Descriptional Complexity Measures of Regular Languages

by

Keith Ellul

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Master of Mathematics

in

Computer Science

Waterloo, Ontario, Canada, 2004

I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Waterloo to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

In this thesis, we study various complexity measures of regular languages. In particular, we study state complexity, nondeterministic state complexity, regular expression size, radius, and nondeterministic state complexity. We compare these different measures, and study the effects of various operations such as union, intersection, concatenation, Kleene closure, and reversal.

Acknowledgements

I would like to thank Jeff Shallit for his supervision and guidance throughout the writing of this thesis. I would also like to thank Jonathan Buss and Dan Brown for agreeing to read this thesis and give suggestions.

Jean–Camille Birget discussed one of his papers with me, and clarified one of his previous results, which I found extremely helpful. Also, I thank Martin Kutrib for sending me one of his papers which contained many useful results. I also thank Ming–Wei Wang, Mike Domaratzki, and Andrew Martinez for our helpful discussions on many problems.

Finally, I would like to thank my parents for their support and encouragement throughout my entire education.

Contents

Chapter 1

Introduction

There are many different areas of theoretical computer science that deal with some sort of complexity. *Computational complexity* measures the difficulty of determining whether a certain word is in a language. This is usually measured in terms of time or space required by an algorithm or Turing machine, although there are other possibilities. For example, *communication complexity* measures the minimum amount of information that must be communicated between two simultaneously–run algorithms, when each algorithm only has access to half of the input.

We are interested in studying regular languages. In terms of computational complexity, these are exactly the languages that can be accepted using constant space. However, we wish to study their complexity in more detail, so we will consider descriptional complexity. Intuitively, the *descriptional complexity* of a language is the amount of information needed to describe the language. As there are many different ways to describe a regular language, there are many different measures of descriptional complexity.

Regular languages are exactly those languages that are accepted by deterministic finite automata (DFAs). One way to measure the descriptional complexity of a regular language is to count the number of states in the smallest DFA which accepts that language. This is known as the *state complexity* of the language. The effects that various operations have on the state complexities of regular languages has been studied by many people, including Yu, Zhuang, and Salomaa [24, 26] and Birget [2]. We will examine state complexity in Chapter 3 of this thesis. However, because state complexity has already been so extensively studied by others, most of this section is simply a summary of previous work.

Another characterization of regular languages is that they are exactly those languages which are recognized by nondeterministic finite automata (NFAs). This leads us to consider another way of measuring the complexity of a regular language, that is, by counting the number of states in the smallest NFA which accepts that language. We will refer to this as the *nondeterministic state complexity* of the language, which will be studied in Chapter 4. Relationships between deterministic and nondeterministic state complexity are quite well known. Rabin and Scott [20] showed that a minimal DFA can require, at most, exponentially more states than an NFA that accepts the same language. Moore [19] showed that this exponential gap is, in fact, achievable. Chrobak [5] studied this problem for unary languages, and found an asymptotically tight bound on the possible increase from nondeterministic to deterministic state complexity in the unary case.

Birget [3, 4] studied the effect of complementation on the nondeterministic state complexity of a regular language. However, other than this, very little attention had been paid to the effects of various operations on the nondeterministic state complexity of a regular language at the time that this thesis was started. Many of these results were discovered and are original to this thesis. However, many of these results were studied independently, and subsequently published, by Holzer and Kutrib [10, 11].

A third characterization of regular languages is that they are exactly those languages that can be specified by a regular expression. Although a regular expression is arguably the most intuitive way of specifying a regular language, very little work has been done regarding minimal regular expressions. Ehrenfeucht and Zeiger [8] studied many different complexity measures of regular expressions. However, instead of using fixed–sized alphabets, they studied languages over alphabets that grew arbitrarily in size. In Chapter 5 we will study the *size* (number of alphabetical symbols in) a minimal regular expression for a language, and how this size is affected by the application of certain regularity–preserving operations. Many of these results can also be found in a paper published by Ellul, Shallit and Wang [9].

