Mathematical Institute
Slovak Academy of Sciences

DESCRIPTIONAL COMPLEXITY OF REGULAR LANGUAGES

Dissertation

Košice, 2010 Alexander Szabari

Abstract

This thesis presents several results on the descriptional complexity of regular languages. We study the deterministic and nondeterministic state complexity of languages that are defined as the number of states in the minimal deterministic or a minimal nondeterministic finite automaton for the given language. By the state complexity of an operation on regular languages we mean the number of states that are sufficient and necessary in the worst case to accept the language resulting from the operation, taken as a function of the complexities of operands.

Our first result shows that the upper bounds on the state complexity of concatenation, that depend on the number of final states in the first automaton, are tight for an arbitrary number of the final states.

The second part of the thesis is devoted to the magic numbers problem. Here we are interested not only in the complexity in the worst case, but we also study all values that can be obtained as the complexity of some operations; the values that cannot be reached in this way are called magic numbers. In particular, we examine magic numbers for the NFA to DFA conversion, union and intersection, and complementation. In all three cases, we show that no magic numbers exist.

Keywords: regular languages, finite automata, concatenation, determinization, union, intersection, complementation, deterministic and nondeterministic state complexity.

Abstrakt

Táto práca sa zaoberá popisnou zložitosťou regulárnych jazykov. Študujeme deterministickú a nedeterministickú stavovú zložitosť jazykov, ktorá je definovaná ako počet stavov v minimálnom deterministickom alebo nedeterministickom konečnostavovom automate pre daný jazyk. Stavovou zložitosťou operácie nad regulárnymi jazykmi rozumieme funkciu závisiacu od zložitostí jednotlivých jazykov, ktorej hodnota je počet stavov potrebných a nevyhnutných v najhoršom prípade na akceptovanie výsledného jazyka.

Náš prvý výsledok ukazuje, že horné odhady stavovej zložitosti zreťazenia, ktoré závisia od počtu koncových stavov prvého automatu, sú dosiahnuteľné pre ľubovoľný počet koncových stavov.

V druhej častiu dizertačnej práce sa zaoberáme problémom magických čísel. Okrem zložitosti v najhoršom prípade skúmame tiež aké hodnoty možno dosiahnuť ako zložitosť nejakej operácie; hodnoty, ktoré nemožno dosiahnuť takýmto spôsobom sa nazývajú magické čísla. Tento problém študujeme v prípade determinizácie, zjednotenia, prieniku a doplnku regulárnych jazykov. Vo všetkých troch prípadoch dokážeme, že magické čísla neexistujú.

Klúčové slová: regulárne jazyky, konečnostavové automaty, zreťazenie, determinizácia, zjednotenie, prienik, doplnok, deterministická a nedeterministická stavová zložitosť .

Acknowledgements

First of all, I would like to thank my adviser Galina Jirásková for granting me the possibility of doing my PhD. study under her supervision. I am also very grateful to her for our cooperation in such an interesting field of computer science as regular languages and their descriptional complexity, and for plenty of time we spent together by solving challenging problems.

Many thanks to Professor Geffert for proposing the problem on the nondeterministic complementation, and to my colleagues for helpful discussions on the topic.

Contents

Chapter 1

Introduction

Finite automata and regular languages are the oldest and the simplest topics in formal language theory. They have been intensively studied for several decades. Nevertheless, some important problems remain open. For example, let us mention the question of how many states are sufficient and necessary for two-way deterministic finite automata to simulate two-way nondeterministic finite automata. The question is related to the well-known open problem whether or not DLOGSPACE equals NLOGSPACE [3, 31, 41].

In recent years, there has been a renewed interest of researchers in automata theory. For a discussion, we refer to [20, 46]. A lot of aspects in this area are now deeply investigated. One of such aspects is descriptional complexity which studies the cost of description of languages by different formal systems.

The state complexity of a regular language is the least number of states in any deterministic finite automaton (DFA) for the language. The nondeterministic state complexity of a regular language is defined as the least number of states in any nondeterministic finite automaton (NFA) accepting the given language. The state complexity (the nondeterministic state complexity) of an operation on regular languages represented by DFAs (NFAs, respectively) is the number of states that are sufficient and necessary in the worst case for a DFA (an NFA, respectively) to accept the language resulting from the operation.

Some early results on state complexity can be found in [33, 34, 36]. The state complexity of some operations such as union, intersection, concatenation and star of languages given by partial DFAs has been investigated Maslov [35]. Similar results for complete DFAs were obtained by Yu, Zhuang, and Salomaa [44]. This first systematic study of the state complexity of regular language operations has been followed by several papers investigating the state complexity of finite language operations and unary language op-

erations [6, 39]. The nondeterministic state complexity of regular language operations has been studied by Holzer and Kutrib in [18]. Domaratzki [9] examined proportional removals, Campeanu et al. [7] investigated the state complexity of shuffle, and Salomaa et al. [42] studied the state complexity of reversals. Further results on this topic can be found in [11, 15].

In [22] Iwama *at al.* stated the question of whether there always exists a minimal NFA of n states whose equivalent minimal DFA has α states for all integers n and α with $n \leqslant \alpha \leqslant 2^n$. The question has also been considered in [23]. In these two papers, it is shown that if $\alpha = 2^n - 2^k$ or $\alpha = 2^n - 2^k - 1$, where $0 \leqslant k \leqslant n/2 - 2$, or if $\alpha = 2^n - k$, where $2 \leqslant k \leqslant 2n - 2$ and some coprimality condition holds, then the corresponding binary n-state NFAs requiring α deterministic states do exist. In [26], appropriate NFAs has been described for all values of n and α, however, the size of the input alphabet for these automata grows exponentially with n. Later, in [12], the size of the input alphabet for the witness automata has been reduced to $n + 2$. The possible holes in the hierarchy are called magic numbers in the literature. It has been recently shown by Geffert [13] that in the case of a unary alphabet, there are a lot of such magic numbers.

1.1 Goals of the Dissertation

The main goals of this dissertation are as follows:

- to investigate the state complexity of the concatenation of two languages represented by deterministic finite automata;
- to compare the nondeterministic state complexity of a regular language and its complement;
- to examine magic numbers for determinization of nondeterministic automata over a fixed alphabet;
- to study magic numbers for union and intersection in the deterministic and nondeterministic case.

1.2 Methods Used in the Dissertation

We use general methods of science as induction, comparison, deduction, and summarising. We usually deal with upper and lower bounds on the complexity of problems. The aim is to show that the two bounds coincide.

To get upper bounds, we use constructive methods: we describe the construction of an appropriate device - deterministic or nondeterministic automaton, the size of which is not greater than the upper bound.

To get a lower bound is usually much more complicated problem. The methods that we use depend on the representation of regular languages.

To prove the minimality of a deterministic automaton, we only need to show the reachability and inequivalence of its states. We prove reachability using the method of mathematical induction, and inequivalence by finding the strings that distinguish the states.

To prove the minimality of a nondeterministic automaton for a regular language, we use a special method called fooling-set lower-bound method: We describe a set of pairs of strings such that the concatenation of the strings in each pair is in the given language, while such a concatenation for two distinct pairs is not. The size of such a fooling set then provides a lower bound on the number of states in any nondeterministic automaton for the given language.

To get the best possible lower bound we have to find a corresponding example, the complexity of which is as high as possible. In the beginning, this leads to some experiments - we construct not too large examples by, in fact, the trial-and-error method, and verify the required properties by using appropriate software.

1.3 Outline of the Dissertation

The present dissertation consists of a five chapters. In Chapter 2 we some basic definitions, notations, and preliminary results. The last section of this chapter describes the fooling-set lower-bound method.

In the third chapter we deal with the state complexity of concatenation of regular languages represented by deterministic automata. The upper bound on the state complexity of concatenation is known to be $m2^n - k2^{n-1}$, where m is the number of states and k is the number of final states in the first automaton, and n is the number of states in the second automaton. We first show that this upper bound is tight for every m, n, k with $0 < k < m$ in the ternary case. Then we give a more complicated prof for a binary alphabet.

Chapter 4 is devoted to the study of magic numbers. Here we are interested not only in the worst-case complexity, but rather in all values that can

be obtained as the deterministic or nondeterministic state complexity of an operation on regular languages. The values that cannot be reached in this way are called magic numbers in the literature [23, 49, 13].

The first section deals with determinization. We prove that in the case of a four-letter alphabet there are no magic numbers, that is, each value from n to 2^n can be obtained as the size of minimal deterministic automaton equivalent to a minimal n-state nondeterministic automaton.

In the next section, we show that there are no magic numbers for union and intersection both in the deterministic and nondeterministic case. In the deterministic case, we show that the entire range of complexities between 1 and mn can be obtained by the union or intersection of an m-state DFA language and an n-state DFA language for any integers m and n such that $m \geqslant 2$ and $n \geqslant 2$. Next, we prove that the nondeterministic state complexity of the union of an m-state NFA language and an n-state NFA language may be arbitrary between 1 and $m+n+1$, except for the case of $m = 1$ and $n = 1$ when the union has nondeterministic state complexity 1 or 3. To prove these results we used a binary alphabet. Finally, we show that the nondeterministic state complexity of the intersection of an m-state NFA language and an n-state NFA language may be arbitrary between 1 and mn. We prove the last result for a ternary alphabet.

The last section studies the magic numbers for complementation of regular languages represented by nondeterministic automata. We show that for all integers n and α with $\log n \leqslant \alpha \leqslant 2^n$, there is a regular language with nondeterministic state complexity n such that the nondeterministic state complexity of its complement is α. We present an easy proof that uses an exponential alphabet, and a more difficult by using an alphabet of size $2n$.

The dissertation ends with some concluding remarks in Chapter 5.

Chapter 2

Preliminaries

In this section, we recall some basic definitions and notations. For further details, we refer to [43, 45].

Let Σ be an alphabet and Σ^* the set of all strings over the alphabet Σ including the empty string ε. The complement of a language L, that is the language $\Sigma^* \setminus L$, is denoted by L^c. The concatenation of two languages K and L is the language $KL = \{uv \mid u \in K \text{ and } v \in L\}$. We denote the cardinality of a finite set A by $|A|$ and its power-set by 2^A.

2.1 Deterministic and Nondeterministic Finite Automata

A *deterministic finite automaton* (DFA) is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is a finite input alphabet, δ is the transition function that maps $Q \times \Sigma$ to Q. q_0 is the initial state, $q_0 \in Q$, and F is the set of accepting states, $F \subseteq Q$. All DFAs are assumed to be complete, i.e., the next state $\delta(q, a)$ is defined for any state q in Q and any symbol a in Σ. The transition function δ can be naturally extended to a function from $Q \times \Sigma^*$ to Q. A string w in Σ^* is accepted by the DFA M if the state $\delta(q_0, w)$ is an accepting state.

A *nondeterministic finite automaton* (NFA) is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$, where Q, Σ, q_0, and F are defined as for a DFA, and $\delta : Q \times \Sigma \to 2^Q$ is the transition function which can be extended to the domain $Q \times \Sigma^*$. A string w in Σ^* is accepted by the NFA M if the set $\delta(q_0, w)$ contains an accepting state.

The *language accepted by* a finite automaton M, denoted $L(M)$, is the set of all strings accepted by the automaton M.

Two automata are said to be *equivalent* if they accept the same language.

A DFA (an NFA) M is called *minimal* if all DFAs (all NFAs, respectively) that are equivalent to M have at least as many states as M. Every nondeterministic finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ can be converted to an equivalent deterministic finite automaton $M' = (2^Q, \Sigma, \delta', q_0', F')$ using an algorithm known as the "subset construction" in the following way. Every state of the DFA M' is a subset of the state set Q. The initial state of the DFA M' is the set $\{q_0\}$. The transition function δ' is defined by $\delta'(R, a) = \bigcup_{r \in R} \delta(r, a)$ for each state R in 2^Q and each symbol a in Σ. A state R in 2^Q is an accepting state of the DFA M' if it contains at least one accepting state of the NFA M. The DFA M' need not be minimal since some states may be unreachable or equivalent.

A language accepted by a DFA (or an NFA) is called *regular*. By a well-known result, each regular language has a unique minimal DFA, up to isomorphism. However, the same result does not hold for minimal NFAs. It is also known [40] that a DFA $M = (Q, \Sigma, \delta, q_0, F)$ is minimal if (i) all its states are reachable from the initial state q_0 and (ii) no two of its states are equivalent; two states p and q are said to be *equivalent* if for all $w \in \Sigma^*$, $\delta(p, w) \in F$ iff $\delta(q, w) \in F$.

2.2 Deterministic and Nondeterministic State Complexity

The *(deterministic) state complexity* of a regular language is the number of states in its minimal DFA. The *nondeterministic state complexity* of a regular language is defined as the number of states in a minimal NFA accepting this language. A regular language with deterministic (nondeterministic) state complexity n is called an n-state DFA language (an n-state NFA language, respectively).

2.3 Fooling-Set Lower-Bound Method

To prove that a NFA is minimal we use a fooling-set lower-bound technique known from communication complexity theory [2, 19]. Although the lower bounds obtained using fooling sets may sometimes be exponentially smaller than the size of minimal NFAs for the corresponding language [21], this technique has been successfully used in the field of regular languages several times [4, 5, 14, 29]. We first define a fooling set. Then we give the lemma from [4] describing a fooling-set lower-bound technique. For the sake of completeness, we present a proof of the lemma here.

Definition 1. *A set of pairs of strings* $\{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$ *is said to be a fooling set for a regular language* L *if for any* i *and* j *in* $\{1, 2, \ldots, n\}$,
(1) the string $x_i y_i$ *is in the language* L, *and*
(2) if $i \neq j$, *then at least one of the strings* $x_i y_j$ *and* $x_j y_i$ *is not in* L.

Lemma 1 (Birget [4])**.** *Let a set of pairs* $\{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$ *be a fooling set for a regular language* L. *Then any NFA for the language* L *needs at least* n *states.* □

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be any NFA accepting the language L. Since $x_i y_i \in L$, there is a state p_i in Q such that $p_i \in \delta(q_0, x_i)$ and $\delta(p_i, y_i) \cap F \neq \varnothing$. Assume that a fixed choice of p_i has been made for any i in $\{1, 2, \ldots, n\}$. We prove that $p_i \neq p_j$ for $i \neq j$. Suppose by contradiction that $p_i = p_j$ for some $i \neq j$. Then the NFA M accepts both strings $x_i y_j$ and $x_j y_i$ which contradicts the assumption that the set $\{(x_i, y_i) \mid 1 \leq i \leq n\}$ is a fooling set for the language L. Hence the NFA M has at least n states. □

Example 1. *Let* $n \geq 1$, *let* $L_n = \{w \in \{a, b\}^* \mid \#_a(w) \equiv 0 \bmod n\}$, *and let*

$$\mathcal{A}_n = \{(a^i, a^{n-i}) \mid i = 1, 2, \ldots, n\}.$$

Note that for every i *and* j *in* $\{1, 2, \ldots, n\}$,
(1) $a^i a^{n-i} \in L_n$, *and*
(2) if $i \neq j$ *then, w.l.o.g.,* $i < j$, *so* $0^i 0^{n-j} \notin L_n$.
Hence the set $\mathcal{A}_n$ *is a fooling set for the language* L_n, *and so any NFA for the language* L_n *needs at least* n *states.* □

Chapter 3

State Complexity of Catenation

The state complexity of concatenation of regular languages represented by deterministic finite automata has been studied by Yu *et al.* [44]. They have shown that $m2^n - k2^{n-1}$ states are sufficient for a DFA to accept the concatenation of an m-state DFA language and an n-state DFA language, where k is the number of the accepting states in the m-state DFA. In the case of $n = 1$, the upper bound m has been shown to be tight, even for a unary alphabet. In the case of $m = 1$ and $n \geq 2$, the worst case $2^n - 2^{n-1}$ has been given by the concatenation of two binary languages. Otherwise, the upper bound $m2^n - 2^{n-1}$ has been shown to be tight for a binary alphabet in [29]. In the case of unary languages, the upper bound on concatenation is mn and it is known to be tight if m and n are relatively prime [44]. The unary case when m and n are not necessarily relatively prime has been studied by Pighizzini and Shallit in [39]. 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.

3.1 Ternary Case

Our first result shows that the upper bounds $m2^n - k2^{n-1}$ are also tight if the first automaton has k accepting states, where $0 < k < m$. We prove it using a ternary alphabet.

Theorem 1. *For any integers m, n, k such that $m \geq 2, n \geq 2$, and $0 < k < m$, there exist a ternary DFA A of m states and k accepting states, and a ternary DFA B of n states such that any DFA accepting the language $L(A)L(B)$ needs at least $m2^n - k2^{n-1}$ states.*

Proof. Let m, n, and k be arbitrary but fixed integers such that $m \geq 2, n \geq 2$, and $0 < k < m$. Let $\Sigma = \{a, b, c\}$.
Define an m-state DFA $A = (Q_A, \Sigma, \delta_A, q_0, F_A)$, where $Q_A = \{q_0, q_1, \ldots, q_{m-1}\}$, $F_A = \{q_{m-k}, q_{m-k+1}, \ldots, q_{m-1}\}$, and for any $i \in \{0, 1, \ldots, m-1\}$,

$$\delta_A(q_i, X) = \begin{cases} q_{i+1}, & \text{if } i < m-k \text{ and } X = a, \\ q_0, & \text{if } i \geq m-k \text{ and } X = a, \\ q_i, & \text{if } X = b, \\ q_{m-1}, & \text{if } i = 0 \text{ and } X = c, \\ q_{i-1}, & \text{if } i > 0 \text{ and } X = c. \end{cases}$$

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

$$\delta_B(i, X) = \begin{cases} 1, & \text{if } i = 0 \text{ and } X = a, \\ i, & \text{if } i > 0 \text{ and } X = a, \\ i+1, & \text{if } i < n-1 \text{ and } X = b, \\ 0, & \text{if } i = n-1 \text{ and } X = b, \\ i, & \text{if } X = c. \end{cases}$$

The DFA A and B are shown in Fig. 3.1 and Fig. 3.2, respectively.

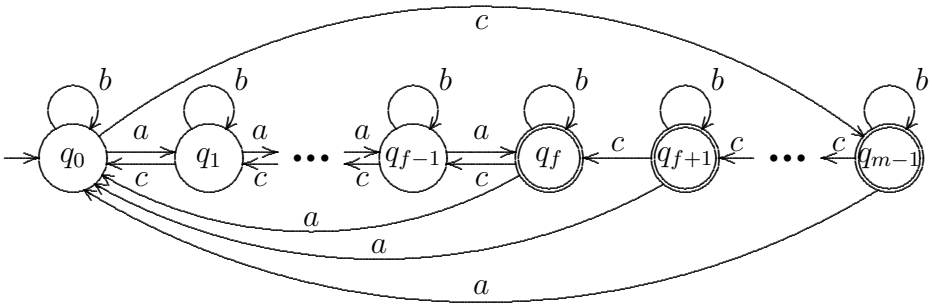

Figure 3.1: The deterministic finite automaton A; $f = m - k$.

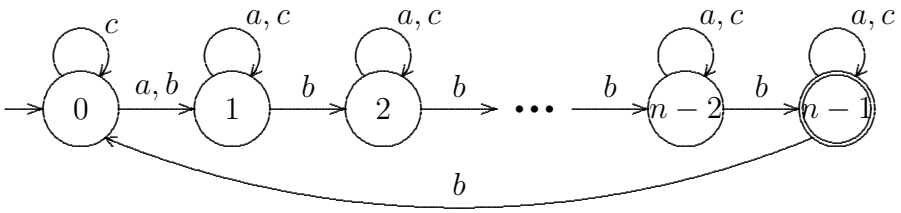

Figure 3.2: The deterministic finite automaton B.

We first describe an NFA accepting the language $L(A)L(B)$, then we construct an equivalent DFA and show that the DFA has at least $m2^n - k2^{n-1}$ states no two of which are equivalent.

Consider the NFA $C = (Q, \Sigma, \delta, q_0, \{n-1\})$, where $Q = Q_A \cup Q_B$, and for $q \in Q$ and $X \in \Sigma$, $\delta(q, X) = \{\delta_A(q, X)\}$ if $q \in Q_A \setminus F_A$, $\delta(q, X) = \{\delta_A(q, X), \delta_B(0, X)\}$ if $q \in F_A$, and $\delta(q, X) = \{\delta_B(q, X)\}$ if $q \in Q_B$, see Fig. 3.3.

