where

$$|Q_4| = (2^{mn-3m-3n-k_ak_b+k_a+k_b+8})(2^{k_ak_b-k_a-k_b+2}-1).$$
(7.4)

<u>Rule 5</u>: The states $\{(q_0, p_0)\}$ and $\{(q_0, p_0), (q_{m-2}, p_{n-2})\}$ are equivalent because the next transition from (q_{m-2}, p_{n-2}) is always (q_{n-1}, p_{n-1}) . Let the next transition from $\{(q_0, p_0)\}$ be Q'. Therefore, the next transition from $\{(q_0, p_0), (q_{m-2}, p_{n-2})\}$ will be $Q' \cup \{(q_{m-1}, p_{n-1})\}$. But, in rule 2, we proved that

$$Q' \equiv Q' \cup \{(q_{m-1}, p_{n-1})\}.$$

So, we can exclude 1 state, i.e.,

$$|Q_5| = 1. (7.5)$$

<u>Rule 6</u>:

Let $i = level(q_i)$ = the length of the longest word from q_0 to q_i in DFA A, for $1 \le i \le m-1$, and

 $j = level(p_j) =$ the length of the longest word from p_0 to p_j in DFA B, for $1 \le j \le n-1$.

In the NFA N, the state (q_{i_1}, p_{j_1}) can only have transition to the state (q_{i_2}, p_{j_2}) if and only if both $i_2 > i_1$ and $j_2 > j_1$. Otherwise, there will not exist any transition from the state (q_{i_1}, p_{j_1}) to the state (q_{i_2}, p_{j_2}) . We will denote such states as mutually exclusive. Now, after subset construction from NFA N to the DFA, there cannot be any set of states where both (q_{i_1}, p_{j_1}) and (q_{i_2}, p_{j_2}) are in the same set in the DFA. That is, if $(q_{i_1}, p_{j_1}), (q_{i_2}, p_{j_2}), ..., (q_{i_k}, p_{j_l})$ are mutually exclusive states in NFA N, then $Q' \notin Q$, where $Q = \{P_1 | P_1 \subseteq Q_N\}$ and $\phi \neq Q' \subseteq \{(q_{i_1}, p_{j_1}), (q_{i_2}, p_{j_2}), ..., (q_{i_k}, p_{j_l})\}\}$ and Q' is not a singleton.

To get the maximum number of states we consider in given DFA A and B, each state except the initial state has level greater than the level of previous state, i.e.,

$$level(q_{i+1}) = level(q_i) + 1$$
$$level(p_{i+1}) = level(p_i) + 1.$$

Since, there is always a transition to the initial state (q_0, p_0) , the final state (q_{m-2}, p_{n-2}) and the sink state (q_{m-1}, p_{n-1}) from all other remaining states in NFA N, to calculate non-reachable set of states in DFA C we have to consider only the set $Q' = Q_N - \{(q_0, p_0), (q_{m-2}, p_{n-2}), (q_{n-1}, p_{n-1})\}$. Therefore, |Q'| = mn - 3m - 3n + 9. Let the reachable set of states in $2^{Q'}$ be \mathcal{R} . So, we have to exclude the non-reachable set of states Q_6 , where

 $Q_6 = \{ P \cup P' \mid P' = (q_0, p_0) \cup P \cup (q_{m-2}, p_{n-2}), \phi \neq P \subseteq Q' \text{ and } P \notin \mathcal{R} \}, \text{ in the DFA.}$

We will define a formula f(m', n') which will calculate the number of reachable set of states from (q_1, p_1) to $(q_{m'}, p_{n'})$. Since we have non-reachable set of states among states (q_1, p_1) to (q_{m-3}, p_{n-3}) , we have $|\mathcal{R}| = f(m-3, n-3)$. Therefore,

$$|Q_6| = 2(2^{mn-3m-3n+9} - 1 - |\mathcal{R}|)$$

= 2{2^{mn-3m-3n+9} - 1 - f(m - 3, n - 3)},

i.e.,

$$|Q_6| = 2^{mn-3m-3n+10} - 2 \cdot f(m-3, n-3) - 2.$$
(7.6)

Finally, subtracting total number of states in 7.2 to 7.6 from 7.1, we get the upper bound, i.e.,

$$|Q| = |Q_1| - (|Q_2| + |Q_3| + \dots + |Q_6|).$$

After calculating we get the following result

$$|Q| = 2 \cdot f(m-3, n-3) + 2. \tag{7.7}$$

<u>Calculating f(m', n')</u>: Here we will use $(q_i, p_j) \rightarrow (q_{i'}, p_{j'})$ notation to denote that the states (q_i, p_j) and $(q_{i'}, p_{j'})$ are not mutually exclusive, i.e., each state in the left side of the arrow has transition to each state in the right side of the arrow. So, in the DFA there can be set of states where both (q_i, p_j) and $(q_{i'}, p_{j'})$ exist in the same set. There are some cases we have to consider:

Case 1. If i = 1 and j = 1 then for (q_i, p_j) , we get

$$(q_1, p_1) \to (q_{i'}, p_{j'})$$
 for $i + 1 \le i' \le m'$ and $j + 1 \le j' \le n'$
 $(q_1, p_1) \to (q_{i'}, p_{j'})$ for $i + 2 \le i' \le m'$ and $j + 1 \le j' \le n' - 1$
...
 $(q_1, p_1) \to (q_{i'}, p_{j'})$ for $i' = m'$ and $j' = j + 1$.

Thus we will get at most

$$1 + \sum_{i'=i+1}^{m'} \left(2^{m'-i'+2} - 2 \right) = 2^{m'} - 2m' - 1$$

reachable set of states.

Case 2. Again if i = 1 and j = 1 then for (q_i, p_j) , we get

$$(q_1, p_1) \to (q_{i'}, p_{j'})$$
 for $i + 1 \le i' \le m' - 1$ and $j + 2 \le j' \le n'$
 $(q_1, p_1) \to (q_{i'}, p_{j'})$ for $i + 2 \le i' \le m' - 2$ and $j + 3 \le j' \le n'$
...
 $(q_1, p_1) \to (q_{i'}, p_{j'})$ for $i' = i + 1$ and $j' = n'$.

Thus we will get at most

$$\sum_{j'=j+2}^{n'} \left(2^{n'-j'+2} - 2 \right) = 2^{n'} - 2n'$$

reachable set of states.

Case 3. For
$$i = 2$$
 to $m' - 1$ and $j = 1$, we get
 $(q_i, p_j) \to (q_{i'}, p_{j'})$ for $i + 1 \le i' \le m' - 1$ and $j + 1 \le j' \le n' - (i - j)$
 $(q_i, p_j) \to (q_{i'}, p_{j'})$ for $i + 2 \le i' \le m' - 2$ and $j + 2 \le j' \le n' - (i - j) - 1$
...
 $(q_i, p_j) \to (q_{i'}, p_{j'})$ for $i' = m'$ and $j' = j + 1$.

Thus we will get at most

$$\sum_{i=2}^{m'-1} \left\{ 1 + \sum_{i'=i+1}^{m'} \left(2^{m'-i'+1} - 1 \right) \right\} = \sum_{i=2}^{m'-1} \left(2^{m'-i+1} - 1 - m' + i \right)$$
$$= 2^{m'} - \left\{ \frac{m'(m'-1)}{2} \right\} - 3$$

reachable set of states.

Case 4. For i = 2 to m' - 1 and j = 1, we get

$$(q_i, p_j) \to (q_{i'}, p_{j'})$$
 for $i + 1 \le i' \le m' - (i - j) + 1$ and $j + 2 \le j' \le n'$
 $(q_i, p_j) \to (q_{i'}, p_{j'})$ for $i + 1 \le i' \le m' - (i - j)$ and $j + 3 \le j' \le n'$
...

$$(q_i, p_j) \rightarrow (q_{i'}, p_{j'})$$
 for $i' = i + 1$ and $j' = n'$.

Thus we will get at most

$$\frac{m'-1}{\sum_{i=2}^{m'-1} \left\{ \sum_{\substack{j'=j+2\\i'=i+1}}^{j'=i} \left(2^{m'-i'+1} - 1 \right) + \sum_{\substack{j'=i+1\\j'=i+1}}^{n'} \left(2^{n'-j'+1} - 1 \right) \right\} \\
= \sum_{i=2}^{m'-1} \left\{ (i-2)(2^{m'-i} - 1) \right\} + \sum_{i=2}^{m'-1} \left(2^{n'-i+1} - 2 - n' + i \right) \\
= 2^{m'-1} - 2m' + 2 - \left\{ \frac{(m'-3)(m'-2)}{2} \right\} + \left[2^{n'} - 2^{n'-m'+2} - m'n' + 2n' + \left\{ \frac{m'(m'-5)}{2} \right\} + 3 \right]$$

K

reachable set of states.