As was noted earlier, regular languages are exactly those languages that can be accepted by finite automata (deterministic or nondeterministic). Now, we will look to another way of measuring their complexity. While state complexity is a measure of the *size* of a finite automaton, if we consider radius as well as size, we get an idea of its *shape*. Intuitively, the radius of a finite automaton is the distance from the start state to the state that is furthest away. As a result, a finite automaton whose radius is small compared to its size has a compact "tree–like" shape, while a finite automaton whose radius is close to its size has more of a "path–like" shape.

The radius of a regular language is the radius of the minimal–radius DFA which accepts that language. Similarly, the nondeterministic radius of a regular language is the radius of the minimal–radius NFA which accepts that language. These are similar measures to the one studied by Barzdin & Koršunov [1]. We will discuss deterministic and nondeterministic radius in Chapters 6 and 7 respectively. As these subjects have not been extensively studied in the past, the results presented in these chapters are original to this thesis.

Chapter 2

General Notation

Throughout this thesis, we will use the following standard notation, much of which is borrowed from Hopcroft & Ullman [12] and the *Handbook of Formal Languages* [21]:

An *alphabet* Σ is a finite set of symbols. Elements of Σ are be referred to as *letters*. The *Kleene closure* of Σ, denoted by Σ^*, is the set of all finite sequences of letters from Σ. Elements of Σ^* are referred to as *words*. For example, $\{a, b\}$ is a two–letter alphabet, and aba is a word in $\{a, b\}^*$.

If an alphabet Σ contains only one letter, then a language over Σ is referred to as a *unary language*. The length of a word x is denoted by $|x|$. For a particular letter a, the number of occurrences of a in x is denoted by $|x|_a$. Also, the ith letter of x is denoted by $x[i]$. Finally, the reversal of x is denoted by x^R. For example, if $\Sigma = \{a, b\}$ and $x = abb$, then $|x| = 3$, $|x|_a = 1$, $|x|_b = 2$, $x[2] = b$, and $x^R = bba$.

For a language L over an alphabet Σ:

- The *complement* of L is denoted by $\overline{L}$ and is defined as $\{x \in \Sigma^* : \ x \notin L\}$.

- The *reversal* of L is defined as $\{w^R : w \in L\}$, and is denoted by L^R.
- $\frac{1}{2}L$ denotes the language $\{x \in \Sigma^* \ : \ \exists y \in \Sigma^*; \ xy \in L; \ |x| = |y|\}$.
- The *left quotient* of L by a word w, denoted by $w^{-1}L$, is $\{x \in \Sigma^* : \ wx \in L\}$.
- The *right quotient* of L by a word w, denoted by Lw^{-1}, is $\{x \in \Sigma^* : \ xw \in L\}$.

It should be noted that, in the literature, different notation is sometimes used for quotients. For example, Hopcroft & Ullman [12] use L/w to denote the (right) quotient of L by w. We use the notation above to eliminate any possible confusion between left and right quotients.

For languages L_1 and L_2 over an alphabet Σ:

- The *union* of L_1 and L_2 is denoted by $L_1 \cup L_2$ and is defined as $\{x \in \Sigma^* : \ x \in L_1 \text{ or } x \in L_2\}$.
- The *intersection* of L_1 and L_2 is denoted by $L_1 \cap L_2$ and is defined as $\{x \in \Sigma^* : \ x \in L_1 \text{ and } x \in L_2\}$.
- The *concatenation* of L_1 and L_2 is denoted by L_1L_2 and is defined as $\{xy \in \Sigma^* : \ x \in L_1 \text{ and } y \in L_2\}$.

Finally, if L is a regular language, then $L^0 = \{\epsilon\}$, and, recursively, $L^i = LL^{i-1}$ for all $i > 0$. The *Kleene closure* of a language L is denoted by L^* and is defined as:

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

When we say that a bound is *tight*, we mean that the bound is achievable for arbitrarily large instances. For example, in Theorem 7 we show that for any NFA of size n, there is an equivalent DFA of size no more than 2^n. Furthermore, we show that this bound is *tight*. In other words, we show that there exist arbitrarily large n such that there is an NFA M_n of size n, with the smallest equivalent DFA having size exactly 2^n. If the bounds is a function of two or more variables, we say that it is tight if it is achievable for arbitrarily large values of all the variables.