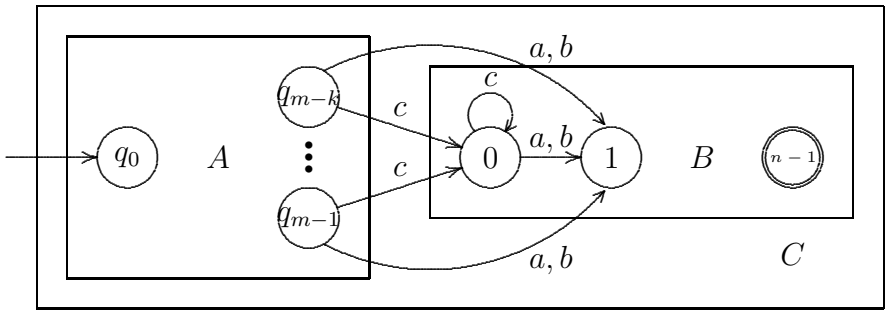

Figure 3.3: The nondeterministic finite automaton C.

The NFA C accepts the language $L(A)L(B)$. Let $C' = (2^Q, \Sigma, \delta', \{q_0\}, F')$ be the DFA obtained from the NFA C by the subset construction. Let $\mathcal{R}$ be the following system of sets: $\mathcal{R} = \{\{q\} \cup S \mid q \in Q_A \setminus F_A \text{ and } S \subseteq Q_B\} \cup \{\{q\} \cup S \mid q \in F_A, S \subseteq Q_B, \text{ and } 0 \in S\}$, i.e., any set in $\mathcal{R}$ consists of exactly one state of Q_A and some states of Q_B, and if a set in $\mathcal{R}$ contains a state of F_A, then it also contains state 0. There are $m2^n - k2^{n-1}$ sets in $\mathcal{R}$. To prove the theorem it is sufficient to show that (I) any set in $\mathcal{R}$ is a reachable state of the DFA C' and (II) no two different states in $\mathcal{R}$ are equivalent.

We prove (I) by induction on the size of sets. The singletons $\{q_0\}, \{q_1\}, \ldots, \{q_{m-k-1}\}$ are reachable since $\{q_i\} = \delta'(\{q_0\}, a^i)$ for $i = 0, 1, \ldots, m-k-1$. Let $1 \leq t \leq n$ and assume that any set in $\mathcal{R}$ of size t is a reachable state of the DFA C'. Using this assumption we prove that any set $\{q_i, j_1, j_2, \ldots, j_t\}$, where $0 \leq j_1 < j_2 < \cdots < j_t < n$ if $0 \leq i < m-k$, and $0 = j_1 < j_2 < \cdots < j_t < n$ if $m-k \leq i < m$, is a reachable state of the DFA C'. There are two cases:

(*i*) $j_1 = 0$. Then we have $\{q_i, 0, j_2, \ldots, j_t\} = \delta'(\{q_0, j_2, \ldots, j_t\}, c^{m-i})$ for $i = 0, 1, \ldots, m-2$, and $\{q_{m-1}, 0, j_2, \ldots, j_t\} = \delta'(\{q_0, j_2, \ldots, j_t\}, c^{m+1})$, where the set $\{q_0, j_2, \ldots, j_t\}$ is reachable by induction.

(*ii*) $j_1 \geq 1$ and $0 \leq i < m-k$. Then we have $\{q_i, j_1, j_2, \ldots, j_t\} = \delta'(\{q_0, 0, j_2 - j_1, \ldots, j_t - j_1\}, b^{j_1}a^i)$, where the latter set is considered in case (*i*).

To prove (II) let $\{q_i\} \cup S$ and $\{q_l\} \cup T$ be two different states in $\mathcal{R}$ with $0 \leq i \leq l \leq m-1$. There are two cases:

(*i*) $i < l$. Then the string $c^i a^{m-k} b^{n-1}$ is accepted by the DFA C' starting in state $\{q_i\} \cup S$ but it is not accepted by C' starting in state $\{q_l\} \cup T$.

(*ii*) $i = l$. Without loss of generality, there is a state j in Q_B such that $j \in S$ and $j \notin T$ (note that $j \geq 1$ if $m-k \leq i \leq m-1$). Then the string b^{n-1-j} is accepted by the DFA C' starting in state $\{q_i\} \cup S$ but it is not accepted by C' starting in state $\{q_l\} \cup T$.

□

3.2 Binary Case

We now strengthen the above result by using a binary alphabet to define witness languages for an arbitrary number of final states in the first automaton.

Theorem 2. *For all integers m, n, k such that $m \geq 2, n \geq 2$, and $0 < k < m$, there exist a binary DFA A of m states and k accepting states, and a binary DFA B of n states such that every DFA accepting the language $L(A)L(B)$ needs at least $m2^n - k2^{n-1}$ states.* □

Proof. Let m, n, and k be arbitrary but fixed integers such that $m \geq 2, n \geq 2$, and $0 < k < m$. Let $\Sigma = \{a, b\}$.

Define an m-state DFA $A = (Q_A, \Sigma, \delta_A, q_0, F_A)$, where $Q_A = \{q_0, \ldots, q_{m-1}\}$, $F_A = \{q_{m-k}, q_{m-k+1}, \ldots, q_{m-1}\}$, and for any $i \in \{0, 1, \ldots, m-1\}$,

$$\delta_A(q_i, X) = \begin{cases} q_{(i+1) \bmod m}, & \text{if } X = a, \\ q_i, & \text{if } X = b. \end{cases}$$

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

$$\delta_B(i, X) = \begin{cases} (i+1) \bmod n, & \text{if } X = a, \\ 0, & \text{if } i = 0 \text{ and } X = b, \\ (i+1) \bmod n, & \text{if } i > 0 \text{ and } X = b. \end{cases}$$

The DFA A and B are shown in Fig. 3.4 and Fig. 3.5, respectively.

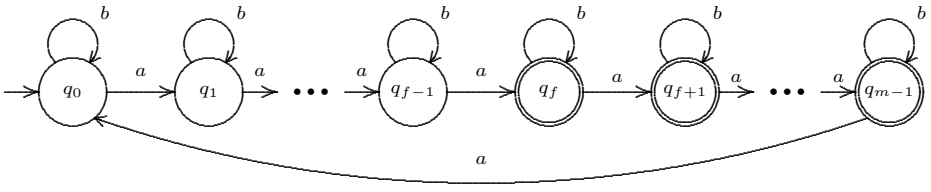

Figure 3.4: The deterministic finite automaton A; $f = m - k$.

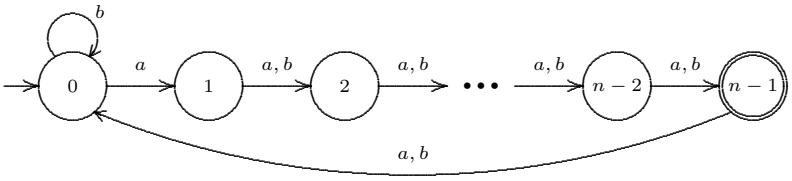

Figure 3.5: The deterministic finite automaton B.

We first describe an NFA accepting the language $L(A)L(B)$, then we construct an equivalent DFA, and show that the DFA has at least $m2^n - k2^{n-1}$ reachable states no two of which are equivalent.

Consider the NFA $C = (Q, \Sigma, \delta, q_0, F)$, where $Q = Q_A \cup Q_B$, $F = \{n-1\}$, and for any $q \in Q$ and any $X \in \Sigma$,

$$\delta(q, X) = \begin{cases} \{\delta_A(q, X)\}, & \text{if } q \in Q_A \setminus F_A, \\ \{\delta_A(q, X), \delta_B(0, X)\}, & \text{if } q \in F_A, \\ \{\delta_B(q, X)\}, & \text{if } q \in Q_B, \end{cases}$$

see Fig. 3.6.

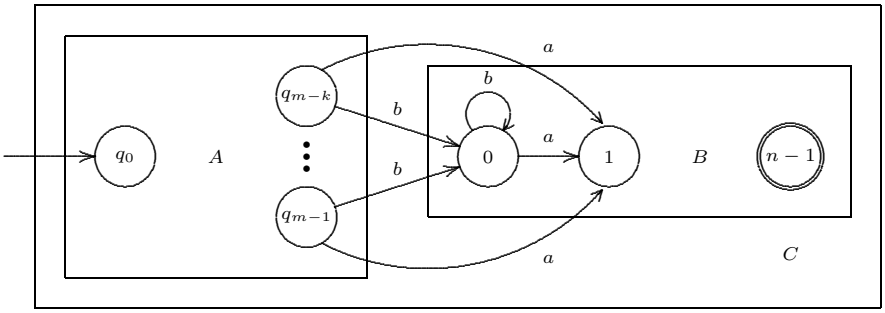

Figure 3.6: The nondeterministic finite automaton C.

Clearly, the NFA C accepts the language $L(A)L(B)$. Let $C' = (2^Q, \Sigma, \delta', \{q_0\}, F')$ be the DFA obtained from the NFA C by the subset construction. Let $\mathcal{R}$ be the following system of sets:

$$\mathcal{R} = \{\{q\} \cup S \mid q \in Q_A \setminus F_A \text{ and } S \subseteq Q_B\} \cup \{\{q\} \cup S \mid q \in F_A, S \subseteq Q_B \text{ and } 0 \in S\},$$

i.e., any set in $\mathcal{R}$ consists of exactly one state of Q_A and some states of Q_B, and if a set in $\mathcal{R}$ contains a state of F_A, then it also contains state 0. There are $m2^n - k2^{n-1}$ sets in $\mathcal{R}$. To prove the theorem it is sufficient to show that (I) any set in $\mathcal{R}$ is a reachable state of the DFA C' and (II) no two different states in $\mathcal{R}$ are equivalent.

We prove (I) by induction on the size of sets. The singletons $\{q_0\}, \{q_1\}, \ldots, \{q_{m-k-1}\}$ are reachable since $\{q_i\} = \delta'(\{q_0\}, a^i)$ for $i = 0, 1, \ldots, m-k-1$. Let $1 \leq s \leq n$ and assume that any set in $\mathcal{R}$ of size s is a reachable state of the DFA C'. Using this assumption we prove that any set $\{q_i, j_1, j_2, \ldots, j_s\}$, where

$0 \leq j_1 < j_2 < \cdots < j_s \leq n-1$ if $0 \leq i \leq m-k-1$, and
$0 = j_1 < j_2 < \cdots < j_s \leq n-1$ if $m-k \leq i \leq m-1$,

is a reachable state of the DFA C'. There are four cases:

(i) $m-k+1 \leq i \leq m-1$ and $j_1 = 0$. We prove this case by induction on i. For $i = m-k+1$, we have

$$\{q_{m-k+1}, 0, j_2, \ldots, j_s\} = \delta'(\{q_{m-k-1}, j_2 - 1, \ldots, j_s - 1\}, aab^{n-1}),$$

where the latter set is reachable by induction on s. Next, since

$$\{q_{i+1}, 0, j_2, \ldots, j_s\} = \delta'(\{q_i, 0, j_2, \ldots, j_s\}, ab^{n-1})$$

for $i = m-k+1, m-k+2, \ldots, m-2$, we are ready in this case.

(ii) $i = 0$. In the case of $k = 1$, we have

$$\{q_0, 0, j_2, \ldots, j_s\} = \delta'(\{q_{m-2}, j_2 - 1, \ldots, j_s - 1\}, aab^{n-1}),$$

and for $j_1 \geq 1$,

$$\{q_0, j_1, j_2, \ldots, j_s\} = \delta'(\{q_{m-2}, j_2 - j_1 - 1, \ldots, j_s - j_1 - 1\}, aab^{j_1-1}),$$

where the sets $\{q_{m-2}, j_2 - 1, \ldots, j_s - 1\}$ and $\{q_{m-2}, j_2 - j_1 - 1, \ldots, j_s - j_1 - 1\}$ of size s are reachable by induction.

In the case of $k \geq 2$, we have

$$\{q_0, 0, j_2, \ldots, j_s\} = \delta'(\{q_{m-1}, 0, j_2, \ldots, j_s\}, ab^{n-1}),$$

and for $j_1 \geq 1$,

$$\{q_0, j_1, j_2, \ldots, j_s\} = \delta'(\{q_{m-1}, 0, j_2 - j_1, \ldots, j_s - j_1\}, ab^{j_1-1}),$$

where the sets $\{q_{m-1}, 0, j_2, \ldots, j_s\}$ and $\{q_{m-1}, 0, j_2 - j_1, \ldots, j_s - j_1\}$ are considered in case (i).

(iii) $1 \leq i \leq m-k-1$. Then we have

$$\{q_i, j_1, j_2, \ldots, j_s\} = \delta'(\{q_0, (j_1 - i) \bmod n, \ldots, (j_s - i) \bmod n\}, a^i),$$

where the latter set is considered in case (ii).

(iv) $i = m - k$ and $j_1 = 0$. Then we have

$$\{q_{m-k}, 0, j_2, \ldots, j_s\} = \delta'(\{q_{m-k-1}, j_2 - 1, \ldots, j_s - 1, n-1\}, a),$$

where the latter set is considered in case (iii).

To prove (II) let $\{q_i\} \cup S$ and $\{q_j\} \cup T$ be two different states in the system $\mathcal{R}$ with $0 \leq i \leq j \leq m-1$. There are two cases:

(i) $i = j$. Without loss of generality, there is a state l in Q_B such that $l \in S$ and $l \notin T$ (note that $l \geq 1$ if $m - k \leq i \leq m-1$). Then, the string a^{n-1-l} is accepted by the DFA C' starting in state $\{q_i\} \cup S$ but it is not accepted by the DFA C' starting in state $\{q_j\} \cup T$.

(ii) $i < j$. We will consider two subcases:

(a) $j - i \leq k$. Let $v = a^{m-1-j}b^n aab^{n-2}$. Then

$$q_{m-j+i} \in \delta(q_i, a^{m-1-j}b^n a),$$

where $m - k \leq m - j + i \leq m - 1$. It follows that

$$n - 1 \in \delta(q_{m-j+i}, ab^{n-2}),$$

and so the string v is accepted by the DFA C' starting in state $\{q_i\} \cup S$. On the other hand, we have

$$\delta'(\{q_j\} \cup T, a^{m-1-j}b^n a) = \{q_0, 1\} \text{ and } \delta'(\{q_0, 1\}, ab^{n-2}) = \{q_1, 0\},$$

so the string v is not accepted by the DFA C' starting in state $\{q_j\} \cup T$.

(b) $j - i > k$. Let $w = a^{m-1-i-k}b^n aab^{n-2}$. Then

$$q_{m-k} \in \delta(q_i, a^{m-1-i-k}b^n a) \text{ and } n - 1 \in \delta(q_{m-k}, ab^{n-2}).$$

It follows that the string w is accepted by the DFA C' starting in state $\{q_i\} \cup S$. On the other hand, we have

$$\delta'(\{q_j\} \cup T, a^{m-1-i-k}b^n a) = \{q_{(m-i-k+j) \bmod m}, 1\},$$

where $(m - i - k + j) \bmod m = j - i - k \leq m - k - 1$, and so

$$\delta'(\{q_{j-i-k}, 1\}, ab^{n-2}) = \{q_{j-i-k+1}, 0\}.$$

Thus the string w is not accepted by the DFA C' starting in state $\{q_j\} \cup T$ which completes our proof. □

Chapter 4

Magic Numbers

We now turn our attention to a different problem. This time, we will be interested not only in the wort-case complexity but also in all values that can be obtained as the complexity of an operation. We are asking whether all values between the lower and upper bound on the complexity can be reached, or whether there are some holes in the hierarchy that are called magic numbers in the literature [23, 13, 49].

4.1 NFA to DFA Conversion

Iwama at al. [22] stated the question of whether there always exists a minimal nondeterministic finite automaton (NFA) of n states whose equivalent minimal deterministic finite automaton (DFA) has α states for all integers n and α such that $n \leqslant \alpha \leqslant 2^n$. The question has also been considered in [23]. In these two papers, it is shown that if $\alpha = 2^n - 2^k$ or $\alpha = 2^n - 2^k - 1$, where $0 \leqslant k \leqslant n/2 - 2$, or if $\alpha = 2^n - k$, where $2 \leqslant k \leqslant 2n - 2$ and some coprimality condition holds, then the corresponding binary n-state NFAs requiring α deterministic states do exist. In [26], appropriate NFAs has been described for all values of n and α, however, the size of the input alphabet for these automata grows exponentially with n. Later, in [12], the size of the input alphabet for the witness automata has been reduced to $n + 2$.

In this section, we continue the research on this topic. We reduce the input alphabet to a fixed size. We prove that for all integers n and α such that $n \leqslant \alpha \leqslant 2^n$, there exists a minimal nondeterministic finite automaton of n states with a four-letter input alphabet whose equivalent minimal deterministic finite automaton has exactly α states. Using terminology of [13], this means that in the case of a four-letter alphabet, there are no magic numbers, i.e., the holes in the hierarchy that cannot be reached as the size of

a minimal DFA corresponding to a minimal n-state NFA. Let us note that in the case of a unary alphabet, all numbers from $e^{(1+o(1))\cdot\sqrt{n \ln n}}$ to 2^n are known to be magic since every n-state unary NFA can be simulated by an $e^{(1+o(1))\cdot\sqrt{n \ln n}}$-state DFA [34, 8, 13]. Moreover, it has been recently shown in [13] that there are much more magic than non-magic numbers in the range from n to $e^{(1\pm o(1))\cdot(\sqrt{n \ln n})}$ in the unary case. The question of whether or not there are some magic numbers for binary and ternary alphabets seems to be a challenging open problem.

To describe appropriate NFAs we use the following result showing that every integer can be expressed as a sum of powers of 2 decreased by 1, if the smallest summand can possibly be taken twice.

Lemma 2. *Let k be a positive integer. Then for each integer m such that $1 \leqslant m < 2^k$, one of the following three cases holds:*

$$\begin{aligned} m &= 2^k - 1 && (4.1)\\ m &= (2^{k_1} - 1) + (2^{k_2} - 1) + \cdots + (2^{k_{\ell-1}} - 1) + (2^{k_\ell} - 1) && (4.2)\\ m &= (2^{k_1} - 1) + (2^{k_2} - 1) + \cdots + (2^{k_{\ell-1}} - 1) + 2 \cdot (2^{k_\ell} - 1) && (4.3) \end{aligned}$$

where $1 \leqslant \ell \leqslant k-1$, and $k - 1 \geqslant k_1 > k_2 > \cdots > k_\ell \geqslant 1$.

Proof. We prove the lemma by induction on k.

Let $k = 1$. Then $m = 1 = 2^1 - 1$, and so (4.1) holds.

Let $k > 1$ and assume that the lemma holds for all integers less then k. Using this assumption we prove that it also holds for k.

Let $1 \leqslant m < 2^k$. If $m = 2^k - 1$, then (4.1) holds. Otherwise, let r be the greatest integer such that $m \geqslant 2^r - 1$. Then $r \leqslant k - 1$ and $m = 2^r - 1 + s$, where $0 \leqslant s < 2^r$. If $s = 0$, then $m = 2^r - 1$, and so (4.2) holds. Otherwise, by induction,

$$\begin{aligned} s &= 2^r - 1, \text{ or}\\ s &= (2^{k_1} - 1) + (2^{k_2} - 1) + \cdots + (2^{k_{\ell-1}} - 1) + (2^{k_\ell} - 1), \text{ or}\\ s &= (2^{k_1} - 1) + (2^{k_2} - 1) + \cdots + (2^{k_{\ell-1}} - 1) + 2 \cdot (2^{k_\ell} - 1), \end{aligned}$$

where $1 \leqslant \ell \leqslant r - 1$, and $r - 1 \geqslant k_1 > k_2 > \cdots > k_\ell \geqslant 1$. Then we have

$$\begin{aligned} m &= 2 \cdot (2^r - 1), \text{ or}\\ m &= (2^r - 1) + (2^{k_1} - 1) + (2^{k_2} - 1) + \cdots + (2^{k_{\ell-1}} - 1) + (2^{k_\ell} - 1), \text{ or}\\ m &= (2^r - 1) + (2^{k_1} - 1) + (2^{k_2} - 1) + \cdots + (2^{k_{\ell-1}} - 1) + 2 \cdot (2^{k_\ell} - 1), \end{aligned}$$

where $1 \leqslant \ell + 1 \leqslant r \leqslant k - 1$, and $k - 1 \geqslant r > k_1 > k_2 > \cdots > k_l \geqslant 1$. This concludes our proof. □

We start by describing two nondeterministic finite automata that we will use later in our constructions. We prove several properties concerning these two automata in the two lemmata below.