Case 5. For i = m' and j = 1, we get only $(q_{m'}, p_1)$, i.e., 1 state.

$$\begin{aligned} Case \ 6. \ For \ i &= 1 \ \text{and} \ j = 2 \ \text{to} \ n' - 1, \ \text{we get} \\ (q_i, p_j) &\to (q_{i'}, p_{j'}) \ \text{for} \ i + 1 \le i' \le m' - 1 \ \text{and} \ j + 1 \le j' \le n' - (i - j) \\ (q_i, p_j) \to (q_{i'}, p_{j'}) \ \text{for} \ i + 2 \le i' \le m' - 2 \ \text{and} \ j + 2 \le j' \le n' - (i - j) - 1 \\ \dots \\ (q_i, p_j) \to (q_{i'}, p_{j'}) \ \text{for} \ i' = m' \ \text{and} \ j' = j + 1. \\ \text{Thus we will get at most} \\ \sum_{j=2}^{n'-1} \left\{ 1 + \sum_{\substack{i'=i+1\\j'=j+1}}^{i'=j} \left(2^{n'-j'+1} - 1 \right) + \sum_{\substack{i'=j+1\\i'=j+1}}^{m'} \left(2^{m'-i'+1} - 1 \right) \right\} \\ &= \sum_{j=2}^{n'-1} \left\{ 1 + (j - 1)(2^{n'-j} - 1) \right\} + \sum_{j=2}^{n'-1} \left(2^{m'-j+1} - 2 - m' + j \right) \\ &= 2^{n'} - 2n' - \left\{ \frac{(n'-3)(n'-2)}{2} \right\} + \left[2^{m'} - 2^{m'-n'+2} - m'n' + 2m' + \left\{ \frac{n'(n'-5)}{2} \right\} + 3 \right] \end{aligned}$$

reachable set of states.

Case 7. For
$$i = 1$$
 and $j = 2$ to $n' - 2$, we get
 $(q_i, p_j) \to (q_{i'}, p_{j'})$ for $i + 1 \le i' \le m' - (i - j) + 1$ and $j + 2 \le j' \le n'$
 $(q_i, p_j) \to (q_{i'}, p_{j'})$ for $i + 1 \le i' \le m' - (i - j)$ and $j + 3 \le j' \le n'$
...
 $(q_i, p_j) \to (q_{i'}, p_{j'})$ for $i' = i + 1$ and $j' = n'$.

Thus we will get at most

$$\sum_{j=2}^{n'-2} \left\{ \sum_{j'=j+2}^{n'} \left(2^{n'-j+1} - 1 \right) \right\} = \sum_{j=2}^{n'-2} \left(2^{n'-j} - n'+j-1 \right)$$
$$= 2^{n'-1} - \left\{ \frac{n'(n'-1)}{2} \right\} - 1$$

reachable set of states.

Case 8. For i = 1 and j = n', we get only $(q_1, p_{n'})$, i.e., 1 state.

Adding the number of above reachable set of states we get

$$f_1(m',n') = 9 \cdot 2^{m'-1} + 7 \cdot 2^{n'-1} - 4(2^{m'-n'} + 2^{n'-m'}) - 2m'n' - 1 - \left\{\frac{m'(m'+3)}{2}\right\} - \left\{\frac{n'(n'+3)}{2}\right\}.$$
(7.8)

We have considered above only 1 state in left side of the arrow. Now we will consider more than one state in left side of the arrow. We have to consider some cases as follows:

Case 1. For i = 1 and j = 1, and i' = i + 1 to m' - 2 we have $(q_1, p_1), (q_{i'}, p_{j'}) \rightarrow (q_{i''}, p_{j''}), i' + 2 \le i'' \le m' \text{ and } j' + 1 \le j'' \le n' - 1$ $(q_1, p_1), (q_{i'}, p_{j'}) \rightarrow (q_{i''}, p_{j''}), i' + 3 \le i'' \le m' \text{ and } j' + 1 \le j'' \le n' - 1$... $(q_1, p_1), (q_2, p_3) \rightarrow (q_3, p_3), i'' = m' \text{ and } i'' = i' + 1$

$$(q_1, p_1), (q_{i'}, p_{j'}) \rightarrow (q_{i''}, p_{j''}), i'' = m' \text{ and } j'' = j' + 1.$$

Thus we will get at most

0

$$\sum_{i'=2}^{m'-2} \left\{ \sum_{i''=i'+2}^{m'} 2\left(2^{m'-i''+1}-1\right) \right\}$$

reachable set of states. But in the left side of the arrow if we have $c \ge 2$ states then we will get at most

$$\sum_{i'=i+1}^{m'-2} \left[\sum_{\substack{s=i'\\c=2,c=c+1}}^{s=m'-2} \left\{ \sum_{k=s+2}^{m'} \left(2^{c-1} \right) \left(2^{m'-k+1} - 1 \right) \right\} \right]$$

reachable set of states. Let

$$f_{3}(m',n',i',j') = \sum_{\substack{s=i'\\c=2,c=c+1}}^{s=m'-2} \left\{ \sum_{k=s+2}^{m'} \left(2^{c-1} \right) \left(2^{m'-k+1} - 1 \right) \right\}$$
$$= (m'-i'-3)(2^{m'-i'+1}) + 2m'-2i'+6.$$
(7.9)

So, we have at most

$$\sum_{i'=i+1}^{m'-2} \{f_3(m',n',i',j')\}$$

reachable set of states.