Chapter 3

State Complexity

The state complexity of a regular language is the number of states in the (unique) minimal deterministic finite automaton which accepts that language. We will study the effects that various regularity–preserving operations have on the state complexity of a regular language.

3.1 Definitions

A deterministic finite automaton (DFA) is defined as a quintuple $(Q, \Sigma, \delta, q_0, F)$, where:

- Q is a finite set of *states*;
- Σ is a finite *alphabet*;
- δ is a *transition function* that maps $Q \times \Sigma$ to Q;
- q_0 is the *start state*, which is an element of Q; and

- $F \subseteq Q$ is the set of *final states*.

Furthermore, we can extend δ to a function δ^* that maps $Q \times \Sigma^*$ to Q as follows: for letters $a_1, a_2, \ldots a_k$, we define $\delta^*(q, a_1 a_2 \cdots a_k) = \delta(\delta(\ldots \delta(\delta(q, a_1), a_2), \ldots a_{k-1}), a_k)$.

For a DFA $M = (Q, \Sigma, \delta, q_0, F)$, the language accepted by M, or $L(M)$, is defined as:

$$L(M) = \{x \in \Sigma^* \ : \ \delta^*(x, q_0) \in F\}.$$

Two DFAs M and M' are said to be *equivalent* if $L(M) = L(M')$. The *size* of a DFA M is the number of states in M, and is denoted by $|M|$. A DFA M is *minimal* if there is no equivalent DFA M' with $|M'| < |M|$. It is well known that each regular language has a unique (up to isomorphism) minimal DFA. This is a direct result of the Myhill–Nerode Theorem (see, for example, Hopcroft & Ullman [12, Theorems 3.9 & 3.10]).

We are now ready to define state complexity. For a regular language L, the *state complexity* of L, denoted by $\mathrm{sc}(L)$, is the number of states in the minimal DFA for L.

We can represent our DFAs visually by using *transition diagrams*. Each state is represented by a node, and there is a directed edge from state p to state q with the label "a" if and only if $\delta(p, a) = q$. Final states are denoted by double circles. For example, see Figure 3.1. In this example, $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $F = \{q_0\}$, and δ is defined as follows:

$$\begin{array}{rclcrclcrcl}
\delta(q_0, a) & = & q_1; & \quad & \delta(q_1, a) & = & q_2; & \quad & \delta(q_2, a) & = & q_2; \\
\delta(q_0, b) & = & q_2; & \quad & \delta(q_1, b) & = & q_0; & \quad & \delta(q_2, b) & = & q_2.
\end{array}$$

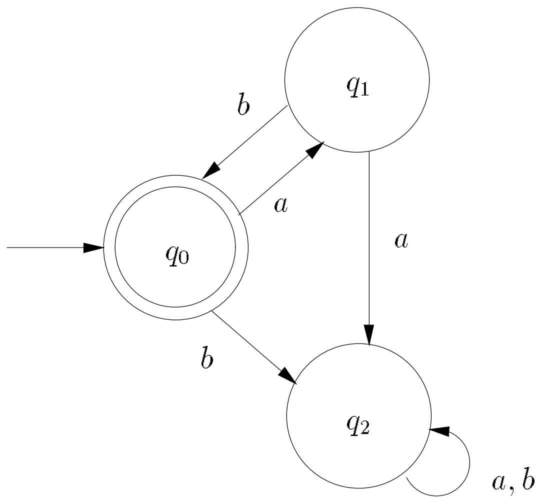

Figure 3.1: The transition diagram for a DFA $M = (Q, \Sigma, \delta, q_0, F)$

3.2 Unary Languages

Suppose L_1 and L_2 are unary regular languages with state complexity m and n respectively. Then, the following inequalities hold, and are tight:

- $\mathrm{sc}(L_1 \cup L_2) \leq mn$,
- $\mathrm{sc}(L_1 \cap L_2) \leq mn$,
- $\mathrm{sc}(L_1 L_2) \leq mn$, and
- $\mathrm{sc}(L_1^*) \leq (m-1)^2 + 1$.