First, let us consider a k-state NFA $A_k = (Q_A, \{a, b\}, \delta_A, 1, F_A)$, where $Q_A = \{1, 2, \ldots, k\}$, $F_A = \{k\}$, and for each i in Q_A,

$$\delta_A(i, a) = \begin{cases} \{1, i+1\}, & \text{if } 1 \leqslant i \leqslant k-1, \\ \varnothing, & \text{if } i = k, \end{cases}$$

$$\delta_A(i, b) = \begin{cases} \{i+1\}, & \text{if } 1 \leqslant i \leqslant k-1, \\ \varnothing, & \text{if } i = k. \end{cases}$$

The automaton A_k is depicted in Figure 4.1. The next lemma shows that every subset of the state set Q_A is a reachable state in the deterministic finite automaton obtained from the NFA A_k by the subset construction.

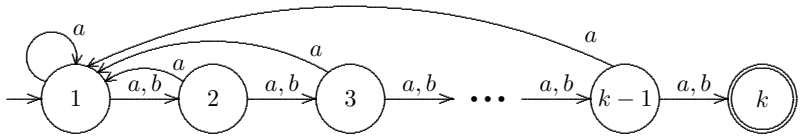

Figure 4.1: The nondeterministic finite automaton A_k.

Lemma 3. *Let $A'_k = (2^{Q_A}, \{a, b\}, \delta'_A, \{1\}, F'_A)$ be the DFA obtained from the NFA A_k by the subset construction. Then every subset of the state set Q_A is reachable in the DFA A'_k.*

Proof. The proof is by induction on the cardinality of subsets. The empty set and all the singletons are reachable because

$$\varnothing = \delta'_A(\{1\}, b^k) \text{ and } \{i\} = \delta'_A(\{1\}, b^{i-1}) \text{ for all } i = 1, 2, \ldots, k.$$

Let $2 \leqslant t \leqslant k$ and assume by induction that every subset of the state set Q_A of size $t-1$ is reachable in the DFA A'_k. Let $\{i_1, i_2, \ldots, i_t\}$ be a subset of size t such that $1 \leqslant i_1 < i_2 < \cdots < i_t \leqslant k$. Then

$$\{i_1, i_2, \ldots, i_t\} = \delta'_A(\{i_2 - i_1, i_3 - i_1, \ldots, i_t - i_1\}, ab^{i_1 - 1}),$$

where the latter subset of size $t-1$ is reachable by induction. Thus the set $\{i_1, i_2, \ldots, i_t\}$ is reachable and our proof is complete. □

Now, consider the following $(k+1)$-state NFA $B_k = (Q_B, \{a, b\}, \delta_B, 0, \{k\})$, where $Q_B = \{0, 1, 2, \ldots, k\}$, and for each i in Q_B,

$$\delta_B(i, a) = \begin{cases} \{0, 1\}, & \text{if } i = 0, \\ \{i + 1\}, & \text{if } 1 \leqslant i \leqslant k - 1, \\ \{1, 2, \ldots, k\}, & \text{if } i = k, \end{cases}$$

$$\delta_B(i, b) = \begin{cases} \{0\}, & \text{if } i = 0, \\ \{i + 1\}, & \text{if } 1 \leqslant i \leqslant k - 1, \\ \{1, 2, \ldots, k\}, & \text{if } i = k. \end{cases}$$

The automaton B_k is shown in Figure 4.2. Note that if we would omit all the transitions defined in the final state k, then the resulting automaton would accept all strings containing a symbol a in the k-th position from the end.

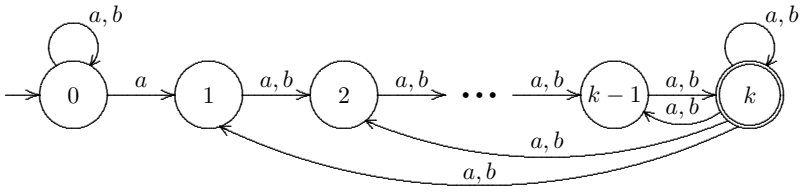

Figure 4.2: The nondeterministic finite automaton B_k.

Let B'_k be the deterministic finite automaton obtained from the NFA B_k by the subset construction. The DFA B'_4 (or, to be more precise, its reachable states, each of which contains the initial state 0 of the NFA B_4) is shown in Figure 4.3. The automaton in the figure looks like a binary tree whose leaves go to state $\{0, 1, 2, 3, 4\}$ on a and b.

In the following, we will consider states $\{0, 2\}, \{0, 2, 3\}, \ldots, \{0, 2, 3, \ldots, r\}, \ldots, \{0, 2, 3, \ldots, k\}$ of the DFA B'_k. Notice that

$$\{0, 2\} \subset \{0, 2, 3\} \subset \ldots \subset \{0, 2, 3, \ldots, r\} \subset \ldots \subset \{0, 2, 3, \ldots, k\}$$

which is a property that will play a crucial role in the proof of our main result. Before stating the next lemma we introduce some notation. Let $2 \leqslant r \leqslant k$. Let

$$\mathcal{R}_{1,1} = \{R \subseteq Q_B \mid R = \delta'_B(\{0, 1\}, w) \text{ for some } w \text{ in } \{a, b\}^*\},$$

$$\mathcal{R}_{1,r} = \{R \subseteq Q_B \mid R = \delta'_B(\{0, 1, 2, 3, \ldots, r\}, w) \text{ for some } w \text{ in } \{a, b\}^*\},$$

$$\mathcal{R}_{2,r} = \{R \subseteq Q_B \mid R = \delta'_B(\{0, 2, 3, 4, \ldots, r\}, w) \text{ for some } w \text{ in } \{a, b\}^*\},$$

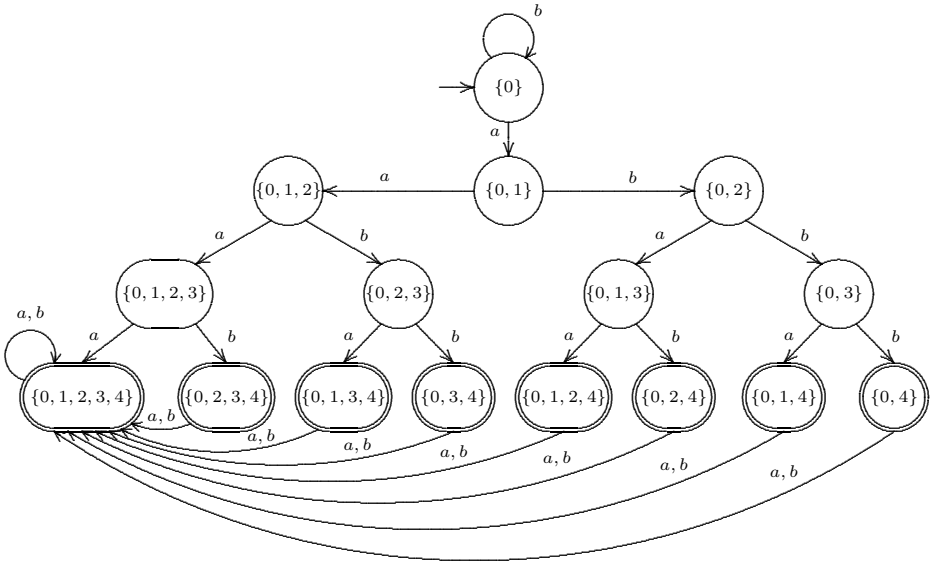

Figure 4.3: The deterministic finite automaton B'_4.

that is, $\mathcal{R}_{1,1}$, $\mathcal{R}_{1,r}$, and $\mathcal{R}_{2,r}$ are the sets of states of the DFA B'_k that are reachable from states $\{0,1\}$, $\{0,1,2,3,\ldots,r\}$, and $\{0,2,3,4,\ldots,r\}$, respectively. For example, in our Figure 4.3, we have

$\mathcal{R}_{1,3} = \big\{\{0,1,2,3\},\{0,1,2,3,4\},\{0,2,3,4\}\big\}$ and
$\mathcal{R}_{2,3} = \big\{\{0,2,3\},\{0,1,3,4\},\{0,3,4\},\{0,1,2,3,4\}\big\}$.

Lemma 4. *Let $2 \leqslant s \leqslant r \leqslant k$ and let $\mathcal{R}_{1,1}$, $\mathcal{R}_{1,r}$, and $\mathcal{R}_{2,r}$ be as above. Then we have:*

(i) State $\{0,1,2,3,\ldots,k\}$ is a member of the sets $\mathcal{R}_{1,1}$, $\mathcal{R}_{1,r}$, and $\mathcal{R}_{2,r}$.

(ii) The size of the set $\mathcal{R}_{1,1}$ is $2^k - 1$.

(iii) The size of the set $\mathcal{R}_{1,r}$ and of the set $\mathcal{R}_{2,r} \setminus \{\{0,1,2,3,\ldots,k\}\}$ is $2^{k-r+1} - 1$.

(iv) The sets $\mathcal{R}_{1,r}$ and $\mathcal{R}_{2,s}$ have only state $\{0,1,2,3,\ldots,k\}$ in common.

(v) If $s < r$, then the sets $\mathcal{R}_{2,s}$ and $\mathcal{R}_{2,r}$ have only state $\{0,1,2,3,\ldots,k\}$ in common.

Proof. First, notice that each reachable state of the DFA B'_k contains the initial state 0 of the NFA B_k since state 0 goes to itself on a and b in the NFA B_k.

To prove *(i)* note that state k of the NFA B_k is reachable from each state of this NFA and the transitions on a and b from state k go to $\{1, 2, \ldots, k\}$.

To prove the rest of the lemma let us see how the sets $\mathcal{R}_{1,r}$ and $\mathcal{R}_{2,r}$ look like. Figures 4.4 and 4.5 show which states are reachable from states $\{0, 1, 2, 3, \ldots, r\}$ and $\{0, 2, 3, 4, \ldots, r\}$, respectively, by a string of length at most two.

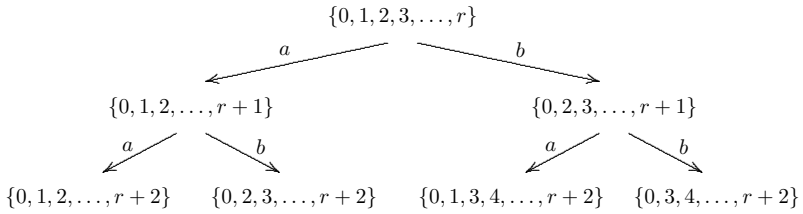

Figure 4.4: The states reachable from state $\{0, 1, 2, 3, \ldots, r\}$ by $\varepsilon, a, b, aa, ab, ba, bb$.

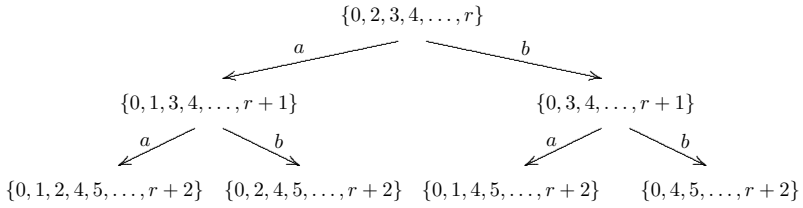

Figure 4.5: The states reachable from state $\{0, 2, 3, 4, \ldots, r\}$ by $\varepsilon, a, b, aa, ab, ba, bb$.

Let $1 \leqslant r \leqslant k$. We prove that the set of states that are reachable from state $\{0, 1, 2, 3, \ldots, r\}$ by strings of length ℓ with $0 \leqslant \ell \leqslant k - r$ is

$$\big\{\{0\} \cup S \cup \{1 + \ell, 2 + \ell, 3 + \ell, \ldots, r + \ell\} \mid S \subseteq \{1, 2, \ldots, \ell\}\big\}.$$

The proof is by induction on ℓ. The claim trivially holds for $\ell = 0$. Assume that $0 \leqslant \ell \leqslant k-r-1$, and that the claim holds for ℓ. Let us show that it also holds for $\ell + 1$. Let w be a string in $\{a, b\}^*$ of length $\ell + 1$. Then $w = va$ or $w = vb$, where v is a string of length ℓ. By induction, state $\{0, 1, 2, 3, \ldots, r\}$ goes by the string v to a state $\{0\} \cup S \cup \{1 + \ell, 2 + \ell, 3 + \ell, \ldots, r + \ell\}$ for some subset S of $\{1, 2, \ldots, \ell\}$. This state goes by a to state

$$\{0\} \cup \{1\} \cup \{j + 1 \mid j \in S\} \cup \{1 + \ell + 1, 2 + \ell + 1, 3 + \ell + 1, \ldots, r + \ell + 1\},$$

and by b to state

$$\{0\} \cup \{j+1 \mid j \in S\} \cup \{1+\ell+1, 2+\ell+1, 3+\ell+1, \ldots, r+\ell+1\},$$

where $\{1\} \cup \{j+1 \mid j \in S\}$ and $\{j+1 \mid j \in S\}$ are subsets of $\{1, 2, \ldots, \ell+1\}$. Now, let S' is a subset of $\{1, 2, \ldots, \ell+1\}$. We need to show that the state

$$\{0\} \cup S' \cup \{1+\ell+1, 2+\ell+1, 3+\ell+1, \ldots, r+\ell+1\} \tag{4.4}$$

is reachable from state $\{0, 1, 2, 3, \ldots, r\}$ by a string of length $\ell+1$. If $1 \notin S'$, then the set (4.4) can be reached from the set

$$\{0\} \cup \{j-1 \mid j \in S'\} \cup \{1+\ell, 2+\ell, 3+\ell, \ldots, r+\ell\} \tag{4.5}$$

by b. If $1 \in S'$, then the set (4.4) can be reached from the set

$$\{0\} \cup \{j-1 \mid j \in S' \text{ and } j \neq 1\} \cup \{1+\ell, 2+\ell, 3+\ell, \ldots, r+\ell\} \tag{4.6}$$

by a. By the induction hypothesis, the sets (4.5) and (4.6) are reachable from state $\{0, 1, 2, 3, \ldots, r\}$ by a string of length ℓ, which completes the proof of our claim.

Next, note that state $\{0, 1, 2, 3, \ldots, r\}$ goes to state $\{0, 1, 2, 3, \ldots, k\}$ by the string a^{k-r}, and that each set reachable from state $\{0, 1, 2, 3, \ldots, r\}$ by a string of length $k-r$ contains state k which goes to $\{1, 2, 3, \ldots, k\}$ on a and b in the NFA B_k. It follows that state $\{0, 1, 2, 3, \ldots, r\}$ goes to state $\{0, 1, 2, 3, \ldots, k\}$ by each string of length more then $k-r$. Thus the size of the set $\mathcal{R}_{1,r}$ is $1 + 2 + 4 + \cdots + 2^{k-r}$.

Now let $2 \leqslant r \leqslant k$. In a similar way as above, we can show that the set of states that are reachable from state $\{0, 2, 3, 4, \ldots, r\}$ by the strings of length ℓ with $0 \leqslant \ell \leqslant k-r$ is $\big\{\{0\} \cup S \cup \{2+\ell, 3+\ell, \ldots, r+\ell\} \mid S \subseteq \{1, 2, \ldots, \ell\}\big\}$, and that state $\{0, 2, 3, 4, \ldots, r\}$ goes to state $\{0, 1, 2, 3, \ldots, k\}$ by each string of length more then $k-r$. Hence $|\mathcal{R}_{2,r} \setminus \{\{0, 1, 2, 3, \ldots, k\}\}| = 1 + 2 + 4 + \cdots + 2^{k-r} = 2^{k-r+1} - 1$.

To show *(iv)* let $2 \leqslant s \leqslant r \leqslant k$. Assume that there is a state different from $\{0, 1, 2, \ldots, k\}$ that is in $\mathcal{R}_{1,r} \cap \mathcal{R}_{2,s}$. Then there must be ℓ and ℓ' with $0 \leqslant \ell, \ell' \leqslant k-r$, and some subsets S of $\{1, 2, \ldots, \ell\}$ and S' of $\{1, 2, \ldots, \ell'\}$ such that the sets

$$\{0\} \cup S \cup \{1+\ell, 2+\ell, 3+\ell, \ldots, r+\ell\} \tag{4.7}$$

$$\{0\} \cup S' \cup \{2+\ell', 3+\ell', \ldots, s+\ell'\} \tag{4.8}$$

are equal. This means that $r + \ell = s + \ell'$. Then $1 + \ell' = 1 + r - s + \ell$, which follows that $1 + \ell'$ is in the set (4.7). However, $1 + \ell'$ is not in the set (4.8). This is a contradiction.

Finally, to show *(v)* let $2 \leqslant s < r \leqslant k$, and assume that there is a state different from $\{0, 1, 2, \ldots, k\}$ that is in $\mathcal{R}_{2,s} \cap \mathcal{R}_{2,r}$. Then for some ℓ and ℓ' with $0 \leqslant \ell, \ell' \leqslant k - r$, and some subsets S of $\{1, 2, \ldots, \ell\}$ and S' of $\{1, 2, \ldots, \ell'\}$, the sets

$$\{0\} \cup S \cup \{2 + \ell, 3 + \ell, \ldots, s + \ell\} \tag{4.9}$$
$$\{0\} \cup S' \cup \{2 + \ell', 3 + \ell', \ldots, r + \ell'\} \tag{4.10}$$

are equal. Thus, $s + \ell = r + \ell'$ and so $1 + \ell = 1 + r - s + \ell'$. This follows that $1 + \ell$ is in the set (4.10). We again have a contradiction since $1 + \ell$ is not in the set (4.9). □

We are now ready to prove the main result of this section showing that in the case of a four-letter alphabet, there are no "magic numbers", that is, each value in the range from n to 2^n can be obtained as the deterministic state complexity of an n-state NFA language over a four-letter alphabet.

Theorem 3. *For all integers n and α such that $n \leqslant \alpha \leqslant 2^n$, there exists a minimal nondeterministic finite automaton of n states with a four-letter input alphabet whose equivalent minimal deterministic finite automaton has α states.*

Proof. Let n and α be arbitrary but fixed integers such that $n \leqslant \alpha \leqslant 2^n$. If $\alpha = n$, then we can consider a unary n-state NFA that counts the numbers of a's modulo n. On the other hand, the n-state NFAs that need 2^n deterministic states are well-known [33, 36, 26] (and also the NFA A_n described above requires 2^n deterministic states).

Let $n < \alpha < 2^n$. Then there is an integer k such that $1 \leqslant k \leqslant n - 1$ and

$$n - k + 2^k \leqslant \alpha < n - (k + 1) + 2^{k+1}.$$

It follows that

$$\alpha = n - (k + 1) + 2^k + m,$$

where m is an integer such that $1 \leqslant m < 2^k$. By Lemma 2, for this integer m, one of the following three cases holds:

$$m = 2^k - 1 \tag{4.11}$$
$$m = (2^{k_1} - 1) + (2^{k_2} - 1) + \cdots + (2^{k_{\ell-1}} - 1) + (2^{k_\ell} - 1) \tag{4.12}$$
$$m = (2^{k_1} - 1) + (2^{k_2} - 1) + \cdots + (2^{k_{\ell-1}} - 1) + 2 \cdot (2^{k_\ell} - 1) \tag{4.13}$$

where $1 \leqslant \ell \leqslant k - 1$, and $k - 1 \geqslant k_1 > k_2 > \cdots > k_{\ell-1} > k_\ell \geqslant 1$.