Case 2. Again for i = 1 and j = 1, i' = i + 1 to m' - 1 we have $(q_1, p_1), (q_{i'}, p_{j'}) \rightarrow (q_{i''}, p_{j''}), i' + 1 \leq i'' \leq m' - 1$ and $j' + 2 \leq j'' \leq n'$ $(q_1, p_1), (q_{i'}, p_{j'}) \rightarrow (q_{i''}, p_{j''}), i' + 1 \leq i'' \leq m' - 2$ and $j' + 3 \leq j'' \leq n'$... $(q_1, p_1), (q_{i'}, p_{j'}) \rightarrow (q_{i''}, p_{j''}), i'' = i' + 1$ and j'' = n'.

Thus we will get at most

$$\sum_{i'=2}^{m'-1} \left\{ \sum_{j''=j'+2}^{n'} 2\left(2^{n'-j''+1}-1\right) \right\}$$

reachable set of states. But in the left side of the arrow if we have $d \ge 2$ states then we will get at most

$$\sum_{\substack{i'=i+1\\j'=j+1}}^{i'=m'-1} \left[\sum_{\substack{t=j'\\d=2,d=d+1}}^{t=n'-2} \left\{ \sum_{l=t+2}^{n'} \left(2^{d-1} \right) \left(2^{n'-l+1} - 1 \right) \right\} \right]$$

reachable set of states. Let

$$f_4(m',n',i',j') = \sum_{\substack{t=j'\\d=2,d=d+1}}^{t=n'-2} \left\{ \sum_{\substack{l=t+2\\l=t+2}}^{n'} (2^{d-1}) \left(2^{n'-l+1} - 1 \right) \right\}$$
$$= (n'-j'-3)(2^{n'-j'+1}) + 2n'-2j' + 6.$$
(7.10)

So, we have at most

$$\sum_{\substack{i'=i+1\\j'=j+1}}^{i'=m'-1} \{f_4(m',n',i',j')\}$$

reachable set of states.

Case 3. Similar to the above cases, for i = 1 and j = 1, and i' = i + 1 and j' = j + 2to n' - 1, we will get at most

$$\sum_{\substack{j'=j+2\\i'=i+1}}^{j'=j+2} \left\{ f_3(m',n',i',j') \right\} + \sum_{\substack{j'=j+2\\j'=j+2}}^{n'-2} \left\{ f_4(m',n',i',j') \right\}$$

reachable set of states.

Case 4. For i = 2 to m' - 1 and j = 1, and i' = i + 1 to m' - 1 we will get at most

$$\sum_{i=2}^{m'-1} \left[\sum_{i'=i+1}^{m'-2} \left\{ f_3(m',n',i',j') \right\} + \sum_{\substack{i'=i+1\\j'=j+1}}^{i'=m'-1} \left\{ f_4(m',n',i',j') \right\} \right]$$

reachable set of states.

Case 5. For i = 2 to m' - 1 and j = 1, and i' = i + 1 and j' = j + 2 to n' - 1 we will get at most

$$\sum_{i=2}^{m'-1} \left[\sum_{\substack{j'=j+2\\i'=i+1}}^{j'=n'-1} \left\{ f_3(m',n',i',j') \right\} + \sum_{\substack{j'=j+2\\j'=j+2}}^{n'-2} \left\{ f_4(m',n',i',j') \right\} \right]$$

reachable set of states.

Case 6. For i = 1 and j = 2 to n' - 1, and i' = i + 1 to m' - 1 we will get at most

$$\sum_{j=2}^{n'-1} \left[\sum_{i'=i+1}^{m'-2} \left\{ f_3(m',n',i',j') \right\} + \sum_{\substack{i'=i+1\\j'=j+1}}^{i'=m'-1} \left\{ f_4(m',n',i',j') \right\} \right]$$

reachable set of states.

Case 7. For i = 1 and j = 2 to n' - 1, and i' = i + 1 and j' = j + 2 to n' - 1 we will get at most

$$\sum_{j=2}^{n'-1} \left[\sum_{\substack{j'=j+2\\i'=i+1}}^{j'=n'-1} \left\{ f_3(m',n',i',j') \right\} + \sum_{j'=j+2}^{n'-2} \left\{ f_4(m',n',i',j') \right\} \right]$$

reachable set of states.