Additionally, the following equalities hold:

- $\mathrm{sc}(L_1^R) = m$, and
- $\mathrm{sc}(\overline{L_1}) = m$.

These results are due to Yu, Zhuang, and Salomaa [26, 24].

In addition to the bounds listed above, there are also tight upper bounds on the state complexity of $\frac{1}{2}L$ and the quotient of L by a word.

Theorem 1 (Domaratzki [6]) *Let L be a unary regular language. Then $\mathrm{sc}(\frac{1}{2}L) \leq \mathrm{sc}(L)$. Additionally, this bound is tight.*

Proof:

Let $M = (Q, \{1\}, \delta, q_0, F)$ be the minimal DFA for L. Then define $\delta' : Q \to Q$ in the following way: $\delta'(q) = \delta^2(q)$. Then the DFA $M' = (Q, \{1\}, \delta', q_0, F)$ accepts $\frac{1}{2}(L)$.

To see that the bound is tight, consider the language $L_k = \{1^n : n \equiv 0 \pmod{k}\}$ for some arbitrary k. Then, when k is odd, $\frac{1}{2}L = L$ and so $\mathrm{sc}(\frac{1}{2}L) = \mathrm{sc}(L)$, as desired.

To see that the inequality is necessary, that is, the equality does not always hold, consider the case of L_k, where k is even. Then $\mathrm{sc}(L) = k$ but $\mathrm{sc}(\frac{1}{2}L) = k/2$. $\square$

This is an alternate proof to the one given by Domaratzki [6].

Finally, we have tight upper bounds for both left and right quotients in the unary case.

Theorem 2 *Let L be a unary regular language, and let w be a unary word. Then $\mathrm{sc}(w^{-1}L) = \mathrm{sc}(Lw^{-1}) \leq \mathrm{sc}(L)$. Additionally, this bound is tight.*

Proof:

First, note that, since Σ is unary, there is at most one word in L of each length. Thus, $w^{-1}L = Lw^{-1} = \left\{x \in 1^* : 1^{|x|+|w|} \in L\right\}$. So clearly $\mathrm{sc}(w^{-1}L) = \mathrm{sc}(Lw^{-1})$, since the languages are the same.

It should be noted that the inequality is simply a special case of Theorem 6 below, where the bounds are proven in the general (not necessarily unary) case. However, for completeness, they will be proven here as well. Let $M = (Q, \{1\}, \delta, q_0, F)$ be the minimal DFA for L. Define $\delta'(q_0) = \delta^{|w|+1}(q_0)$ and $\delta'(q) = \delta(q)$ for all $q \neq q_0$. Then the DFA $M' = (Q, \{1\}, \delta', q_0, F)$ accepts $w^{-1}L = Lw^{-1}$.

To see that the bound is tight, consider the language $L_k = \{1^n : n \equiv 0 \pmod{k}\}$ for some arbitrary k. Clearly $\mathrm{sc}(L_k) = k$. Now, consider an arbitrary word $w \in 1^*$. Then $w^{-1}L_k = \{1^n : n - |w| \equiv 0 \pmod{k}\}$, which has state complexity k.

Finally, to see that the inequality is necessary, that is, the equality does not always hold, define $L_k = \left\{1^k\right\}$. Then $\mathrm{sc}(L) = k+1$ and if we choose $|w| \leq k$, then $\mathrm{sc}(w^{-1}L) = k - |w| + 1$. (If $|w| > k$, then $\mathrm{sc}(w^{-1}L) = 1$.) $\square$

3.3 Languages Over an Arbitrary Alphabet

Suppose that L_1 and L_2 are regular languages over an arbitrary (not necessarily unary) alphabet Σ, and suppose that they are accepted by minimal DFAs $M_1 =$

$(Q_1, \Sigma, \delta, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta, q_2, F_2)$ respectively. Furthermore, suppose that $|Q_1| = m$ and $|Q_2| = n$. Then, the following bounds are tight:

- $\mathrm{sc}(L_1 \cup L_2) \leq mn$,
- $\mathrm{sc}(L_1 \cap L_2) \leq mn$,
- $\mathrm{sc}(L_1 L_2) \leq m2^n - k2^{n-1}$, where $k = |F_1|$,
- $\mathrm{sc}(L_1^*) \leq 2^{m-1} + 2^{m-2}$, and
- $\mathrm{sc}(L_1^R) \leq 2^m$.