Define an n-state NFA $C = C_{n,k,m} = (Q, \{a, b, c, d\}, \delta, q_0, \{k\})$, where $Q = \{0, 1, 2, \ldots, n-1\}$, $q_0 = n-1$ if $k < n-1$ and $q_0 = 1$ if $k = n-1$, and for each i in Q,

$$\delta(i,a) = \begin{cases} \{0,1\}, & \text{if } i = 0, \\ \{1, i+1\}, & \text{if } 1 \leqslant i \leqslant k-1, \\ \{1,2,\ldots,k\}, & \text{if } i = k, \\ \varnothing & \text{if } k+1 \leqslant i \leqslant n-1, \end{cases}$$

$$\delta(i,b) = \begin{cases} \{0\}, & \text{if } i = 0, \\ \{i+1\}, & \text{if } 1 \leqslant i \leqslant k-1, \\ \{1,2,\ldots,k\}, & \text{if } i = k, \\ \{1\}, & \text{if } i = k+1, \\ \{i-1\}, & \text{if } k+2 \leqslant i \leqslant n-1, \end{cases}$$

$$\delta(i,c) = \begin{cases} \{i+1\}, & \text{if } 0 \leqslant i \leqslant k-1, \\ \varnothing, & \text{if } k \leqslant i \leqslant n-1. \end{cases}$$

Transitions on d are defined depending on m as follows:
If (4.11) holds for m, then

$$\delta(i,d) = \begin{cases} \{0,1\}, & \text{if } i = 1, \\ \varnothing, & \text{otherwise.} \end{cases}$$

If (4.12) holds for m, then

$$\delta(i,d) = \begin{cases} \{0,2,3,4,\ldots,k-k_i+1\}, & \text{if } 1 \leqslant i \leqslant \ell-1, \\ \{0,1,2,3,\ldots,k-k_\ell+1\}, & \text{if } i = \ell, \\ \varnothing, & \text{otherwise.} \end{cases}$$

If (4.13) holds for m, then

$$\delta(i,d) = \begin{cases} \{0,2,3,4,\ldots,k-k_i+1\}, & \text{if } 1 \leqslant i \leqslant \ell, \\ \{0,1,2,3,\ldots,k-k_\ell+1\}, & \text{if } i = \ell+1, \\ \varnothing, & \text{otherwise.} \end{cases}$$

The NFA $C_{n,k,m}$ for $m = 2^{k-1} - 1$ is shown in Figure 4.6; notice that in this case, transitions on d are defined only in state 1 and go to $\{0,1,2\}$. Figure 4.7 shows transitions on b and d in the NFA $C_{n,k,m}$ for

$$m = (2^{k-1} - 1) + 2 \cdot (2^{k-2} - 1);$$

here, transitions on d are defined in states 1, 2, and 3 and go to sets $\{0,2\}, \{0,2,3\}$, and $\{0,1,2,3\}$, respectively.

Note that the transitions on a and b in states $1, 2, \ldots, k-1$ are defined in the same way as for the automaton A_k, while the transitions on these two letters in states $0, 1, 2, \ldots, k$ are defined as for the automaton B_k except for transitions on a going to state 1 from states $1, 2, \ldots, k-1$. Transitions on d are defined in states $1, 2, \ldots, \ell$ if (4.12) holds for m, and in states $1, 2, \ldots, \ell, \ell+1$ if (4.13) hold for m, and go either to a set $\{0, 2, 3, 4, \ldots, r\}$ or to a set $\{0, 1, 2, 3, \ldots, r\}$. As we will see later, this assures that all subsets of the set $\{1, 2, \ldots, k\}$ and m subsets of the set $\{0, 1, 2, \ldots, k\}$ containing state 0 are reachable in the DFA C' obtained from the NFA C by the subset construction. Transitions on c will be used to prove the inequivalence of these reachable subsets.

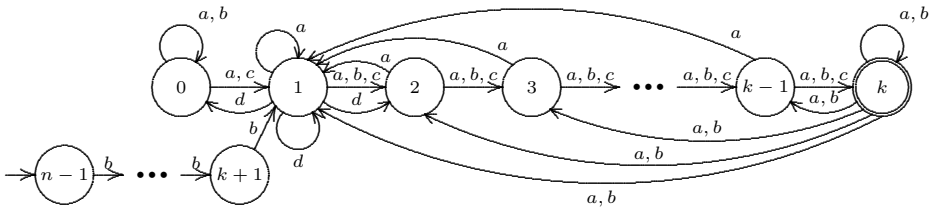

Figure 4.6: The NFA $C_{n,k,m}$, where $m = 2^{k-1} - 1$.

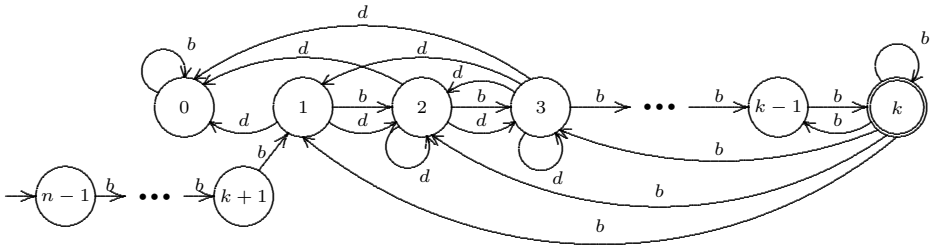

Figure 4.7: Transitions on b, d in $C_{n,k,m} : m = (2^{k-1} - 1) + 2 \cdot (2^{k-2} - 1)$.

We are going to prove that:

(i) The NFA C is minimal.

(ii) The DFA C' obtained from the NFA C by the subset construction has $n - (k+1) + 2^k + m$ reachable states.

(iii) The reachable states of the DFA C' are pairwise inequivalent.

Then, since $\alpha = n - (k+1) + 2^k + m$, the theorem follows immediately.

To prove *(i)* consider the following sets of pairs of strings

$$\begin{aligned}\mathcal{A} &= \big\{(b^{i-1}, b^{n-k-i}c^{k-1}) \mid i = 1, 2, \ldots, n-k\big\},\\ \mathcal{B} &= \big\{(b^{n-k-1}c^{i}, c^{k-1-i}) \mid i = 1, 2, \ldots, k-1\big\},\\ \mathcal{C} &= \big\{(b^{n-k-1}d, c^{k})\big\}.\end{aligned}$$

We will prove that the set $\mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$ is a fooling set for the language $L(C)$. We need to show that (1) and (2) in Definition 1 hold.

To prove (1) note that the strings $b^{n-k-1}c^{k-1}$ and $b^{n-k-1}dc^{k}$ are in the language $L(C)$ since the initial state of the NFA C goes to state 1 by the string b^{n-k-1} and then to the accepting state k by the string c^{k-1}; next, state 1 goes to state 0 by d and then to the accepting state k by the string c^{k}.

To prove (2) we have five cases to consider:

(a) Let $(b^{i-1}, b^{n-k-i}c^{k-1})$ and $(b^{j-1}, b^{n-k-j}c^{k-1})$, where $1 \leqslant i < j \leqslant n-k$, be two different pairs in $\mathcal{A}$; note that in such a case we have $k < n-1$. Then the string $b^{i-1}b^{n-k-j}c^{k-1}$ is not in the language $L(C)$ since the initial state $n-1$ goes by $b^{n-k-1-j+i}$ to rejecting state $k+j-i$, in which transitions on c are not defined because $k+j-i > k$.

(b) Let $(b^{n-k-1}c^{i}, c^{k-1-i})$ and $(b^{n-k-1}c^{j}, c^{k-1-j})$ where $1 \leqslant i < j \leqslant k-1$, be two different pairs in $\mathcal{B}$. Then the string $b^{n-k-1}c^{j}c^{k-1-i}$ is not in the language $L(C)$ since the initial state goes to state 1 by the string b^{n-k-1}, from which the string $c^{k-1+j-i}$ is not accepted because $k-1+j-i > k-1$.

(c) Let $(b^{i-1}, b^{n-k-i}c^{k-1})$, where $1 \leqslant i \leqslant n-k$, be a pair in $\mathcal{A}$ and $(b^{n-k-1}c^{j}, c^{k-1-j})$, where $1 \leqslant j \leqslant k-1$, be a pair in $\mathcal{B}$. Then the string $b^{i-1}c^{k-1-j}$ is not in the language $L(C)$ since the initial state goes by b^{i-1} either to a state from $\{k+1, k+2, \ldots, n-1\}$ or to state 1. From no one of these states, the string c^{k-1-j} is accepted.

(d) Let $(b^{i-1}, b^{n-k-i}c^{k-1})$, where $1 \leqslant i \leqslant n-k$, be a pair in $\mathcal{A}$ and $(b^{n-k-1}d, c^{k})$ be the pair in $\mathcal{C}$. Then the string $b^{i-1}c^{k}$ is not in the language $L(C)$ since the string c^{k} is accepted neither from a state in $\{k+1, k+2, \ldots, n-1\}$ nor from state 1.

(e) Let $(b^{n-k-1}c^{j}, c^{k-1-j})$, where $1 \leqslant j \leqslant k-1$, be a pair in $\mathcal{B}$ and $(b^{n-k-1}d, c^{k})$ be the pair in $\mathcal{C}$. Then the string $b^{n-k-1}c^{j}c^{k}$ is not in the language $L(C)$ since the initial state goes by b^{n-k-1} to state 1, from which the string c^{k+j} is not accepted.

Thus the set $\mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$ is a fooling set for the language $L(C)$ of size n. By Lemma 1, every NFA for the language $L(C)$ needs at least n states. So the NFA C is minimal and *(i)* is proved.

To prove *(ii)* let $C' = (2^Q, \{a, b, c, d\}, \delta', q'_0, F')$ be the DFA obtained from the NFA C by the subset construction. Consider the following systems $\mathcal{S}_1, \mathcal{S}_2$, and $\mathcal{S}_3$ of sets of states of the NFA C (remind that the sets $\mathcal{R}_{i,r}$ were defined before Lemma 4 on page 18):

$$\begin{aligned}
\mathcal{S}_1 &= \big\{\{n-1\}, \{n-2\}, \ldots, \{k+1\}\big\} \cup 2^{\{1,2,\ldots,k\}} \cup \mathcal{R}_{1,1} \\
\mathcal{S}_2 &= \big\{\{n-1\}, \{n-2\}, \ldots, \{k+1\}\big\} \cup 2^{\{1,2,\ldots,k\}} \cup \\
&\quad \mathcal{R}_{2,k-k_1+1} \cup \mathcal{R}_{2,k-k_2+1} \cup \cdots \cup \mathcal{R}_{2,k-k_{\ell-1}+1} \cup \mathcal{R}_{1,k-k_\ell+1}, \\
\mathcal{S}_3 &= \big\{\{n-1\}, \{n-2\}, \ldots, \{k+1\}\big\} \cup 2^{\{1,2,\ldots,k\}} \cup \\
&\quad \mathcal{R}_{2,k-k_1+1} \cup \mathcal{R}_{2,k-k_2+1} \cup \cdots \cup \mathcal{R}_{2,k-k_{\ell-1}+1} \cup \mathcal{R}_{2,k-k_\ell+1} \cup \mathcal{R}_{1,k-k_\ell+1}.
\end{aligned}$$

We will prove that $\mathcal{S}_1$ ($\mathcal{S}_2, \mathcal{S}_3$) is the system of all reachable states of the DFA C' in the case that (4.11) holds for m ((4.12), (4.13) holds for m, respectively). We give the proof for $\mathcal{S}_2$; the other cases are similar.

Let $m = (2^{k_1} - 1) + (2^{k_2} - 1) + \cdots + (2^{k_{\ell-1}} - 1) + (2^{k_\ell} - 1)$, where $1 \leqslant \ell \leqslant k-1$, and $k - 1 \geqslant k_1 > k_2 > \cdots > k_\ell \geqslant 1$. We need to show that each set in $\mathcal{S}_2$ is a reachable state of the DFA C' and that no other subset of the state set Q is reachable in C'.

The singletons $\{n-1\}, \{n-2\}, \ldots, \{k+1\}$, and $\{1\}$ are reachable since they can be reached from the initial state of the DFA C' by reading an appropriate numbers of b's.

By Lemma 3, every nonempty subset of the set $\{1, 2, \ldots, k\}$ is reachable because state $\{1\}$ is reachable and the transitions on a and b in states $1, 2, \ldots, k-1$, that were used in the proof of Lemma 3, are the same as in the NFA A_k. The empty set is reachable since $\varnothing = \delta'(\{1\}, c^k)$.

Next, state $\{i\}$, which is reachable from state $\{1\}$ by b^{i-1}, goes by d to state $\{0, 2, 3, \ldots, k - k_i + 1\}$ for $i = 1, 2, \ldots, \ell - 1$, and state $\{\ell\}$ goes by d to state $\{0, 1, 2, 3 \ldots, k - k_\ell + 1\}$. The reachability of the sets $\mathcal{R}_{2,k-k_1+1}$, $\mathcal{R}_{2,k-k_2+1}, \ldots, \mathcal{R}_{2,k-k_{\ell-1}+1}$, and $\mathcal{R}_{1,k-k_\ell+1}$ then follows from their definition and the fact that the transitions on a and b in states $0, 1, 2, \ldots, k$ are almost the same as in the NFA B_k; notice that the transitions on a to state 1 do not mind since the sets in $\mathcal{R}_{i,r}$ always contain state 0 and this state goes to state 1 on a in the NFA B_k.

Thus each set in the system $\mathcal{S}_2$ is a reachable state of the DFA C'. To prove that no other subset of the state set Q is reachable it is sufficient to show that for each set S in the system $\mathcal{S}_2$, the sets $\delta'(S, a), \delta'(S, b), \delta'(S, c)$, and $\delta'(S, d)$ are again in the system $\mathcal{S}_2$. Let us show this first for the symbols a, b, and c.

If S is one of the singletons $\{n-1\}, \{n-2\}, \ldots, \{k+1\}$, then the sets $\delta'(S,a)$ and $\delta'(S,c)$ are empty, and the set $\delta'(S,b)$ is either one of the singletons $\{n-2\}$, $\{n-3\}, \ldots, \{k+1\}$, or is equal to $\{1\}$. If S is a subset of the set $\{1,2,\ldots,k\}$, then the sets $\delta'(S,a), \delta'(S,b)$, and $\delta'(S,c)$ are also some subsets of the set $\{1,2,\ldots,k\}$. If S is a set in $\mathcal{R}_{2,r}$ (or in $\mathcal{R}_{1,r}$) for some r, then the sets $\delta'(S,a)$ and $\delta'(S,b)$ are also in $\mathcal{R}_{2,r}$ (in $\mathcal{R}_{1,r}$, respectively), and the set $\delta'(S,c)$ is a subset of the set $\{1,2,\ldots,k\}$.

In the case of the symbol d, it is important to notice that $\delta(1,d) \subset \delta(2,d) \subset \ldots \subset \delta(\ell-1,d) \subset \delta(\ell,d)$, and in the other states, transitions on d are not defined. It follows that $\delta'(S,d)$ is either the empty set or is equal to $\delta(j,d)$ for the greatest integer j in $\{1,2,\ldots,\ell\}$ that is in S.

We have shown that for each set S in the system $\mathcal{S}_2$, the sets $\delta'(S,a), \delta'(S,b)$, $\delta'(S,c)$, and $\delta'(S,d)$ are again in the system $\mathcal{S}_2$. This means that $\mathcal{S}_2$ is the system of all reachable states of the DFA C'.

By Lemma 4, the size of the set $\mathcal{R}_{2,k-k_i+1} \setminus \{\{0,1,2,3,\ldots,k\}\}$ is $2^{k_i}-1$ for $i = 1,2,\ldots,\ell-1$, the size of the set $\mathcal{R}_{1,k-k_\ell+1}$ is $2^{k_\ell}-1$, and these sets are pairwise disjoint accept for they all contain state $\{0,1,2,3,\ldots,k\}$. It follows that

$$\begin{aligned} |\mathcal{S}_2| &= n-(k+1)+2^k+(2^{k_1}-1)+\cdots+(2^{k_{\ell-1}}-1)+(2^{k_\ell}-1) = \\ & \quad n-(k+1)+2^k+m=\alpha. \end{aligned}$$

Hence the DFA C' has exactly α reachable states, which proves *(ii)*.

To prove *(iii)* let S and T be two different reachable states of the DFA C'. If both S and T are some subsets of the set $\{0,1,2,\ldots,k\}$, then, without loss of generality, there is a state j in $\{0,1,2,\ldots,k\}$ such that $j \in S$ and $j \notin T$. Then the string c^{k-j} is accepted by the DFA C' from state S but not from state T. If S is a subset of $\{0,1,2,\ldots,k\}$ and T is one of the singletons $\{n-1\}, \{n-2\}, \ldots, \{k+1\}$, then a string from c^* distinguishes S and T if $S \neq \varnothing$, and a string from b^+c^{k-1} distinguishes them otherwise. If $S = \{i\}$ and $T = \{j\}$, where $k+1 \leqslant i < j \leqslant n-1$, then the string $b^{i-k}c^{k-1}$ is accepted by the DFA C' from state S but not from state T. This completes our proof. □

4.2 Union and Intersection

In this section we study the magic numbers problem for deterministic and nondeterministic state complexity of union and intersection of two regular languages.

4.2.1 Union and Intersection and State Complexity

We first investigate the state complexity of languages resulting from the union or intersection of two DFA languages. It is known [35, 44] that the union of an m-state DFA language and an n-state DFA language can be accepted by an mn-state DFA. This upper bound can be reached by the union of two binary languages [35]. The same result for intersection follows from de Morgan's law and the fact that the state complexity of a regular language is the same as the state complexity of its complement.

We show that for all integers m, n, and α such that $m \geqslant 2, n \geqslant 2$, and $1 \leqslant \alpha \leqslant mn$, there exist an m-state DFA language and an n-state DFA language such that the minimal DFA for the union of these languages has exactly α states. The same result for intersection then follows immediately. In the case of $m = 1$, the language accepted by a 1-state DFA is either empty or equals Σ^*, and so we get the following result.

Proposition 1. *The state complexity of the union of a 1-state DFA language and an n-state DFA language is either 1 or n.* □

In the following two lemmata, we assume $2 \leqslant m \leqslant n$. The first lemma shows that all values of α between 1 and m can be reached by the union of an m-state DFA language and an n-state DFA language. The second lemma shows the same result for the values between $m+1$ and $m+n-2$. To prove the results we use a unary alphabet in Lemma 5 and a binary alphabet in Lemma 6.

Lemma 5. *For all integers m, n, α such that $m \geqslant 2$ and $1 \leqslant \alpha \leqslant m \leqslant n$, there exist a minimal DFA A of m states and a minimal DFA B of n states such that the minimal DFA for the language $L(A) \cup L(B)$ has α states.*

Proof. Let m, n, and α be arbitrary but fixed integers with $1 \leqslant \alpha \leqslant m \leqslant n$. Let $\Sigma = \{a\}$.

Let $A = (\{q_0, q_1, \ldots, q_{m-1}\}, \Sigma, \delta_A, q_0, \{q_{m-1}\})$, where $\delta(q_i, a) = q_{i+1}$ for $i = 0, 1, \ldots, m-2$, and $\delta(q_{m-1}, a) = q_{m-1}$. The DFA A accepts the language $\{a^i \mid i \geqslant m-1\}$ and is minimal since it does not contain equivalent states. In the case of $\alpha = m = n$, we set $B = A$ and then the lemma follows. Otherwise, let $B = (\{p_0, p_1, \ldots, p_{n-1}\}, \Sigma, \delta_B, p_0, \{p_{\alpha-1}, \ldots, p_{n-2}\})$, where $\delta_B(p_i, a) = p_{i+1}$ for $i = 0, 1, \ldots, n-2$, and $\delta_B(p_{n-1}, a) = p_{n-1}$. The DFA B is the minimal DFA for the language $\{a^i \mid \alpha - 1 \leqslant i \leqslant n-2\}$. The union of the languages accepted by the DFAs A and B is the language $L(A) \cup L(B) = \{a^i \mid i \geqslant \alpha - 1\}$. The minimal DFA for this language has α states. □

Lemma 6. *For all integers m, n, α such that $2 \leqslant m \leqslant n$ and $m+1 \leqslant \alpha \leqslant m+n-2$, there exist a minimal DFA A of m states and a minimal DFA B of n states such that the minimal DFA for the language $L(A) \cup L(B)$ has α states.*

Proof. Let m, n, and α be arbitrary but fixed integers such that $2 \leqslant m \leqslant n$ and $m+1 \leqslant \alpha \leqslant m+n-2$. Then α can be expressed as $\alpha = m+k$ for some integer k with $1 \leqslant k \leqslant n-2$. Let $\Sigma = \{a, b\}$.

Let B be the minimal n-state DFA over Σ for the language $\{b^i \mid i \geqslant k\} \cup \{a^{n-k-2}\}$. To define the DFA A we consider two cases:

(i) $m \leqslant n-k$. Let A be the minimal m-state DFA over the alphabet Σ for the language $\{a^i \mid i \geqslant m-2\}$. Since $m \leqslant n-k$, the union of the languages accepted by the DFAs A and B is the language $L(A) \cup L(B) = \{a^i \mid i \geqslant m-2\} \cup \{b^i \mid i \geqslant k\}$. The minimal DFA for this language has $m+k$ states.