Finally, adding the above reachable set of states we get $\ensuremath{\Gamma}$

$$f_{2}(m',n') = \sum_{i=1,j=1}^{i=m'-1} \left[\sum_{i'=i+1}^{m'-2} \{f_{3}(m',n',i',j')\} + \sum_{\substack{i'=i+1\\j'=j+1}}^{i'=m'-1} \{f_{4}(m',n',i',j')\} + \sum_{\substack{j'=j+2\\i'=i+1}}^{n'-2} \{f_{4}(m',n',i',j')\} \right] + \sum_{\substack{j'=j+2\\i'=i+1}}^{n'-2} \{f_{4}(m',n',i',j')\} + \sum_{\substack{j'=j+2\\j'=j+1}}^{n'-2} \{f_{4}(m',n',i',j')\} + \sum_{\substack{j'=j+1\\j'=j+1}}^{n'-2} \{f_{4}(m',n',i',j')\} + \sum_{\substack{j'=j+2\\j'=j+2\\i'=i+1}}^{n'-2} \{f_{4}(m',n',i',j')\} - \sum_{\substack{j'=j+2\\i'=i+1}}^{n'-2} \{f_{4}(m',n',i',j')\} - \sum_{\substack{j'=j+2\\j'=j+2}}^{n'-2} \{f_{4}(m',n',i',j')\} - \sum_{\substack{j'=j+2}}^{n'-2} \{f_{4}(m',n',i',j')\} - \sum_{\substack{j'=j+2}^{n'-2} \{f_{4}(m',n',i',j')\} - \sum_{\substack{j'=$$

i.e.,

$$f_{2}(m',n') = \{m'(n')^{2} + m'n' - 5(n')^{2} - 13n' + 4\}2^{m'-2} + \{(m')^{2}n' + m'n' - 5(m')^{2} - 13m' - 2n' + 14\}2^{n'-2} + \frac{1}{3}\{(m')^{3} + (n')^{3} - 6(m')^{2} - 6(n')^{2} + 9(m')^{2}n' + 9m'(n')^{2} - 6m'n' + 29m' + 23n' - 30\}.$$
(7.11)

Therefore,

$$f(m',n') = f_1(m',n') + f_2(m',n')$$

$$= \{m'(n')^2 + m'n' - 5(n')^2 - 13n' + 22\}2^{m'-2}$$

$$+\{(m')^2n' + m'n' - 5(m')^2 - 13m' - 2n' + 28\}2^{n'-2}$$

$$-4(2^{m'-n'} + 2^{n'-m'})$$

$$+\frac{1}{6}\{2(m')^3 + 2(n')^3 + 18(m')^2n' + 18m'(n')^2 - 15(m')^2$$

$$-15(n')^2 - 24m'n' + 49m' + 37n' - 66\}.$$
(7.12)

Now, if we put m' = m - 3 and n' = n - 3 in equation (7.12), we will get

$$f(m-3, n-3) = (mn^2 - 8n^2 - 5mn + 6m + 32n - 2)2^{m-5} + (m^2n - 8m^2 - 5mn + 32m + 4n + 10)2^{n-5} - 4(2^{m-n} + 2^{n-m}) + \frac{1}{6}(2m^3 + 2n^3 + 18m^2n + 18mn^2 - 87m^2 - 87n^2 - 240mn + 751m + 739n - 1890).$$

Finally, putting this value in equation (7.7), we get

$$|Q| = (mn^{2} - 8n^{2} - 5mn + 6m + 32n - 2)2^{m-4} + (m^{2}n - 8m^{2} - 5mn + 32m + 4n + 10)2^{n-4} - 8(2^{m-n} + 2^{n-m}) + \frac{1}{3}(2m^{3} + 2n^{3} + 18m^{2}n + 18mn^{2} - 87m^{2} - 87n^{2} - 240mn + 751m + 739n - 1884).$$
(7.13)

We can see that if $k_a = 1$ and $k_b = 1$, then to perform the star operation, instead of adding ε -transition from the final state of NFA N to the initial state (q_0, p_0) , we can simply merge the final state (q_{m-2}, p_{n-2}) with the initial state (q_0, p_0) and remove the sink state (q_{m-1}, p_{n-1}) . Thus we will get the minimized DFA accepting $(L(A) \cap L(B))^*$, where the DFA will have only mn - 3m - 3n + 10 states. So, we need more than two final states in the NFA N, excluding (q_0, p_0) , i.e., $k_a, k_b \ge 2$, to get the above upper bound. Hence, we get the following theorem.