(ii) $m > n-k$. Let A be the minimal m-state DFA over the alphabet Σ for the language $\{a^{m-2}\}$. Since $m > n-k$, the union of the languages accepted by the DFAs A and B is the language $L(A) \cup L(B) = \{a^{n-k-2}, a^{m-2}\} \cup \{b^i \mid i \geqslant k\}$. The minimal DFA for this language has $m+k$ states. □

The next lemma shows that all the values between $m+n-1$ and mn can be reached by the union of an m-state DFA language and an n-state DFA language over a ternary alphabet.

Lemma 7. *For all integers m, n, α with $m \geqslant 2, n \geqslant 2$, and $m+n-1 \leqslant \alpha \leqslant mn$, there exist a minimal DFA A of m states and a minimal DFA B of n states such that the minimal DFA for the language $L(A) \cup L(B)$ has α states.*

Proof. Let m, n, and α be arbitrary but fixed integers such that $m \geqslant 2, n \geqslant 2$, and $m+n-1 \leqslant \alpha \leqslant mn$. Then α can be expressed as $\alpha = m + s(n-1) + t$ for some integers s and t such that $1 \leqslant s \leqslant m-1$ and $0 \leqslant t \leqslant n-1$. Let $\Sigma = \{a, b, c\}$.

Define an m-state DFA $A = (Q_A, \Sigma, \delta_A, q_0, F_A)$, where $Q_A = \{q_0, \ldots, q_{m-1}\}$, $F_A = \{q_{m-1}\}$, and for each $i \in \{0, 1, \ldots, m-1\}$ and each $X \in \Sigma$,

$$\delta_A(q_i, X) = \begin{cases} q_{(i+1) \bmod m}, & \text{if } X = a, \\ q_i, & \text{if } i < s \text{ and } X = b, \\ q_0, & \text{if } i \geqslant s \text{ and } X = b, \\ q_i, & \text{if } i = s \text{ and } X = c, \\ q_0, & \text{if } i \neq s \text{ and } X = c. \end{cases}$$

Define an n-state DFA $B = (Q_B, \Sigma, \delta_B, p_0, F_B)$, where $Q_B = \{p_0, \ldots, p_{n-1}\}$, $F_B = \{p_{n-1}\}$, and for each $i \in \{0, 1, \ldots, n-1\}$ and each $X \in \Sigma$,

$$\delta_B(p_i, X) = \begin{cases} p_0, & \text{if } X = a, \\ p_{(i+1) \bmod n}, & \text{if } X = b, \\ p_{i+1}, & \text{if } i < t \text{ and } X = c, \\ p_0, & \text{if } i \geqslant t \text{ and } X = c. \end{cases}$$

The DFA A and B are shown in Fig. 4.8.

Both automata are minimal since the string a^{m-1-i} distinguishes states q_i and q_j with $i \neq j$, and the string b^{n-1-i} distinguishes states p_i and p_j with $i \neq j$.

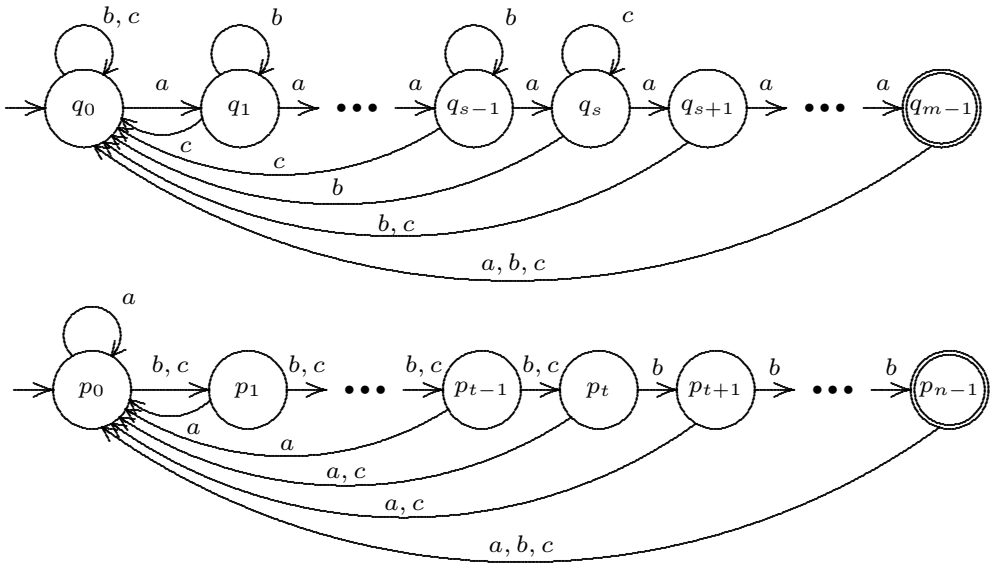

Figure 4.8: The deterministic finite automata A and B.

Let $C = (Q_A \times Q_B, \Sigma, \delta, [q_0, p_0], F)$, where $F = \{[q, p] \in Q_A \times Q_B \mid q \in F_A \text{ or } p \in F_B\}$, be the cross-product of automata A and B accepting the language $L(A) \cup L(B)$. We are going to prove that the DFA C has exactly α reachable states no two of which are equivalent. Let $\mathcal{R}$ be the following set of states of the DFA C

$$\mathcal{R} = \{[q_i, p_0] \mid 0 \leqslant i < m\} \cup \{[q_i, p_j] \mid 0 \leqslant i < s, 1 \leqslant j < n\} \cup \{[q_s, p_j] \mid 1 \leqslant j \leqslant t\}.$$

There are $m + s(n-1) + t$ states in the set $\mathcal{R}$. Any state in $\mathcal{R}$ is reachable in the DFA C because we have:

$[q_i, p_0] = \delta([q_0, p_0], a^i)$ for $i = 0, 1, \ldots, m-1$,
$[q_i, p_j] = \delta([q_0, p_0], a^i b^j)$ for $i = 0, 1, \ldots, s-1$ and $j = 1, 2, \ldots, n-1$, and
$[q_s, p_j] = \delta([q_0, p_0], a^s c^j)$ for $j = 1, 2, \ldots, t$.

To prove that no other state is reachable in the DFA C note that the initial state $[q_0, p_0]$ of the DFA C is in $\mathcal{R}$ and for each state $[q, p]$ in $\mathcal{R}$ and each symbol X in Σ the state $\delta([q, p], X)$ is in $\mathcal{R}$ as well.

To prove that no two different states in $\mathcal{R}$ are equivalent let $[q_i, p_j]$ and $[q_k, p_l]$ be two different states in $\mathcal{R}$. Then, either $i \neq k$ or $j \neq l$. In the first case, these states are distinguished by the string a^{m-1-i}, and in the second case by the string b^{n-1-j}. □

To prove the result in the lemma above we used a three-letter alphabet. The next lemma shows that the same result holds for a binary alphabet as well.

Lemma 8. *For any integers m, n, α with $m \geqslant 2$, $n \geqslant 2$, and $m+n-1 \leqslant \alpha \leqslant mn$, there exist a minimal binary DFA A of m states and a minimal binary DFA B of n states such that the minimal DFA for the language $L(A) \cup L(B)$ has α states.*

Proof. Let m, n, and α be arbitrary but fixed integers such that $m \geqslant 2$, $n \geqslant 2$, and $m + n - 1 \leqslant \alpha \leqslant mn$. Let $\Sigma = \{a, b\}$.

First, let α can be expressed as $\alpha = rm + sn - rs$ for some integers r and s such that $1 \leqslant r \leqslant n-1$ and $1 \leqslant s \leqslant m-1$, that is, in the cross-product automaton, we want to reach the first s rows and the first r columns.

Define an m-state DFA $A_{rs} = (\{q_0, q_1, \ldots, q_{m-1}\}, \Sigma, \delta_A^{(rs)}, q_0, \{q_{m-1}\})$, where for each $i \in \{0, 1, \ldots, m-1\}$ and each $X \in \Sigma$,

$$\delta_A^{(rs)}(q_i, X) = \begin{cases} q_{(i+1) \bmod m}, & \text{if } X = a, \\ q_i, & \text{if } i < s \text{ and } X = b, \\ q_0, & \text{if } i \geqslant s \text{ and } X = b. \end{cases}$$

Define an n-state DFA $B_{rs} = (\{p_0, p_1, \ldots, p_{n-1}\}, \Sigma, \delta_B^{(rs)}, p_0, \{p_{n-1}\})$, where for each $i \in \{0, 1, \ldots, n-1\}$ and each $X \in \Sigma$,

$$\delta_B^{(rs)}(p_i, X) = \begin{cases} p_{(i+1) \bmod n}, & \text{if } X = b, \\ p_i, & \text{if } i < r \text{ and } X = a, \\ p_0, & \text{if } i \geqslant r \text{ and } X = a. \end{cases}$$

Let $C_{rs} = (Q, \Sigma, \delta^{(rs)}, [q_0, p_0], F)$ be the cross-product of automata A_{rs} and B_{rs} accepting the language $L(A) \cup L(B)$. We are going to prove that

the DFA C_{rs} has exactly α reachable states no two of which are equivalent. Let $\mathcal{R}_{rs}$ be the following set of states:

$$\mathcal{R}_{rs} = \{[q_i, p_j] \mid (0 \leqslant i < s \text{ and } 0 \leqslant j < n) \text{ or } (0 \leqslant j < r \text{ and } 0 \leqslant i < m)\}.$$

There are $sn + rm - rs$ states in the set $\mathcal{R}_{rs}$. Any state in $\mathcal{R}_{rs}$ is reachable in the DFA C_{rs} because we have $[q_i, p_j] = \delta([q_0, p_0], a^i b^j)$ if $i \leqslant s - 1$, and $[q_i, p_j] = \delta([q_0, p_0], b^j a^i)$ if $j \leqslant r - 1$. To prove that no other state is reachable in the DFA C_{rs} note that the initial state $[q_0, p_0]$ of the DFA C_{rs} is in $\mathcal{R}_{rs}$ and for any state $[q, p]$ in $\mathcal{R}_{rs}$ and any symbol X in Σ the state $\delta([q, p], X)$ is in $\mathcal{R}_{rs}$ as well. To prove that no two different states in $\mathcal{R}_{rs}$ are equivalent let $[q_i, p_j]$ and $[q_k, p_l]$ be two different states in $\mathcal{R}_{rs}$. The string a^{m-1-i} distinguishes states $[q_i, p_j]$ and $[q_k, p_l]$ if $i \neq k$ and the string b^{n-1-j} distinguishes these states if $j \neq l$.

Now, let r and s be an arbitrary integers such that $r < n$ and $s < m$. Let t be an integer such that $1 \leqslant t \leqslant \min\{m - s, n - r\} - 1$. We are going to define binary automata A and B such that the minimal DFA for the language $L(A) \cup L(B)$ has $rm + sn - rs + t$ states.

Define an m-state DFA $A = (\{q_0, q_1, \ldots, q_{m-1}\}, \Sigma, \delta_A, q_0, \{q_{m-1}\})$, where
$\delta_A(q_{s+i}, b) = q_{s+i-1}$, if $i \leqslant t$ and i is odd, and
$\delta_A(q_i, X) = \delta_A^{(rs)}(q_i, X)$, otherwise.

Define an n-state DFA $B = (\{p_0, p_1, \ldots, p_{n-1}\}, \Sigma, \delta_B, p_0, \{p_{n-1}\})$, where
$\delta_B(p_{r+i}, a) = p_{r+i-1}$, if $i \leqslant t$ and i is even, and
$\delta_B(p_i, X) = \delta_B^{(rs)}(p_i, X)$, otherwise.

Then, the set of reachable and pairwise inequivalent states of the cross-product automaton for the language $L(A) \cup L(B)$ is

$$\mathcal{R}_{rs} \cup \{[q_s, p_{r+j}] \mid j \leqslant t \text{ and } j \text{ is even}\} \cup \{[q_{s+i}, p_r] \mid i \leqslant t \text{ and } i \text{ is odd}\};$$

no other states are reached since each new state goes to one of these states by a and b, and inequivalence is proved in the same way as above.

Finally, note that if $t = \min\{m - s, n - r\}$, say $t = m - s$, then $rm + sn - rs + t = (r + 1)m + sn - (r + 1)s$, i.e., $rm + sn - rs + t$ is the size of the minimal DFA for the language $L(A_{r+1s}) \cup L(B_{r+1s})$. The automaton C_{rs} with $s = m - 1$ and $r = n - 1$ has $mn - 1$ states. This completes our proof since mn is the upper bound on the state complexity of intersection, which is tight in the binary case [44]. □

As a corollary of the four lemmata above, we get the following result.

Theorem 4. *For all integers m, n, α such that $m \geqslant 2$, $n \geqslant 2$, and $1 \leqslant \alpha \leqslant mn$, there exist a minimal binary DFA A of m states and a minimal binary DFA B of n states such that the minimal DFA for the language $L(A) \cup L(B)$ has α states.* □

The same result for intersection follows from de Morgan's law and the fact that the state complexity of a regular language equals the state complexity of its complement.

Theorem 5. *For all integers m, n, α such that $m \geqslant 2$, $n \geqslant 2$, and $1 \leqslant \alpha \leqslant mn$, there exist a minimal binary DFA A of m states and a minimal binary DFA B of n states such that the minimal DFA for the language $L(A) \cap L(B)$ has α states.* □

4.2.2 Union and Nondeterministic State Complexity

We now turn our attention to the nondeterministic state complexity of languages resulting from the union of two NFA languages. It is known [18] that the union of an m-state NFA language and an n-state NFA language can be accepted by an $(m + n + 1)$-state NFA and this upper bound is tight for a binary alphabet.

In this section, we show that the nondeterministic complexity of the union of an m-state NFA language and an n-state NFA language may be arbitrary between 1 and $m + n + 1$ except for the case of $m = 1$ and $n = 1$. To prove the result we use a fooling-set lower-bound technique.

We start our investigations with the following lemma.

Lemma 9. *For all integers m, n, α such that $1 \leqslant \alpha \leqslant m \leqslant n$, there exist a minimal NFA A of m states and a minimal NFA B of n states such that every minimal NFA for the language $L(A) \cup L(B)$ has α states.*

Proof. Let m, n, and α be arbitrary but fixed integers such that $1 \leqslant \alpha \leqslant m \leqslant n$. Let $\Sigma = \{a\}$. Let A be a minimal m-state NFA for the language $\{a^i \mid i \geqslant m - 1\}$ and let B be a minimal n-state NFA for the language $\{a^i \mid \alpha - 1 \leqslant i \leqslant n - 1\}$; note that the set $\{(a^{i-1}, a^{m-i}) \mid i = 1, 2, \ldots, m\}$ is a fooling set for the language $L(A)$ and the set $\{(a^{i-1}, a^{n-i}) \mid i = 1, 2, \ldots, n\}$ is a fooling set for the language $L(B)$. The union of the languages accepted by the NFAs A and B is the language $L(A) \cup L(B) = \{a^i \mid i \geqslant \alpha - 1\}$. Any minimal NFA for this language has α states. □

The next two lemmata deal with the cases of $m = 1$ and $m = 2$. Note that the nondeterministic state complexity of the union of two 1-state NFA languages is 1 or 3.

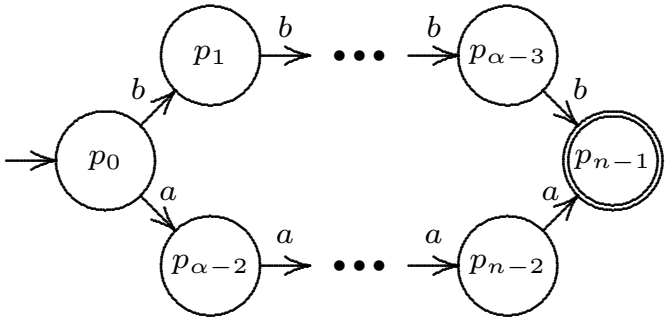

Figure 4.9: The nondeterministic finite automaton B.

Lemma 10. *For all integers n, α such that $n \geqslant 2$ and $2 \leqslant \alpha \leqslant n+1$, there exist a 1-state NFA A and a minimal NFA B of n states such that every minimal NFA for the language $L(A) \cup L(B)$ has α states.*

Proof. Let n be an arbitrary but fixed integer with $n \geqslant 2$. Let A be a 1-state NFA for the language $\{a^i \mid i \geqslant 0\}$.

If $\alpha = 2$, let B be a minimal n-state NFA for the language $\{a^{n-2}\} \cup \{ba^i \mid i \geqslant 0\}$; note that the set of pairs of strings $\{(a^i, a^{n-2-i}) \mid i = 0, 1, \ldots, n-2\} \cup \{(b, a^n)\}$ is a fooling set for the language $L(B)$. Then, $L(A) \cup L(B) = \{a^i \mid i \geqslant 0\} \cup \{ba^i \mid i \geqslant 0\}$, for which any minimal NFA has 2 states since the set $\{(\varepsilon, b), (a, a)\}$ is a fooling set for this language.

If $3 \leqslant \alpha \leqslant n+1$, let B be a minimal n-state NFA for the language $\{b^{\alpha-2}\} \cup \{a^{n-\alpha+2}\}$, see Fig. 4.9. Then, $L(A) \cup L(B) = \{a^i \mid i \geqslant 0\} \cup \{b^{\alpha-2}\}$, any minimal NFA for which has α states since the set $\{(a, a)\} \cup \{(b^i, b^{\alpha-2-i}) \mid i = 0, 1, \ldots, \alpha-2\}$ is a fooling set for this language. □

Lemma 11. *For all integers n, α such that $n \geqslant 2$ and $3 \leqslant \alpha \leqslant n+1$, there exist a minimal 2-state NFA A and a minimal NFA B of n states such that every minimal NFA for the language $L(A) \cup L(B)$ has α states.*

Proof. Let n and α be arbitrary but fixed integers such that $n \geqslant 2$ and $3 \leqslant \alpha \leqslant n+1$. Let A be a minimal 2-state NFA for the language $\{a^i \mid i > 0\}$. Let B be a minimal n-state NFA for the language $\{b^{\alpha-2}\} \cup \{a^{n-\alpha+2}\}$. Then, $L(A) \cup L(B) = \{a^i \mid i > 0\} \cup \{b^{\alpha-2}\}$. Any minimal NFA for this language has α states. □

In the next lemma, we assume m and n to be at least 3 and show that all values between $m+1$ and $m+n-2$ can be reached by the union of appropriate NFAs.

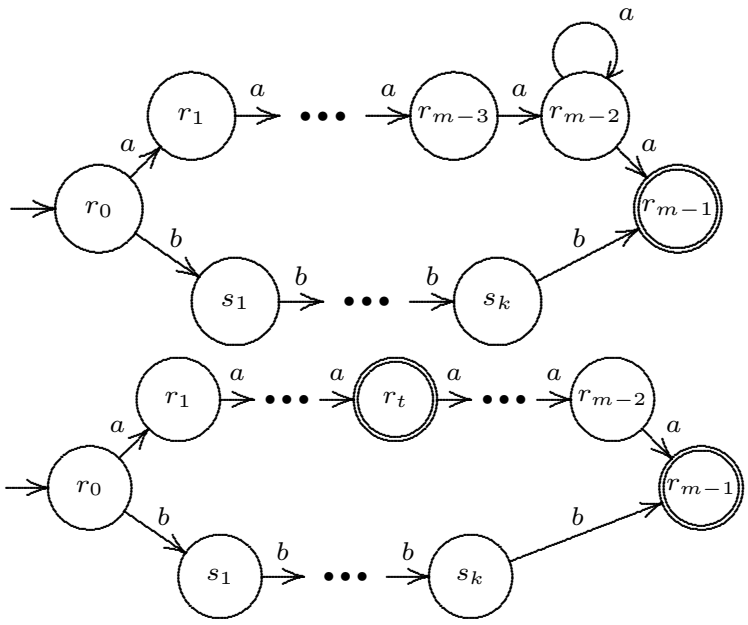

Figure 4.10: The nondeterministic finite automata C and C'; $t = n - k - 1$

Lemma 12. *For all integers m, n, α such that $3 \leqslant m \leqslant n$ and $m+1 \leqslant \alpha \leqslant m+n-2$, there exist a minimal NFA A of m states and a minimal NFA B of n states such that every minimal NFA for the language $L(A) \cup L(B)$ has α states.*

Proof. Let m, n, and α be arbitrary but fixed integers such that $3 \leqslant m \leqslant n$ and $m+1 \leqslant \alpha \leqslant m+n-2$. Then α can be expressed as $\alpha = m+k$ for some integer k with $1 \leqslant k \leqslant n-2$. Let $\Sigma = \{a, b\}$. Let B be a minimal n-state NFA for the language $\{a^{n-k-1}, b^{k+1}\}$. To define the DFA A we consider two cases:

(i) $m \leqslant n-k$. Let A be a minimal m-state NFA for the language $\{a^i \mid i \geqslant m-1\}$. The union of the languages accepted by the NFAs A and B is the language $\{a^i \mid i \geqslant m-1\} \cup \{b^{k+1}\}$ which is accepted by an $(m+k)$-state NFA C shown in Fig. 4.10 (top). The NFA C is minimal since the set $\{(a^{i-1}, a^{m-i}) \mid i = 1, 2, \ldots, m\} \cup \{(b^i, b^{k+1-i}) \mid i = 1, 2, \ldots, k\}$ is a fooling set for the language $L(A) \cup L(B)$.

(ii) $m > n-k$. Let A be a minimal m-state NFA for the language $\{a^{m-1}\}$. The union of languages accepted by the NFAs A and B is the language $\{a^{n-k-1}, a^{m-1}, b^{k+1}\}$ which is accepted by an $(m+k)$-state NFA C' shown in Fig. 4.10 (bottom). The NFA C' is minimal since the set

of pairs of strings $\{(a^{i-1}, a^{m-i}) \mid i = 1, 2, \ldots, m\} \cup \{(b^i, b^{k+1-i}) \mid i = 1, 2, \ldots, k\}$ is a fooling set for the language $L(A) \cup L(B)$. □

The next two lemmata deal with the cases of $\alpha = m+n-1$ and $\alpha = m+n$.

Lemma 13. *For all integers m, n such that $m \geqslant 2$ and $n \geqslant 2$, there exist a minimal NFA A of m states and a minimal NFA B of n states such that every minimal NFA for the language $L(A) \cup L(B)$ has $m + n - 1$ states.*

Proof. Let m and n be arbitrary but fixed integers such that $m \geqslant 2$, $n \geqslant 2$, and let $\Sigma = \{a, b\}$. Let A be an m-state NFA shown in Fig. 4.11 (top) and let B be an n-state NFA shown in Fig. 4.11 (bottom). Both automata are minimal since the set of pairs of strings $\{(a^{i-1}, a^{m-i}) \mid i = 1, 2, \ldots, m\}$ is a fooling set for the language $L(A)$ and the set $\{(b^{i-1}, b^{n-i}) \mid i = 1, 2, \ldots, n\}$ is a fooling set for the language $L(B)$.

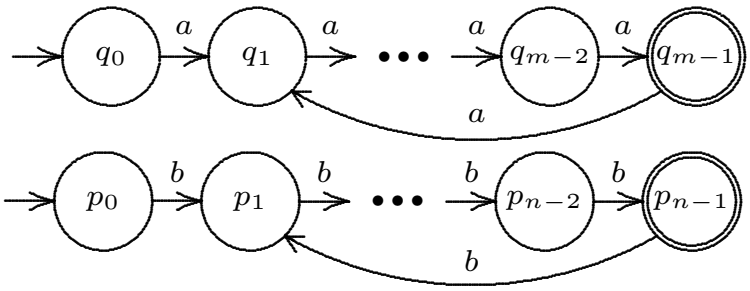

Figure 4.11: The nondeterministic finite automata A and B.

The language $L(A) \cup L(B)$ is accepted by an $(m + n - 1)$-state NFA (a cycle on a from q_1 through q_{m-1} to q_1, a cycle on b from p_1 through p_{n-1} to p_1, and a new initial state q_I going to q_1 by a and to p_1 by b). Since the set of pairs of strings $\{(a^i, a^{m-i-1}) \mid i = 0, 1, \ldots, m-1\} \cup \{(b^i, b^{n-1-i}) \mid i = 1, 2, \ldots, n-2\} \cup \{(b^{n-1}, b^{n-1})\}$ is a fooling set for the language $L(A) \cup L(B)$, any minimal NFA for this language has $m + n - 1$ states. □

Lemma 14. *For all positive integers m, n such that $n \geqslant 2$, there exist a minimal NFA A of m states and a minimal NFA B of n states such that every minimal NFA for the language $L(A) \cup L(B)$ has $m + n$ states.*

Proof. Let m and n be arbitrary but fixed integers with $m \geqslant 1$, $n \geqslant 2$, and let $\Sigma = \{a, b\}$. Let A be a minimal m-state NFA for the language $\{a^m\}^*$; note that the set $\{(a^i, a^{m-i}) \mid i = 0, 1, \ldots, m-1\}$ is a fooling set for this language. Let B be a minimal n-state NFA for the language $\{b^{n-1}\}$. Then the union of the languages accepted by the NFAs A and B is accepted by an

$(m+n)$-state NFA. Since the set of pairs $\{(a^i, a^{m-i}) \mid i = 1, 2, \ldots, m-1\} \cup \{(a^m, a^m)\} \cup \{(b^i, b^{n-1-i}) \mid i = 0, 1, \ldots, n-1\}$ is a fooling set for the language $L(A) \cup L(B)$, any minimal NFA for this language has $m+n$ states. □

The following result is proved in [18].

Lemma 15. *For any positive integers m and n, any minimal NFA for the union of languages $\{a^m\}^*$ and $\{b^n\}^*$ has $m+n+1$ states.* □

As a corollary of the lemmata above, we get the following result.

Theorem 6. *For any integers m, n, α such that $m \geqslant 2$ or $n \geqslant 2$, and $1 \leqslant \alpha \leqslant m+n+1$, there exist a minimal binary NFA A of m states and a minimal binary NFA B of n states such that any minimal NFA for the language $L(A) \cup L(B)$ has α states.* □

4.2.3 Intersection and Nondeterministic Complexity

In this section, we study the nondeterministic state complexity of languages resulting from the intersection of two NFA languages. The intersection of an m-state NFA language and an n-state NFA language can be accepted by an mn-state NFA and this upper bound is known to be tight for a binary alphabet [18]. We show that the nondeterministic state complexity of the intersection of an m-state NFA language and an n-state NFA language may be arbitrary between 1 and mn. We prove the result for a ternary alphabet.

Lemma 16. *For all integers m, n, α such that $1 \leqslant \alpha \leqslant m \leqslant n$, there exist a minimal NFA A of m states and a minimal NFA B of n states such that every minimal NFA for the language $L(A) \cap L(B)$ has α states.*

Proof. Let m, n, and α be arbitrary but fixed integers such that $1 \leqslant \alpha \leqslant m \leqslant n$ and let A and B be m-state and n-state NFAs accepting languages $\{a, b\}^{\alpha-1} \cup \{a^{m-1}\}$ and $\{a, b\}^{\alpha-1} \cup \{b^{n-1}\}$, respectively. Both NFAs are minimal since the sets of pairs of strings $\{(a^{i-1}, a^{m-i}) \mid i = 1, 2, \ldots, m\}$ and $\{(b^{i-1}, b^{n-i}) \mid i = 1, 2, \ldots, n\}$ are fooling sets for the languages $L(A)$ and $L(B)$, respectively. The intersection of these languages is the language $\{a, b\}^{\alpha-1}$ consisting of all strings over the alphabet $\{a, b\}$ of length $\alpha - 1$. Any minimal NFA for this language has α states. □

Lemma 17. *For all positive integers m, n, α such that $m \leqslant n$ and $m \leqslant \alpha \leqslant mn$, there exist a minimal NFA A of m states and a minimal NFA B of n states such that every minimal NFA for the language $L(A) \cap L(B)$ has α states.*

Proof. Let m, n, and α be arbitrary but fixed positive integers such that $m \leqslant n$ and $m \leqslant \alpha \leqslant mn$. Then α can be expressed as $\alpha = m + s(n-1) + t$, where $0 \leqslant s \leqslant m$ and $0 \leqslant t < n-1$. Let $\Sigma = \{a, b, c\}$.

Define an m-state NFA $A = (Q_A, \Sigma, \delta_A, q_0, F_A)$, where $Q_A = \{q_0, q_1, \ldots, q_{m-1}\}$, $F_A = \{q_0\}$ and for each $i \in \{0, 1, \ldots, m-1\}$ and each $X \in \Sigma$,

$$\delta_A(q_i, X) = \begin{cases} \{q_{(i+1) \bmod m}\}, & \text{if } X = a, \\ \{q_i\}, & \text{if } i \leqslant s-1 \text{ and } X = b, \\ \{q_i\}, & \text{if } i = s \text{ and } X = c, \\ \varnothing, & \text{otherwise.} \end{cases}$$

Define an n-state DFA $B = (Q_B, \Sigma, \delta_B, p_0, F_B)$, where $Q_B = \{p_0, p_1, \ldots, p_{n-1}\}$, $F_B = \{p_0\}$, and for each $i \in \{0, 1, \ldots, n-1\}$ and each $X \in \Sigma$,

$$\delta_B(p_i, X) = \begin{cases} \{p_0\}, & \text{if } i = 0 \text{ and } X = a, \\ \{p_{(i+1) \bmod n}\}, & \text{if } X = b, \\ \{p_{i+1}\}, & \text{if } i \leqslant t-1 \text{ and } X = c, \\ \{p_0\}, & \text{if } i = t \text{ and } X = c, \\ \varnothing, & \text{otherwise.} \end{cases}$$

The NFA A and B are shown in Fig. 4.12. Both automata are minimal since the sets of pairs of strings $\{(a^i, a^{m-i}) \mid i = 0, 1, \ldots, m-1\}$ and $\{(b^i, b^{n-i}) \mid i = 0, 1, \ldots, n-1\}$ are fooling sets for the languages $L(A)$ and $L(B)$, respectively.

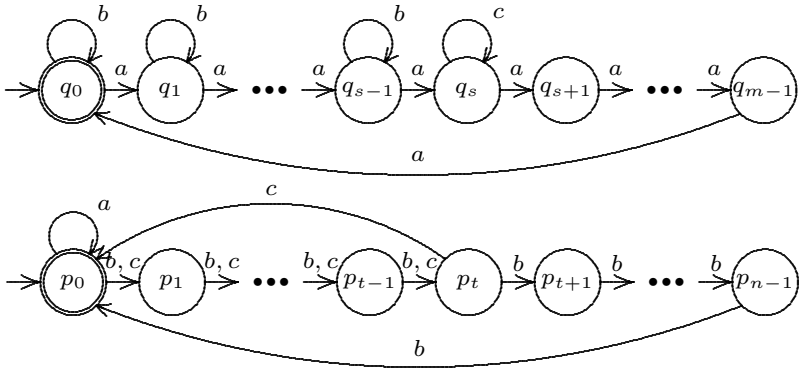

Figure 4.12: The nondeterministic finite automata A and B.

Let $C = (Q_A \times Q_B, \Sigma, \delta, [q_0, p_0], F)$, where $\delta([q, p], X) = [\delta_A(q, X), \delta_B(p, X)]$ for any $X \in \Sigma$, and $F = F_A \times F_B$, be the cross-product of automata A and B accepting the language $L(A) \cap L(B)$. Let $\mathcal{R}$ be the following set of states.

$\mathcal{R} = \{[q_i, p_0] \mid 0 \leqslant i < m\} \cup \{[q_i, p_j] \mid 0 \leqslant i < s, 1 \leqslant j < n\} \cup \{[q_s, p_j] \mid 1 \leqslant j \leqslant t\}$. Each state in $\mathcal{R}$ is reachable in the NFA C because we have

$[q_i, p_0] \in \delta([q_0, p_0], a^i)$ for $i = 0, 1, \ldots, m-1$, $[q_i, p_j] \in \delta([q_0, p_0], a^i b^j)$ for $i = 0, 1, \ldots, s-1$ and $j = 1, 2, \ldots, n-1$, and $[q_s, p_j] \in \delta([q_0, p_0], a^s c^j)$ for $j = 1, 2, \ldots, t$. Next, no other state in $Q_A \times Q_B$ is reachable in the NFA C because the initial state $[q_0, p_0]$ of the NFA C is in $\mathcal{R}$ and for any state $[q, p]$ in $\mathcal{R}$ and any symbol X in Σ either $\delta([q, p], X) \subseteq \mathcal{R}$ or $\delta([q, p], X) = \varnothing$. Thus the language $L(A) \cap L(B)$ is accepted by an $(m + s(n-1) + t)$-state NFA. To prove the lemma consider the following sets of pairs of strings:

$\mathcal{A} = \{(a^i, a^{m-i}) \mid i = 0, 1, \ldots, m-1\}$,
$\mathcal{B}_k = \{(a^k b^i, b^{n-i} a^{m-k}) \mid i = 1, 2, \ldots, n-1\}$, for $k = 0, 1, \ldots, s-1$,
$\mathcal{C} = \{(a^s c^i, c^{t+1-i} a^{m-s}) \mid i = 1, 2, \ldots, t\}$.

Let $\mathcal{D} = \mathcal{A} \cup \mathcal{B}_0 \cup \mathcal{B}_1 \cup \cdots \cup \mathcal{B}_{s-1} \cup \mathcal{C}$. We will show that the set $\mathcal{D}$ is a fooling set for the language $L(A) \cap L(B)$. We need to show that
(1) for any pair (x_i, y_i) in $\mathcal{D}$, the string $x_i y_i$ is in the language $L(A) \cap L(B)$, and (2) for any two different pairs (x_i, y_i) and (x_j, y_j) in $\mathcal{D}$, at least one of the strings $x_i y_j$ and $x_j y_i$ is not in the language $L(A) \cap L(B)$.

To prove (1) note that the strings a^m, $a^k b^n a^{m-k}$ $(k = 0, 1, \ldots, s-1)$, and $a^s c^{t+1} a^{m-s}$ are in the language $L(A) \cap L(B)$. To prove (2) we have six cases:

(i) Both pairs are in $\mathcal{A}$. Let $0 \leqslant i < j \leqslant m-1$. Then the string $a^i a^{m-j}$ is not in the language $L(A)$ since $0 < m - j + i < m$.

(ii) Both pairs are in $\mathcal{B}_k$ for some k $(0 \leqslant k \leqslant s-1)$. Let $1 \leqslant i < j \leqslant n-1$. Then the string $a^k b^i b^{n-j} a^{m-k}$ is not in the language $L(B)$ since $0 < n - j + i < n$.

(iii) Both pairs are in $\mathcal{C}$. Let $1 \leqslant i < j \leqslant t$. Then the string $a^s c^i c^{t+1-j} a^{m-s}$ is not in the language $L(B)$ since $0 < t + 1 - j + i < t + 1$.

(iv) The first pair is in $\mathcal{A}$ and the second in some $\mathcal{B}_k$ (or in $\mathcal{C}$). Then the string $a^k b^i a^{m-j}$ (the string $a^s c^i a^{m-j}$, respectively) is not in the language $L(B)$ since $1 \leqslant i \leqslant n-1$ ($1 \leqslant i \leqslant t$, respectively).

(v) The first pair is in some $\mathcal{B}_k$ and the second in some $\mathcal{B}_l$ with $0 \leqslant k < l \leqslant s-1$. Then the string $a^k b^i b^{n-j} a^{m-l}$ is not in the language $L(A)$ since $0 < k + m - l < m$.

(vi) The first pair is in some $\mathcal{B}_k$ and the second in $\mathcal{C}$. Then the string $a^s c^i b^{n-j} a^{m-k}$ is not in the language $L(A)$ since $i \geqslant 1$ and $n - j \geqslant 1$.

Thus we have shown that the set $\mathcal{D}$ is a fooling set for the language $L(A) \cap L(B)$. By Lemma 1, any NFA for the language $L(A) \cap L(B)$ needs at least $m + s(n-1) + t$ states and our proof is complete. □

As a corollary of the two lemmata above, we get the following result.

Theorem 7. *For any positive integers m, n, α such that $1 \leqslant \alpha \leqslant mn$, there exist a minimal NFA A of m states and a minimal NFA B of n states such that any minimal NFA for the language $L(A) \cap L(B)$ has α states.* □

4.3 Complementation

We now consider the complementation operation. For DFAs, it is an efficient operation since to accept the complement we can simply exchange accepting and rejecting states. On the other hand, the complementation of NFAs is an expensive task. The upper bound on the size of an NFA accepting the complement of an n-state NFA language is 2^n and it is known to be tight for a binary alphabet [28]. For complementation of unary NFA languages a crucial role is played by the function

$$F(n) = \max\{\mathrm{lcm}(x_1, \ldots, x_k) \mid x_1 + \ldots + x_k = n\}.$$

It is known that $F(n) \in \mathrm{e}^{\Theta(\sqrt{n \ln n})}$ and that $O(F(n))$ states suffice to simulate any unary n-state NFA by a DFA [8]. This means that $O(F(n))$ states are sufficient for a NFA to accept the complement of an n-state unary NFA language. The lower bound is known to be $F(n-1)+1$ in this case [28].

In this section, we deal with the question of which kind of relations between the nondeterministic complexity of a regular language and the nondeterministic complexity of its complement are possible. We provide a complete solution by showing that for all positive integers n, α with $\log n \leq \alpha \leq 2^n$, there exists an n-state NFA language such that a minimal NFA for its complement has exactly α states. To obtain the result we again use a fooling-set lower-bound technique. We start our investigation with two propositions.

Proposition 2. *For each α in $\{1, 2\}$, there is a 1-state NFA D_α such that every minimal NFA accepting the complement of $L(D_\alpha)$ has α states.*

Proof. Let $\Sigma = \{a, b\}$. Consider the following 1-state NFAs:
$D_1 = (\{s\}, \Sigma, \delta_1, s, \{s\})$ with $\delta_1(s, X) = \{s\}$ for any $X \in \Sigma$,
$D_2 = (\{s\}, \Sigma, \delta_2, s, \{s\})$ with $\delta_2(s, a) = \{s\}$ and $\delta_2(s, b) = \varnothing$.
The NFAs D_1 and D_2 do satisfy the proposition since the complement of the language $L(D_1)$ is the empty language, and the set of pairs of strings $\{(\varepsilon, b), (b, \varepsilon)\}$ is a fooling set for the complement of the language $L(D_2)$. □

Proposition 3. *For every integer $n \geq 2$ there is a minimal n-state NFA N such that every minimal NFA for the complement of $L(N)$ has n states.*

Proof. Let n be arbitrary but fixed integer with $n \geq 2$. Let $\Sigma = \{a, b\}$.

Define an n-state NFA $N = (Q, \Sigma, \delta, n, F)$, see Fig. 4.13, where $Q = \{1, 2, \ldots, n\}$, $F = \{2, 3, \ldots, n\}$, and for any $i \in Q$,

$\delta(1, a) = \delta(1, b) = \{2\}$,

$\delta(i, a) = \{i - 1\}$ and $\delta(i, b) = \{1\}$ if $i > 1$.

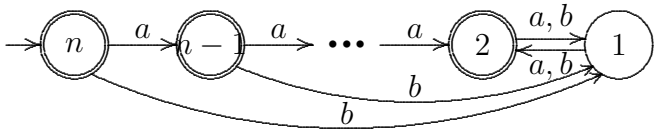

Figure 4.13: The nondeterministic finite automaton N.

We are going to show that *(i)* the NFA N is a minimal NFA for the language $L(N)$; *(ii)* the language $L^c(N)$ is accepted by an n-state DFA; *(iii)* any NFA for the language $L^c(N)$ needs at least n states. Then, the proposition follows.

Consider the set of pairs $\mathcal{A} = \{(a^{i-1}, a^{n-i}b) \mid i = 1, 2, \ldots, n\}$. The set $\mathcal{A}$ is a fooling set for the language $L(N)$ because for any i and j in $\{1, 2, \ldots, n\}$, (1) $a^{i-1}a^{n-i}b \in L(N)$ since the string $a^{n-1}b$ is accepted by the NFA N, and (2) if $i < j$, then $a^{i-1}a^{n-j}b \notin L(N)$ since any string $a^l b$ with $0 \leq l < n - 1$ is not accepted by the NFA N. By Lemma 1, any NFA for the language $L(N)$ needs at least n states which proves *(i)*.

To prove *(ii)* note that the NFA N is, in fact, deterministic, and so after exchanging the accepting and the rejecting states we obtain an n-state DFA for the language $L^c(N)$.