Theorem 6. Let $A = (Q_A, \Sigma, \delta_A, q_0, F_A)$ be an *m*-state complete DFA accepting the finite language L(A) and $B = (Q_B, \Sigma, \delta_B, p_0, F_B)$ be an *n*-state complete DFA accepting the finite language L(B) such that $|F_A - \{q_0\}| = k_a$ and $|F_B - \{p_0\}| = k_b$ and $k_a, k_b \ge 2$. Then there exists a DFA of $(mn^2 - 8n^2 - 5mn + 6m + 32n - 2)2^{m-4}$ $+ (m^2n - 8m^2 - 5mn + 32m + 4n + 10)2^{n-4} - 8(2^{m-n} + 2^{n-m}) + \frac{1}{3}(2m^3 + 2n^3 + 18m^2n + 18mn^2 - 87m^2 - 87n^2 - 240mn + 751m + 739n - 1884)$ states that accepts the language $(L(A) \cap L(B))^*$.

7.2 Lower Bound: Open Problem

It is still an open problem whether there is a worst-case example that reaches the above upper bound. It is obvious that for a fixed size alphabet this upper bound cannot be reached. We didn't get the lower bound for star of intersection operation. But applying the following example, we found in our experiment that the number of states in the minimal DFA accepting $(L(A) \cap L(B))^*$ is close to the upper bound. So, we propose the following example that might be suggestive and helpful to find out the lower bound.

Example. Let m and n be positive numbers and $|\Sigma| = m + n$ where

$$\Sigma = \{c, d, e, f, g, h\} \cup \{a_k, b_l | 1 \le k \le m - 3 \text{ and } 1 \le l \le n - 3\}.$$

Let $A = (Q_A, \Sigma, \delta_A, q_0, F_A)$ where $Q_A = \{q_0, q_1, ..., q_{m-1}\}, F_A = \{q_{m-3}, q_{m-2}\}$ and δ_A be defined as follows:

$$\delta_{A}(q_{i},c) = \begin{cases} q_{i+1} & \text{for } 0 \leq i \leq m-2 \\ q_{m-1} & \text{otherwise} \end{cases}$$

$$\delta_{A}(q_{i},d) = \begin{cases} q_{i+1} & \text{for } 0 \leq i \leq m-2 \text{ and } i \neq m-4 \\ q_{m-1} & \text{otherwise} \end{cases}$$

$$\delta_{A}(q_{i},e) = \begin{cases} q_{m-2} & \text{if } i = m-4 \\ q_{m-1} & \text{otherwise} \end{cases}$$

$$\delta_{A}(q_{i},f) = \begin{cases} q_{m-2} & \text{if } i = m-3 \\ q_{m-1} & \text{otherwise} \end{cases}$$

$$\delta_{A}(q_{i},f) = \begin{cases} q_{i+1} & \text{for } 0 \leq i \leq m-2 \text{ and } i \neq m-4 \\ q_{m-2} & \text{if } i = m-4 \\ q_{m-1} & \text{otherwise} \end{cases}$$

$$\delta_{A}(q_{i},g) = \begin{cases} q_{i+2} & \text{for } 0 \leq i \leq m-2 \text{ and } i \neq m-4 \\ q_{m-1} & \text{otherwise} \end{cases}$$

$$\delta_{A}(q_{i},h) = \begin{cases} q_{i+2} & \text{for } 0 \leq i \leq m-4 \\ q_{m-2} & \text{if } i = m-3 \\ q_{m-1} & \text{otherwise} \end{cases}$$

$$\delta_{A}(q_{i},a_{k}) = q_{i+k+1} & \text{for } 1 \leq k \leq m-3 \text{ and } 0 \leq i \leq m-2 \\ q_{m-1} & \text{for } 1 \leq l \leq n-3 \text{ and } 0 \leq i \leq m-2 \\ q_{m-1} & \text{for } 1 \leq l \leq n-3 \text{ and } i = m-1. \end{cases}$$

All other transitions that are not mentioned above go to the sink state q_{m-1} . DFA A is shown in the figure 7.1 where m = 6 and q_5 is the sink state. In the figure, we omitted some transitions to the sink state for simplicity.



Figure 7.1: DFA A of 6 states to get the lower bound for star-of-intersection

Let $B = (Q_B, \Sigma, \delta_B, p_0, F_B)$ where $Q_B = \{p_0, p_1, ..., p_{n-1}\}, F_B = \{p_{n-3}, p_{n-2}\}$ and δ_B be defined as follows:

• $\delta_B(p_i, c) = \begin{cases} p_{i+1} & \text{for } 0 \le i \le n-2\\ p_{n-1} & \text{otherwise} \end{cases}$ • $\delta_B(p_i, d) = \begin{cases} p_{i+1} & \text{for } 0 \le i \le n-2 \text{ and } i \ne n-4\\ q_{n-1} & \text{otherwise} \end{cases}$ • $\delta_B(p_i, e) = \begin{cases} p_{n-2} & \text{if } i = n-4\\ p_{n-1} & \text{otherwise} \end{cases}$ • $\delta_B(p_i, f) = \begin{cases} p_{n-2} & \text{if } i = n-3\\ p_{n-1} & \text{otherwise} \end{cases}$ • $\delta_B(p_i, g) = \begin{cases} p_{i+2} & \text{for } 0 \le i \le n-4\\ p_{n-2} & \text{if } i = n-3\\ p_{n-1} & \text{otherwise} \end{cases}$