Finally, consider the set of pairs $\mathcal{B} = \{(a^{i-1}, a^{n-i}) \mid i = 1, 2, \ldots, n\}$. The set $\mathcal{B}$ is a fooling set for the language $L^c(N)$ because for any i and j in $\{1, 2, \ldots, n\}$, (1) $a^{i-1}a^{n-i} \in L^c(N)$ since the string a^{n-1} is not accepted by the NFA N, and (2) if $i < j$, then $a^{i-1}a^{n-j} \notin L^c(N)$ since any string a^l with $0 \leq l < n - 1$ is accepted by the NFA N. By Lemma 1, any NFA for the language $L^c(N)$ needs at least n states and our proof is complete. □

The following theorem is proved in [28].

Theorem 8 ([28])**.** *For every positive integer n, there exists a binary NFA M of n states such that every NFA accepting the complement of the language $L(M)$ needs at least 2^n states.* □

4.3.1 Exponential Alphabet

In the next theorem, we show that the nondeterministic state complexity of the complement of an n-state NFA language may be arbitrary between $n+1$ and 2^n-1. We first prove the result using an alphabet that grows exponentially with n.

Theorem 9. *For all positive integers n and α with $3 \leq n+1 \leq \alpha \leq 2^n-1$, there is a minimal NFA M of n states such that a minimal NFA accepting the complement of the language $L(M)$ has α states.*

Proof. Let n and α be arbitrary but fixed positive integers such that $3 \leq n+1 \leq \alpha \leq 2^n-1$. Then α can be expressed as $\alpha = n+k$ for an integer k with $1 \leq k \leq 2^n-1-n$. Let $\Sigma = \{a,b\} \cup \{c_1, c_2, \ldots, c_k\} \cup \{d_1, d_2, \ldots, d_k\}$ be a $(2k+2)$-letter alphabet. We are going to define a minimal n-state NFA M over the alphabet Σ such that a minimal NFA for the language $L^c(M)$ has $n+k$ states. To this aim let $S_1, S_2, \ldots, S_{2^n-1-n}$ be a sequence of subsets of the set $\{1,2,\ldots,n\}$ containing at least two elements and ordered in such a way that for any i and j in $\{1,2,\ldots,2^n-1-n\}$, the following two conditions hold:

(1) if $\max S_i < \max S_j$, then $i < j$;

(2) if $\max S_i = \max S_j$ and $1 \in S_i \setminus S_j$, then $i < j$,

i.e., the subsets are ordered according to their maxima, and if two sets have the same maximum, then all sets containing state 1 precede the sets not containing state 1. Clearly, there are several such orderings, we choose one of them. Note that $S_1 = \{1,2\}$. For example, the subsets of $\{1,2,3,4\}$ containing at least two elements could be ordered as follows: $S_1 = \{1,2\}, S_2 = \{1,3\}, S_3 = \{1,2,3\}, S_4 = \{2,3\}, S_5 = \{1,4\}, S_6 = \{1,2,4\}, S_7 = \{1,3,4\}, S_8 = \{1,2,3,4\}, S_9 = \{2,4\}, S_{10} = \{3,4\}, S_{11} = \{2,3,4\}$.

Define an n-state NFA $M = (Q, \Sigma, \delta, n, F)$, where $Q = \{1,2,\ldots,n\}$, $F = \{2,3,\ldots,n\}$, and for any $i \in Q$ and any $j \in \{1,2,\ldots,k\}$,

$$\delta(i,X) = \begin{cases} \{1,2\}, & \text{if } i=1 \text{ and } X=a, \\ \{i-1\}, & \text{if } i>1 \text{ and } X=a, \\ \{2\}, & \text{if } i=1 \text{ and } X=b, \\ \{1\}, & \text{if } i>1 \text{ and } X=b, \\ S_j, & \text{if } i=1 \text{ and } X=c_j, \\ \{1\}, & \text{if } i>1 \text{ and } X=c_j, \\ \{1\}, & \text{if } i \in S_j \text{ and } X=d_j, \\ \{2\}, & \text{if } i \notin S_j \text{ and } X=d_j. \end{cases}$$

We will show that
(a) the NFA M is a minimal NFA for the language $L(M)$;
(b) the language $L^c(M)$ can be accepted by an $(n+k)$-state DFA;
(c) every NFA for the language $L^c(M)$ needs at least $n+k$ states.
Then, the theorem follows immediately.

To prove (a) consider the following set of pairs of strings

$$\mathcal{A} = \{(a^{i-1}, a^{n-i}b) \mid i = 1, 2, \ldots, n\}.$$

The set $\mathcal{A}$ is a fooling set for $L(M)$ because for any i and j in $\{1, 2, \ldots, n\}$,

(1) $a^{i-1}a^{n-i}b \in L(M)$ since the string $a^{n-1}b$ is accepted by the NFA M, and

(2) if $i < j$, then $a^{i-1}a^{n-j}b \notin L(M)$ since for any l with $0 \leq l < n-1$, the string $a^l b$ is not accepted by the NFA M.

By Lemma 1, any NFA for $L(M)$ needs at least n states which proves (a).

To prove (b) let $M' = (2^Q, \Sigma, \delta', \{n\}, F')$ be the DFA obtained from the NFA M by the subset construction. Let $\mathcal{R}$ be the following system of sets:

$$\mathcal{R} = \{\{1\}, \{2\}, \ldots, \{n\}, S_1, S_2, \ldots, S_k\}.$$

Note that the initial state $\{n\}$ of the DFA M' and the state $S_1 = \{1, 2\}$ belong to $\mathcal{R}$. We are going to prove that any set in $\mathcal{R}$ is a reachable state of the DFA M' and no other states are reachable in the DFA M'. Clearly, any set of the system $\mathcal{R}$ is reachable since we have $\{i\} = \delta'(\{n\}, a^{n-i})$ for $i = 1, 2, \ldots, n$, and $S_j = \delta'(\{1\}, c_j)$ for $j = 1, 2, \ldots, k$. To prove that no other subset of Q is a reachable state of the DFA M' it is sufficient to show that for any state R in $\mathcal{R}$ and any symbol X in Σ, the state $\delta'(R, X)$ is a member of $\mathcal{R}$. There are three cases:

(i) $R = \{1\}$. Then we have ($j = 1, 2, \ldots, k$):

$$\delta'(\{1\}, X) = \begin{cases} \{1, 2\}, & \text{if } X = a, \\ \{2\}, & \text{if } X = b, \\ S_j, & \text{if } X = c_j, \\ \{1\}, & \text{if } 1 \in S_j \text{ and } X = d_j, \\ \{2\}, & \text{if } 1 \notin S_j \text{ and } X = d_j. \end{cases}$$

Since all sets on the right are in the system $\mathcal{R}$, we are ready in this case.

(ii) $R = \{i\}$ for an $i \neq 1$. Then for any X in Σ, the set $\delta'(\{i\}, X)$ is a singleton set and so is in $\mathcal{R}$.

(iii) $R = S_j$ for a j in $\{1, 2, \ldots, k\}$. Then the set $\delta'(S_j, a)$ is a subset of the set $\{1, 2, \ldots, \max S_k - 1\}$ or equals $\{1, 2\}$. Since the sets $S_1, S_2, \ldots, S_k$ are ordered according to their maxima, any subset of $\{1, 2, \ldots, \max S_k - 1\}$ is in $\mathcal{R}$. Next, the set $\delta'(S_j, b)$ is equal either to $\{1\}$ or to $\{1, 2\}$, and the set $\delta'(S_j, d_l),\ l = 1, 2, \ldots, k$, is equal either to $\{1\}$, or to $\{2\}$, or to $\{1, 2\}$.

Finally, the set $\delta'(S_j, c_l),\ l = 1, 2, \ldots, k$, is equal either to $\{1\}$ or to $S_l \cup \{1\}$. Since the set $S_l \cup \{1\}$ precedes the set S_l or equals S_l, we are ready in this case.

Thus we have shown that the DFA M' obtained from the NFA M by the subset construction has exactly $n + k$ reachable states. After exchanging the accepting and the rejecting states of the DFA M' we obtain an $(n + k)$-state DFA accepting the language $L^c(M)$ which proves (b).

To prove (c) consider the following sets of pairs of strings:

$$\mathcal{B} = \{(a^{i-1}, a^{n-i}) \mid i = 1, 2, \ldots, n\},$$

$$\mathcal{C} = \{(a^{n-1}c_j, d_j) \mid j = 1, 2, \ldots, k\}.$$

We will show that the set $\mathcal{B} \cup \mathcal{C}$ is a fooling set for the language $L^c(M)$:

(1) For any $i \in \{1, 2, \ldots, n\}$, the string $a^{i-1}a^{n-i}$ is a member of the language $L^c(M)$ since the string a^{n-1} is not accepted by the NFA M.
For any $j \in \{1, 2, \ldots, k\}$, the string $a^{n-1}c_jd_j$ is a member of the language $L^c(M)$ since

$$\delta(n, a^{n-1}) = \{1\}, \quad \delta(\{1\}, c_j) = S_j, \quad \delta(S_j, d_j) = \{1\}, \text{ and } 1 \notin F,$$

and so the string $a^{n-1}c_jd_j$ is not accepted by the NFA M.

(2) If $1 \leq i < s \leq n$, then the string $a^{i-1}a^{n-s}$ is not a member of the language $L^c(M)$ since the NFA M accepts any string a^l with $0 \leq l < n - 1$.
Next, if $1 \leq j, t \leq k$ and $j \neq t$, then, w.l.o.g., there is a state p in Q such that $p \in S_j$ and $p \notin S_t$, so

$$p \in \delta(n, a^{n-1}c_j) \text{ and } 2 \in \delta(p, d_t)$$

and so the string $a^{n-1}c_jd_t$ is accepted by the NFA M, i.e., is not a member of the language $L^c(M)$.

Finally, if $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, k\}$, then the string $a^{n-1}c_j a^{n-i}$ is not a member of the language $L^c(M)$ since $\delta(n, a^{n-1}c_j) = S_j$, the size of the set S_j is at least two, and the string a^{n-i} is not accepted by the NFA M starting in state $n - i + 1$ but it is accepted by M starting in any other state.

Thus the set $\mathcal{B} \cup \mathcal{C}$ is a fooling set for the language $L^c(M)$. By Lemma 1, any NFA for the language $L^c(M)$ needs at least $n + k$ states which completes our proof. □

Corollary 1. *For all positive integers r and s with $\log r \leq s \leq r$, there is a minimal NFA E of r states such that every minimal NFA accepting the complement of the language $L(E)$ has s states.*

Proof. Let r and s be arbitrary but fixed positive integers with $\log r \leq s \leq r$. Then we have

$$s \leq r \leq 2^s,$$

and by the above results, there is a minimal s-state NFA S such that a minimal NFA, say R, for the language $L^c(S)$ has r states. Set $E = R$. Then the NFA E is a minimal r-state NFA for the language $L^c(S)$ and a minimal NFA for the complement of the language $L^c(S)$, i.e., for the language $L^c(E)$, has s states. □

Hence, we have shown the following result.

Theorem 10. *For any positive integers n and α with $\log n \leq \alpha \leq 2^n$, there is a minimal NFA M of n states such that a minimal NFA accepting the complement of the language $L(M)$ has α states.* □

4.3.2 Linear Alphabet

In the case of complementation, we have shown that for any positive integers n and α with $\log n \leq \alpha \leq 2^n$, there is a minimal NFA M of n states such that a minimal NFA for the complement of the language $L(M)$ has exactly α states. However, the input alphabet size grows exponentially with n. We now show that the input alphabet can be decreased to linear size.

The next theorem shows that the nondeterministic state complexity of the complement of an n-state NFA language over a $2n$-letter alphabet may reach any value between $n + 1$ and 2^n.

Theorem 11. *For all integers n and α with $3 \leqslant n+1 \leqslant \alpha \leqslant 2^n$, there exists a minimal NFA M of n states with a $2n$-letter input alphabet such that every minimal NFA for the complement of the language $L(M)$ has exactly α states.*

Proof. Let n and α be arbitrary but fixed integers such that $n+1 \leqslant \alpha \leqslant 2^n$. If $\alpha < 2^n$, then there is an integer k in $\{1, 2, \ldots, n-1\}$ such that $n-k+2^k \leqslant \alpha < n-(k+1)+2^{k+1}$, and α can be expressed as

$$\alpha = n - k + 2^k + \sum_{i=0}^{k-1} c_i 2^i,$$

where $c_i \in \{0, 1\}$ for $i = 0, 1, \ldots, k-1$. If $\alpha = 2^n$, we take $k = n-1$ and $c_i = 1$ for $i = 0, 1, \ldots, k-1$. Let $I = \{i \in \{0, 1, \ldots, k-1\} \mid c_i = 1\}$ and let

$$\Sigma = \{a, a_k, b_k\} \cup \{a_i, b_i \mid i \in I\}.$$

We are going to define a minimal n-state NFA M with the input alphabet Σ such that the DFA obtained from the NFA M by the subset construction has α reachable states. After exchanging the accepting and the rejecting states we obtain an α-state DFA for the language $L(M)^c$. Then we show that any NFA for the language $L(M)^c$ has at least α states.

Define an n-state NFA $M = (\{1, 2, \ldots, n\}, \Sigma, \delta, n, \{1\})$, where for any $q \in \{1, 2, \ldots, n\}$ and for any $i \in I$,

$$\delta(q, a) = \begin{cases} \varnothing, & \text{if } q = 1 \text{ or } q > k+1, \\ \{k+1, q-1\}, & \text{if } 1 < q \leqslant k+1, \end{cases}$$

$$\delta(q, a_k) = \begin{cases} \varnothing, & \text{if } q = 1 \text{ or } q \geqslant k+1, \\ \{k+1, q-1\}, & \text{if } 1 < q < k+1, \end{cases}$$

$$\delta(q, b_k) = \begin{cases} \{1, 2, \ldots, k+1\}, & \text{if } q = 1, \\ \{k+1, q-1\}, & \text{if } 1 < q \leqslant k+1, \\ \{q-1\}, & \text{if } q > k+1, \end{cases}$$

$$\delta(q, a_i) = \begin{cases} \varnothing, & \text{if } q = 1 \text{ or } q \geqslant i+1, \\ \{i+1, q-1\}, & \text{if } 1 < q < i+1, \end{cases}$$

$$\delta(q, b_i) = \begin{cases} \varnothing, & \text{if } q = 1 \text{ or } i+1 < q < n, \\ \{i+1, q-1\}, & \text{if } 1 < q \leqslant i+1, \\ \{i+1\}, & \text{if } q = n. \end{cases}$$

The NFA M for $n = 8, k = 5$, and $I = \{3\}$ is shown in Fig. 4.14. Note that the symbols a_0 and a_1 are never used, and so the alphabet Σ has at most $2n$ letters.

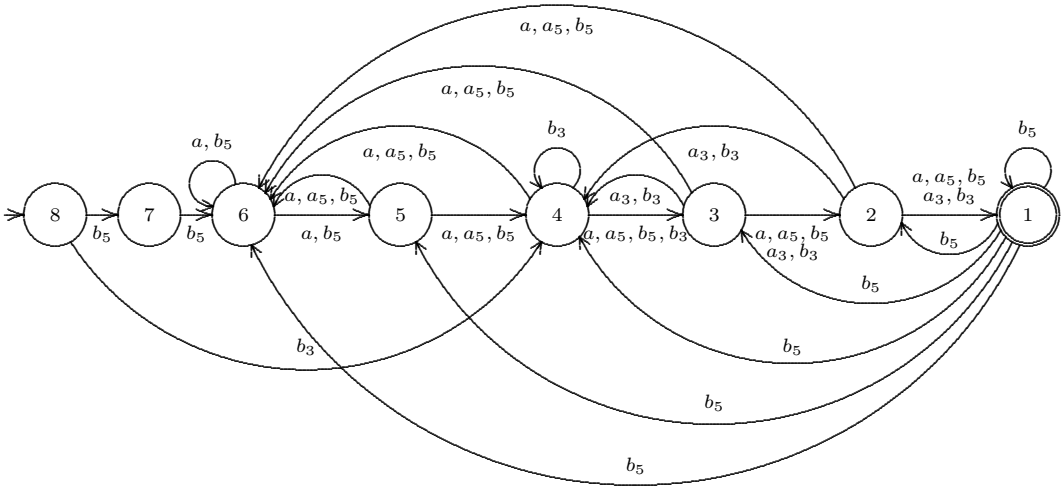

Figure 4.14: The NFA $M; n = 8, k = 5, I = \{3\}$.

To show that the NFA M is minimal consider the set of pairs of strings

$$\{(b_k^{i-1}, b_k^{n-i}) \mid i = 1, 2, \ldots, n\}.$$

It is a fooling set for the language $L(M)$ because for all i and j in $\{1, 2, \ldots, n\}$,
(1) $b_k^{i-1} b_k^{n-i} = b_k^{n-1}$ and the string b_k^{n-1} is in the language $L(M)$, and
(2) if $i < j$, then $b_k^{i-1} b_k^{n-j} = b_k^{n-1-(j-i)}$ and the string $b_k^{n-1-(j-i)}$ is not in the language $L(M)$ since the NFA M does not accept any string b_k^r with $r < n - 1$. By Lemma 1, any NFA for the language $L(M)$ needs at least n states and so the NFA M is minimal.

Let M' be the DFA obtained from the NFA M by the subset construction. We first show that the DFA M' has α reachable states.

The singletons $\{n\}, \{n-1\}, \ldots, \{k+2\}$, and the empty set are reachable since $\{q\} = \delta(\{n\}, b_k^{n-q})$ for $q = k+2, k+3, \ldots, n$ and $\varnothing = \delta(\{n\}, a_k)$.

We show that for any subset S of $\{1, 2, \ldots, k\}$ the set $\{k+1\} \cup S$ is reachable. We prove this by induction on the size of S. The set $\{k+1\}$ and the subsets $\{k+1, 1\}, \{k+1, 2\}, \ldots, \{k+1, k\}$ are reachable since $\{k+1\} = \delta(\{k+2\}, b_k)$ and $\{k+1, q\} = \delta(\{k+1\}, b_k a_k^{k-q})$ for $q = 1, 2, \ldots, k$. Let $2 \leqslant m \leqslant k$ and assume that any subset $\{k+1\} \cup S$ with $|S| = m - 1$ is reachable. Let $\{j_1, j_2, \ldots, j_m\}$, where $k \geqslant j_1 > j_2 \cdots > j_m \geqslant 1$ be a subset of size m. Then we have $\{k+1, j_1, j_2, \ldots, j_m\} =$
$\delta(\{k+1, k-j_1+j_2+1, k-j_1+j_3+1, \ldots, k-j_1+j_m+1\}, b_k a_k^{k-j_1})$, where the latter set is reachable by induction (note that $k \geqslant k - j_1 + j_r + 1 \geqslant 2$ for $r = 2, 3, \ldots, m$).

Next, we show that for any $i \in I$ and any subset S of $\{1, 2, \ldots, i\}$, the set $\{i+1\} \cup S$ is reachable. We prove this again by induction on the size of S. The set $\{i+1\}$ and the subsets $\{i+1, 1\}, \{i+1, 2\}, \ldots, \{i+1, i\}$ are reachable since $\{i+1\} = \delta(\{n\}, b_i)$ and $\{i+1, q\} = \delta(\{i+1\}, b_i a_i^{i-q})$ for $q = 1, 2, \ldots, i$. Let $2 \leqslant m \leqslant i$ and assume that any subset $\{i+1\} \cup S$ with $|S| = m-1$ is reachable. Let $\{j_1, j_2, \ldots, j_m\}$, where $i \geqslant j_1 > j_2 \cdots > j_m \geqslant 1$ be a subset of size m. Then we have $\{i+1, j_1, j_2, \ldots, j_m\} = \delta(\{i+1, i-j_1+j_2+1, i-j_1+j_3+1, \ldots, i-j_1+j_m+1\}, b_i a_i^{i-j_1})$, where the latter set is reachable by induction (note that $i \geqslant k - j_1 + j_r + 1 \geqslant 2$ for $r = 2, 3, \ldots, m$).

Thus we have shown that the DFA M' obtained from the NFA M by the subset construction has $n - k + 2^k + \sum_{i \in I} 2^i$ reachable states. Let $\mathcal{R}$ be the system of these α reachable sets, i.e., $\mathcal{R}$ contains the empty set, the singletons $\{n\}, \{n-1\}, \ldots, \{k+2\}$, the set $\{k+1\} \cup S$ for any $S \subseteq \{1, 2, \ldots, k\}$, and for any $i \in I$, the set $\{i+1\} \cup T$ for any $T \subseteq \{1, 2, \ldots, i\}$. To see that no other state is reachable note that the initial state $\{n\}$ is in $\mathcal{R}$ and for any set S in $\mathcal{R}$ and any symbol $X \in \Sigma$, the set $\delta(S, X)$ is also in $\mathcal{R}$ (the set $\delta(S, X)$ is either the empty set, or a subset of $\{1, 2, \ldots, k+1\}$ containing state $k+1$, or it is a subset of $\{1, 2, \ldots, i+1\}$ containing state $i+1$ for some $i \in I$). Hence the DFA M' obtained from the NFA M by the subset construction has exactly α reachable states. After exchanging the accepting and the rejecting states we obtain an α-state DFA for the language $L(M)^c$.

It remains to show that any NFA for the language $L(M)^c$ needs at least α states. To do this we describe a fooling set for this language of size α. We will do it in the following way. For any set S in $\mathcal{R}$, we define a pair of strings (x_S, y_S) such that the string $x_S y_S$ is in the language $L(M)^c$ and, moreover, if S and T are two different sets in $\mathcal{R}$, then at least one of the strings $x_S y_T$ and $x_T y_S$ is not in the language $L(M)^c$. Let $S \in \mathcal{R}$. Define the pair (x_S, y_S) as follows.

If $S = \varnothing$, let $x_S = a_k$ and $y_S = b_k^{n-1}$.

If $S = \{q\}$, where $k+2 \leqslant q \leqslant n$, let $x_S = b_k^{n-q}$ and $y_S = b_k^{q-2}$.

If S is a nonempty subset of $\{1, 2, \ldots, k+1\}$ that is in $\mathcal{R}$, let x_S be an arbitrary string such that $\delta(n, x_S) = S$.

If $S = \{1, 2, \ldots, k+1\}$, let $y_S = a_k^k$.

Otherwise, let $l = \max\{j \in \{1, 2, \ldots, k+1\} \mid j \notin S\}$, i.e., $l \notin S$ and $\{l+1, l+2, \ldots, k+1\} \subseteq S$. Define the string y_S of length $l-1$ as follows: $y_S = y_1 y_2 \cdots y_{l-1}$, where for any $i = 1, 2, \ldots, l-1$,

$$y_i = a \text{ if } i \in S, \text{ and } y_i = b_k \text{ if } i \notin S.$$

We first prove the following claim.

Claim. *For each subset S of $\{1,\ldots,k+1\}$ and each state p in $\{1,\ldots,k+1\}$,*
(a) if $p \in S$, then $1 \notin \delta(p, y_S)$,
(b) if $p \notin S$, then $1 \in \delta(p, y_S)$,
that is, the string y_S is not accepted by the NFA M starting in any state of S, but it is accepted by M starting in each state of $\{1, 2, \ldots, k+1\}$ that is not in the set S.

Proof of Claim. The claim holds if $S = \varnothing$ or $S = \{1, 2, \ldots, k+1\}$.

Otherwise, $y_S = y_1 y_2 \cdots y_{l-1}$, where $l \notin S, \{l+1, l+2, \ldots, k+1\} \subseteq S$, and for any $i \in \{1, 2, \ldots, l-1\}, y_i = a$ if $i \in S$ and $y_i = b_k$ if $i \notin S$.

To prove (a) let p be any state such that $p \in S$. There are two cases:

(i) $p > l$. Then the accepting state 1 cannot be reached from state p after reading the string y_S since the length of y_S is less than l. Hence $1 \notin \delta(p, y_S)$.

(ii) $p < l$. Then $y_S = y_1 y_2 \cdots y_{p-1} a y_{p+1} \cdots y_{l-1}$. Starting in state p and after reading the string $y_1 y_2 \cdots y_{p-1}$ we can either reach state 1 where the transition on reading the symbol a is not defined, or we can reach state $k+1$ after reading symbol y_j for some $j \leqslant p-1$ and then the length of the string $y_{j+1} y_{j+2} \cdots y_{l-1}$ is too short to reach state 1. Thus $1 \notin \delta(p, y_S)$.

To prove (b) let p be any state in $\{1, 2, \ldots, k+1\} \setminus S$. There are two cases:

(i) $p = l$. Then state 1 can be reached from state p after reading any string in $\{a, b_k\}^*$ of length $l-1$, so $1 \in \delta(p, y_S)$.

(*ii*) $p < l$. Then $y_S = y_1 y_2 \cdots y_{p-1} b_k y_{p+1} \cdots y_{l-1}$. Denote by d the length of the string $y_{p+1} y_{p+2} \cdots y_{l-1}$. Then $d \leqslant l-2 \leqslant k-1$. Starting in state p and after reading the string $y_1 y_2 \cdots y_{p-1}$ of length $p-1$ we can reach state 1. Then, on reading the symbol b_k we can reach state $d+1$, and then after reading the string $y_{p+1} y_{p+2} \cdots y_{l-1}$ of length d we can reach state 1. Hence $1 \in \delta(p, y_S)$ which completes the proof of the Claim.

Now, we are going to show that the set of pairs $\{(x_S, y_S) \mid S \in \mathcal{R}\}$ is a fooling set for the language $L(M)^c$. We need to show that (1) for any $S \in \mathcal{R}$, the string $x_S y_S$ is in the language $L(M)^c$, and (2) if $S \neq T$, then at least one of the strings $x_S y_T$ and $x_T y_S$ is not in $L(M)^c$.

To prove (1) let $S \in \mathcal{R}$. We have three cases:

(i) $S = \varnothing$. Then $x_S y_S = a_k b_k^{n-1}$. The string $a_k b_k^{n-1}$ is not accepted by the NFA M and so it is in the language $L^c(M)$.

(ii) $S = \{q\}$, where $k+2 \leqslant q \leqslant n$. Then $x_S y_S = b_k^{n-q} b_k^{q-2} = b_k^{n-2}$. The string b_k^{n-2} is not accepted by the NFA M and so it is in $L(M)^c$.

(iii) S is a nonempty subset of $\{1, 2, \ldots, k+1\}$. Then $\delta(n, x_S) = S$ and, by Claim (a), the string y_S is not accepted by the NFA M starting in any state of S. Hence the string $x_S y_S$ is in the language $L(M)^c$.

To prove (2) let S and T be two different sets in $\mathcal{R}$. We have four cases:

(i) $S = \varnothing$ and T is a nonempty subset of $\{1, 2, \ldots, n\}$. Then $x_T y_S = x_T b_k^{n-1}$, where $\delta(n, x_T) = T$. Since the string b_k^{n-1} is accepted by the NFA M starting in any state of T, the string $x_T b_k^{n-1}$ is not in $L(M)^c$.

(ii) $S = \{p\}$ and $T = \{q\}$, where $k+2 \leqslant p < q \leqslant n$. Then we have $x_S y_T = b_k^{n-p} b_k^{q-2} = b_k^{n-2+q-p}$. Since $n-2+q-p \geqslant n-1$, the string $b_k^{n-2+q-p}$ is accepted by the NFA M, so the string $x_S y_T$ is not in $L(M)^c$.

(iii) $S = \{q\}$, where $k+2 \leqslant q \leqslant n$, and T is a nonempty subset of $\{1, 2, \ldots, k+1\}$. Then $x_T y_S = x_T b_k^{q-2}$, where $\delta(n, x_T) = T$. Since $q-2 \geqslant k$, the string b_k^{q-2} is accepted by the NFA M starting in any state of the nonempty set T. Hence the string $x_T y_S$ is not in $L(M)^c$.

(iv) S and T are two different nonempty subsets of $\{1, 2, \ldots, k+1\}$. Then, without loss of generality, there is a state p in $\{1, 2, \ldots, k+1\}$ such that $p \notin S$ and $p \in T$. By Claim (b), the string y_S is accepted by the NFA M starting in state p. Since $\delta(n, x_T) = T$ and $p \in T$, the string $x_T y_S$ is accepted by the NFA M and so it is not in the language $L(M)^c$.

We have shown that the set of pairs of strings $\{(x_S, y_S) \mid S \in \mathcal{R}\}$ is a fooling set for the language $L(M)^c$. By Lemma 1, any NFA for the language $L(M)^c$ needs at least α states and our proof is complete. □

If $\log n \leqslant \alpha \leqslant n$, then $\alpha \leqslant n \leqslant 2^\alpha$, and so there is an α-state NFA A such that minimal NFAs for the language $L^c(A)$ have n states. Let M be a minimal NFA for the language $L^c(A)$. Then M is a minimal n-state NFA such that minimal NFAs for the language $L(M)^c$ have α states. Thus, as a corollary of the above results we have the following theorem which shows that the nondeterministic state complexity of the complement of an n-state NFA language may reach the entire range of values from $\log n$ to 2^n.

Theorem 12. *For all positive integers n and α such that $\log n \leq \alpha \leq 2^n$, there exists a minimal NFA M of n states with a $2n$-letter input alphabet such that every minimal NFA for the complement of the language $L(M)$ has exactly α states.* □

Chapter 5

Conclusion

In this thesis, we investigated the deterministic and nondeterministic state complexity of several operations on regular languages.

We showed that the upper bounds $m2^n - k2^{n-1}$ on the concatenation of an m-state DFA language and an n-state DFA language, where k is the number of the accepting states in the m-state automaton, are tight for each integer k with $0 < k < m$. We proved the result at first for a ternary and then for a binary alphabet.

Then we continued the investigation of the relations between the sizes of minimal nondeterministic and deterministic finite automata. We proved that for all integers n and α such that $1 \leqslant n \leqslant \alpha \leqslant 2^n$, there exists a minimal nondeterministic finite automaton of n states with a four-letter input alphabet whose equivalent minimal deterministic finite automaton has α states. This improves the result of [12] that has been obtained using a growing alphabet of size $n+2$. Recently, it has been shown in [30] that no magic numbers exist even in the ternary case.

We next examined the state complexity and the nondeterministic state complexity of languages resulting from the union and intersection of two regular languages. In the deterministic case, we showed that the entire range of complexities between 1 and mn can be obtained by the union or intersection of an m-state DFA language and an n-state DFA language for any integers m and n such that $m \geqslant 2$ and $n \geqslant 2$. Next, we proved that the nondeterministic state complexity of the union of an m-state NFA language and an n-state NFA language may be arbitrary between 1 and $m+n+1$, except for the case of $m = 1$ and $n = 1$ when the union has nondeterministic state complexity 1 or 3. To prove these results we used a binary alphabet. In the case of a unary alphabet, similar results probably do not hold. Finally, we showed that the nondeterministic state complexity of the intersection of an m-state NFA language and an n-state NFA language may be arbitrary between 1 and

mn. We proved the last result for a ternary alphabet. The question whether this result holds likewise for a binary alphabet remains open.

At the end, we studied the nondeterministic state complexity of complements of regular languages. We showed that for all integers n and α with $\log n \leqslant \alpha \leqslant 2^n$, there is a regular language with nondeterministic state complexity n such that the nondeterministic state complexity of its complement is α. We presented an easy proof that uses an exponential alphabet, and a more elaborated one using an alphabet of size $2n$. Nevertheless, the problem remains open for a fixed alphabet.

Bibliography

[1] H. N. Adorna, 3-Party message complexity is better than 2-party ones for proving lower bounds on the size of minimal nondeterministic finite automata. *J. Autom. Lang. Comb.* **7** (2002) 419-432.

[2] A. V. Aho, J. D. Ullman, and M. Yannakakis, On notions of information transfer in VLSI circuits. In: *Proc. 15th Annual ACM Symp. on Theory of Computing (STOC),* 1983, pp. 133–139.

[3] P. Berman, A. Lingas, *On the complexity of regular languages in terms of finite automata,* Technical Report 304, Polish Academy of Sciences, 1977.

[4] J.C. Birget, Intersection and union of regular languages and state complexity, *Inform. Process. Lett.* **43** (1992) 185–190.

[5] J.C. Birget, Partial orders on words, minimal elements of regular languages, and state complexity, *Theoret. Comput. Sci.* **119** (1993) 267–291. ERRATUM: Partial orders on words, minimal elements of regular languages, and state complexity, 2002. Available at `http://clam.rutgers.edu/~birget/papers.html`.

[6] C. Câmpeanu, K. Culik II, K. Salomaa, S. Yu, State complexity of basic operations on finite languages, in: O. Boldt, H. Jürgensen (Eds.), *Proc. 4th International Workshop on Implementing Automata (WIA'99),* LNCS 2214, Springer-Verlag, Heidelberg, 2001, pp. 60–70.

[7] C. Câmpeanu, K. Salomaa, S. Yu, Tight lower bound for the state complexity of shuffle of regular languages, *J. Autom. Lang. Comb.* **7** (2002) 303–310.

[8] M. Chrobak, Finite automata and unary languages. *Theoret. Comput. Sci.* **47** (1986) 149–158. ERRATUM: *Theoret. Comput. Sci.* **302** (2003) 497–498.

[9] M. Domaratzki, State complexity and proportional removals, *J. Autom. Lang. Comb.* **7** (2002) 455–468.

[10] K. Ellul, *Descriptional complexity measures of regular languages,* Master's thesis, University of Waterloo, 2002.

[11] K. Ellul, B. Krawetz, J. Shallit, and M. W. Wang, Regular expressions: new results and open problems, *J. Autom. Lang. Comb.* 10 (2005) 407437.

[12] V. Geffert, (Non)determinism and the size of one-way finite automata. In: C. Mereghetti, B. Palano, G. Pighizzini, D. Wotschke (Eds.), *Proc. 7th Workshop on Descriptional Complexity of Formal Systems (DCFS 2005),* University of Milano, Italy, 2005, pp. 23–37.

[13] V. Geffert, Magic numbers in the state hierarchy of finite automata. In: R. Královič, P. Urzyczyn (Eds.) *Proc. 31st International Symposium on Mathematical Foundations of Computer Science (MFCS 2006),* Lecture Notes in Comput. Sci., 4162, Springer, Berlin, 2006, pp. 412–423.

[14] I. Glaister, J. Shallit, A lower bound technique for the size of nondeterministic finite automata, *Inform. Process. Lett.* **59** (1996) 75–77.

[15] M. Holzer, K. Salomaa, and S. Yu, On the state complexity of k-entry deterministic finite automata, *J. Autom. Lang. Comb.* **6** (2001) 453–466.

[16] M. Holzer, M. Kutrib, State complexity of basic operations on nondeterministic finite automata, in: J.M. Champarnaud, D. Maurel (Eds.), *Implementation and Application of Automata (CIAA 2002),* LNCS 2608, Springer-Verlag, Heidelberg, 2003, pp. 148–157.

[17] M. Holzer, M. Kutrib, Unary language operations and their nondeterministic state complexity, in: M. Ito, M. Toyama (Eds.), *Developments in Language Theory (DLT 2002),* LNCS 2450, Springer-Verlag, Heidelberg, 2003, pp. 162–172.

[18] M. Holzer, M. Kutrib, Nondeterministic descriptional complexity of regular languages, *Internat. J. Found. Comput. Sci.* **14** (2003) 1087-1102.

[19] J. Hromkovič, *Communication Complexity and Parallel Computing,* Springer-Verlag, Berlin, Heidelberg, 1997.

[20] J. Hromkovič, Descriptional complexity of finite automata: concepts and open problems, *J. Autom. Lang. Comb.* **7** (2002) 519–531.

[21] J. Hromkovič, S. Seibert, J. Karhumäki, H. Klauck, and G. Schnitger, Communication complexity method for measuring nondeterminism in finite automata. *Inform. and Comput.* **172** (2002) 202–217.

[22] K. Iwama, Y. Kambayashi and K. Takaki, Tight bounds on the number of states of DFAs that are equivalent to n-state NFAs. *Theoret. Comput. Sci.* **237** (2000) 485–494.

[23] K. Iwama, A. Matsuura and M. Paterson, A family of NFAs which need $2^n - \alpha$ deterministic states, *Theoret. Comput. Sci.* **301** (2003) 451–462.

[24] J. Jirásek, G. Jirásková, A. Szabari, State complexity of concatenation and complementation, *Internat. J. Found. Comput. Sci.* **16** (2005) 511–529.

[25] G. Jirásková, Note on minimal finite automata, in: *Proc. MFCS 2001,* Lecture Notes in Comput. Sci., 2136, Springer, Berlin, 2001, pp. 421–431.

[26] G. Jirásková, Note on minimal finite automata. In: J. Sgall, A. Pultr, P. Kolman (Eds.), *Proc. 26th International Symposium on Mathematical Foundations of Computer Science (MFSC 2001),* Lecture Notes in Comput. Sci., 2136, Springer, Berlin, 2001, pp. 421–431.

[27] G. Jirásková, Note on minimal automata and uniform communication protocols, in: C. Martin-Vide and V. Mitrana (Eds.), *Grammars and Automata for String Processing: From Mathematics and Computer Science to Biology, and Back,* Taylor and Francis, London, 2003, pp. 163–170.

[28] G. Jirásková, State complexity of some operations on regular languages, in: E. Csuhaj-Varjú, C. Kintala, D. Wotschke, Gy. Vaszil (Eds.), *Proc. 5th Workshop Descriptional Complexity of Formal Systems,* MTA SZTAKI, Budapest, 2003, pp. 114–125.

[29] G. Jirásková, State complexity of some operations on binary regular languages, *Theoret. Comput. Sci.* **330** (2005) 287–298.

[30] G. Jirásková: Magic numbers and ternary alphabet. In: Proceedings of the 13th International Conference on Developments in Language Theory (DLT 2009, Stuttgart, Germany, June 30 - July 3), Diekert, V., Nowotka, D. (eds.), Lecture Notes in Computer Science 5583, Springer, Heidelberg, 2009, pp. 300-311.

[31] A. Kapoutsis, Ch.A.: Size Complexity of Two-Way Finite Automata. Developments in Language Theory, 47-66, 2009.

[32] E. Leiss, Succinct representation of regular languages by boolean automata, *Theoret. Comput. Sci.* **13** (1981) 323–330.

[33] O. B. Lupanov, A comparison of two types of finite automata, *Problemy Kibernetiki,* **9** (1963) 321-326 (in Russian).

[34] Yu. I. Lyubich, Estimates for optimal determinization of nondeterministic autonomous automata, *Sibirskii Matematichskii Zhurnal,* **5** (1964) 337-355 (in Russian).

[35] A. N. Maslov, Estimates of the number of states of finite automata, *Dokl. Akad. Nauk SSSR,* **194** (1970) 1266–1268 (in Russian). English translation: *Soviet Math. Dokl.* **11** (1970) 1373–1375.

[36] F. R. Moore, On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata, *IEEE Trans. Comput.* **20** (1971) 1211–1214.

[37] A.R. Meyer and M.J. Fischer, Economy of description by automata, grammars and formal systems, in: *Proc. 12th Annual Symposium on Switching and Automata Theory,* 1971, pp. 188–191.

[38] G. Pighizzini, Unary language concatenation and its state complexity, in: S. Yu, A. Pun (Eds.), *Implementation and Application of Automata: 5th International Conference, CIAA 2000,* LNCS 2088, Springer-Verlag, 2001, pp. 252–262.

[39] G. Pighizzini, J. Shallit, Unary language operations, state complexity and Jacobsthal's function, *Internat. J. Found. Comput. Sci.* **13** (2002) 145–159.

[40] M. Rabin, D. Scott, Finite automata and their decision problems, *IBM Res. Develop.* **3** (1959) 114–129.

[41] W.J. Sakoda, M. Sipser, Nondeterminism and the size of two-way finite automata, in: *Proc. 10th Annual ACM Symposium on Theory of Computing,* 1978, pp. 275–286.

[42] A. Salomaa, D. Wood, and S. Yu, On the state complexity of reversals of regular languages, *Theoret. Comput. Sci.* **320** (2004) 315–329.

[43] M. Sipser, *Introduction to the theory of computation,* PWS Publishing Company, Boston, 1997.

[44] S. Yu, Q. Zhuang, K. Salomaa, The state complexity of some basic operations on regular languages, *Theoret. Comput. Sci.* **125** (1994) 315–328.

[45] S. Yu, Chapter 2: Regular languages, in: G. Rozenberg, A. Salomaa, (Eds.), *Handbook of Formal Languages - Vol. I,* Springer-Verlag, Berlin, New York, pp. 41–110.

[46] S. Yu, A renaissance of automata theory? *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS* **72** (2000) 270–272.

[47] S. Yu, State complexity of finite and infinite regular languages, *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS* **76** (2000) 270–272.

[48] S. Yu, State complexity of regular languages, *J. Autom. Lang. Comb.* **6** (2001) 221–234.

[49] Zijl, L.: Magic numbers for symmetric difference NFAs. Internat. J. Found. Comput. Sci. 16, 1027–1038 (2005)