•
$$\delta_B(p_i, h) = \begin{cases} p_{i+1} & \text{for } 0 \le i \le n-2 \text{ and } i \ne n-4 \\ p_{n-2} & \text{if } i = n-4 \\ p_{n-1} & \text{otherwise} \end{cases}$$

• $\delta_B(p_i, a_k) = \begin{cases} p_{i+1} & \text{for } 1 \le k \le m-3 \text{ and } 0 \le i \le n-2 \\ p_{n-1} & \text{for } 1 \le k \le m-3 \text{ and } i = n-1 \end{cases}$

• $\delta_B(p_i, b_l) = p_{i+l+1}$ for $1 \le l \le n-3$ and $0 \le i \le n-l-2$.

All other transitions that are not mentioned above go to the sink state p_{n-1} . DFA *B* is shown in the figure 7.2 where n = 6 and p_5 is the sink state. In the figure, we omitted some transitions to the sink state for simplicity.



Figure 7.2: DFA B of 6 states to get the lower bound for star-of-intersection

Chapter 8 Conclusion and Future Work

8.1 Conclusion

In this thesis we have studied the state complexity of star of union, star of catenation, and star of intersection combined operations on finite languages. For the first combined operation, we have shown an exact bound and shown a worst-case example which reaches the bound. We have proved an upper bound and a lower bound for the second combined operation and shown an approximation of the state complexity with a good ratio bound. Finally, for star of intersection combined operation we have given an upper bound and suggested a worst-case example to achieve the lower bound. From these results, we have seen that the state complexities of these combined operations on finite languages are different from the mathematical compositions of the state complexities of individual operations on finite languages. Also, these results are very different from the results of the similar combined operations on general regular languages.

8.2 Future Work

There are still many combined operations on finite languages, such as reversal of union, reversal of intersection, star of reversal etc., which have not been studied yet and can be worthy research topics for future study. Also, combined operations on more than two finite languages are worth of study. It will also be interesting to know the lower bound for both star of union and star of intersection combined operations for a fixed size alphabet.

Bibliography

- Jean-Camille Birget, Intersection and union of regular languages and state complexity, Inf. Process. Lett. 43 (1992), 185–190.
- [2] Cezar Càmpeanu, K. Culik, Kai Salomaa, and Sheng Yu, State complexity of basic operations on finite languages, Proceedings of the 4th International Workshp on Automata Implementation, Lecture Notes in Computer Science, vol. 2214, Springer Berlin / Heidelberg, 2001, pp. 60–70.
- [3] Cezar Câmpeanu, Kai Salomaa, and Sheng Yu, Tight lower bound for the state complexity of shuffle of regular languages, J. Autom. Lang. Comb. 7 (2002), 303-310.
- [4] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer, Alternation, J. ACM 28 (1981), 114–133.
- [5] Zoltán Esik, Yuan Gao, Guangwu Liu, and Sheng Yu, Estimation of state complexity of combined operations, Theor. Comput. Sci. 410 (2009), 3272–3280.
- [6] Yuan Gao, Kai Salomaa, and Sheng Yu, The state complexity of two combined operations: Star of catenation and star of reversal, Fundam. Inf. 83 (2008), 75–89.
- [7] Yo-Sub Han and Kai Salomaa, State complexity of union and intersection of finite languages, Proceedings of the 11th international conference on Developments in language theory (Berlin, Heidelberg), DLT'07, Springer-Verlag, 2007, pp. 217– 228.

- [8] Yo-Sub Han, Kai Salomaa, and Derick Wood, Nondeterministic state complexity of basic operations for prefix-free regular languages, Fundam. Inf. 90 (2009), 93– 106.
- [9] Yo-Sub Han, Kai Salomaa, and Sheng Yu, State complexity of combined operations for prefix-free regular languages, Proceedings of the 3rd International Conference on Language and Automata Theory and Applications (Berlin, Heidelberg), LATA '09, Springer-Verlag, 2009, pp. 398–409.
- [10] Markus Holzer and Martin Kutrib, State complexity of basic operations on nondeterministic finite automata, Proceedings of the 7th international conference on Implementation and application of automata (Berlin, Heidelberg), CIAA'02, Springer-Verlag, 2003, pp. 148–157.
- [11] Markus Holzer and Martin Kutrib, Unary language operations and their nondeterministic state complexity, Proceedings of the 6th international conference on Developments in language theory (Berlin, Heidelberg), DLT'02, Springer-Verlag, 2003, pp. 162–172.
- [12] John E. Hopcroft, An n log n algorithm for minimizing states in a finite automaton, Technical Report CS-71-190 (1971).
- [13] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman, Introduction to automata theory, languages, and computation (3rd edition), Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.
- [14] Jozef Jiràsek, Galina Jirásková, and Alexander Szabari, State complexity of concatenation and complementation of regular languages, Implementation and Application of Automata (Michael Domaratzki, Alexander Okhotin, Kai Salomaa, and Sheng Yu, eds.), Lecture Notes in Computer Science, vol. 3317, Springer Berlin / Heidelberg, 2005, pp. 178–189.

- [15] Galina Jirásková, State complexity of some operations on binary regular languages, Theor. Comput. Sci. 330 (2005), 287–298.
- [16] Galina Jirásková, On the state complexity of complements, stars, and reversals of regular languages, Proceedings of the 12th international conference on Developments in Language Theory (Berlin, Heidelberg), DLT '08, Springer-Verlag, 2008, pp. 431–442.
- [17] George Anton Kiraz, Compressed storage of sparse finite-state transducers, Revised Papers from the 4th International Workshop on Automata Implementation, WIA '99, Springer-Verlag, 2001, pp. 109–121.
- [18] Ernst Leiss, Succinct representation of regular languages by boolean automata, Theoretical Computer Science 13 (1981), no. 3, 323 - 330.
- [19] Guangwu Liu, Carlos Martin-Vide, Arto Salomaa, and Sheng Yu, State complexity of basic language operations combined with reversal, Inf. Comput. 206 (2008), 1178–1186.
- [20] A. N. Maslov, Estimates of the number of states of finite automata, Dokl. Akad. Nauk SSSR 194 (1970) 1266-1268 (in Russian) (1970).
- [21] Filippo Mera and Giovanni Pighizzini, Complementing unary nondeterministic automata, Theor. Comput. Sci. 330 (2005), 349–360.
- [22] Cyril Nicaud, Average state complexity of operations on unary automata, Proceedings of the 24th International Symposium on Mathematical Foundations of Computer Science, MFCS '99, Springer-Verlag, 1999, pp. 231–240.
- [23] Giovanni Pighizzini, Unary language concatenation and its state complexity, Revised Papers from the 5th International Conference on Implementation and Application of Automata (London, UK), CIAA '00, Springer-Verlag, 2001, pp. 252– 262.

- [24] M. O. Rabin and D. Scott, Finite automata and their decision problems, IBM J.
 Res. Dev. 3 (1959), 114–125.
- [25] B. Ravikumar and O. H. Ibarra, Relating the type of ambiguity of finite automata to the succinctness of their representation, SIAM J. Comput. 18 (1989), 1263– 1282.
- [26] Arto Salomaa, On the reducibility of events represented in automata, (Annales Academiae scientiarum Fennicae), Suomalainen tiedeakatemia, 1964.
- [27] Arto Salomaa, Kai Salomaa, and Sheng Yu, State complexity of combined operations, Theor. Comput. Sci. 383 (2007), 140–152.
- [28] Arto Salomaa and Ian N. Sneddon, *Theory of automata*, Pergamon Press Reprint, 1969.
- [29] Kai Salomaa and Sheng Yu, On the state complexity of combined operations and their estimation., Int. J. Found. Comput. Sci. (2007), 683–698.
- [30] Jeffrey Shallit, State complexity and Jacobsthal's function, Revised Papers from the 5th International Conference on Implementation and Application of Automata (London, UK), CIAA '00, Springer-Verlag, 2001, pp. 272–278.
- [31] Sheng Yu, Handbook of formal languages, vol. 1. Chapter 2: Regular languages, Springer-Verlag New York, Inc., New York, NY, USA, 1997, pp. 41–110.
- [32] Sheng Yu, State complexity of regular languages, J. Autom. Lang. Comb. 6 (2001), 221–234.
- [33] Sheng Yu, State complexity: Recent results and open problems, Fundam. Inf. 64 (2004), 471–480.

- [34] Sheng Yu and Yuan Gao, State complexity research and approximation, Developments in Language Theory (Giancarlo Mauri and Alberto Leporati, eds.), Lecture Notes in Computer Science, vol. 6795, Springer Berlin / Heidelberg, 2011, 10.1007/978-3-642-22321, pp. 46–57.
- [35] Sheng Yu and Qingyu Zhuang, On the state complexity of intersection of regular languages, SIGACT News 22 (1991), 52–54.
- [36] Sheng Yu, Qingyu Zhuang, and Kai Salomaa, The state complexities of some basic operations on regular languages, Theor. Comput. Sci. 125 (1994), 315–328.