Also, $\mathrm{sc}(\overline{L_1}) = m$. These results are due to Yu, Zhuang, and Salomaa [26, 24].

In fact, these results may be extended to cover the intersection and union of arbitrarily many languages. Yu & Zhuang [25] showed the following:

Theorem 3 (Yu & Zhuang [25]) *Given any integer $k > 0$ and k nonnegative integers $n_1, \ldots, n_k$, there exist k regular languages $L_1, \ldots, L_k$ where for each $1 \leq i \leq k$, L_i is accepted by an n_i–state DFA, and any DFA accepting $\bigcap_{1 \leq i \leq k} L_i$ has at least $n_1 \cdots n_k$ states.*

Birget [2] extended this result as follows:

Theorem 4 (Birget [2]) *For any n, and any $k \leq n$, there exist languages $L_1, L_2, \ldots, L_k$, such that the state complexity of each L_i is n, and the state complexities of both $\cap_{1 \leq i \leq k} L_i$ and $\cup_{1 \leq i \leq k} L_i$ is exactly n^k.*

3.3.1 Proportional Removals and $\frac{1}{2}L$

Domaratzki [6] has extensively studied the effects of *proportional removals* on the state complexity of a language. (The "$\frac{1}{2}$" operator is just a special case of a proportional removal) He gives a tight upper bound on the state complexity of $\frac{1}{2}L$.

Theorem 5 (Domaratzki [6]) *If L is a regular language with state complexity n, then $\mathrm{sc}(\frac{1}{2}L) = O(n \cdot e^{\sqrt{n \log n}(1+o(1))})$, and this bound is tight.*

3.3.2 Quotients

Suppose that L is a regular language over Σ, and w is a word in Σ^*. Unlike the special case of unary languages, $w^{-1}L$ and Lw^{-1} may be different languages, so we must treat the cases of left and right quotient separately.

Theorem 6 *For a regular language L over Σ, and a word $w \in \Sigma^*$, if $\mathrm{sc}(L) = n$, then:*

- $\mathrm{sc}(Lw^{-1}) \leq n$, *and*
- $\mathrm{sc}(w^{-1}L) \leq n$.

Furthermore, these bounds are tight.

Proof:

Let $M = (Q, \Sigma, \delta, q_0, F)$ be the minimal DFA for L. Define F' as follows: $F' = \{q \in Q : \delta^*(q, w) \in F\}$. Then $M' = (Q, \Sigma, \delta, q_0, F')$ accepts Lw^{-1}. Similarly, $M'' = (Q, \Sigma, \delta, \delta^*(q_0, w), F)$ accepts $w^{-1}L$. This establishes the bounds.

To see that these bounds are tight, refer to Theorem 2. The examples that show these bounds to be tight in the unary case also show them to be tight in the general case. □

Chapter 4

Nondeterministic State Complexity

In this section, we will discuss the nondeterministic state complexity of regular languages. We will examine how the nondeterministic state complexity of a regular language is related to its (deterministic) state complexity, and give tight bounds on the increase in nondeterministic state complexity when certain operations, which preserve regularity, are applied to a regular language.

4.1 Definitions

A nondeterministic finite automaton (NFA) is defined as a quintuple $(Q, \Sigma, \delta, q_0, F)$, where Q, Σ, q_0 and F are defined as for a DFA (see Section 3.1). The transition function, δ, is a function that maps $Q \times \Sigma$ to 2^Q, the power set of Q. That is, for any alphabet symbol a and any state q, $\delta(q, a)$ is a *set of states*. Recall that, for a DFA, $\delta(q,a)$ must be a *single state*. Therefore, NFAs are less restrictive than DFAs.

As was the case with DFAs, we can represent our NFAs as directed graphs. Each state will be represented by one node, and, for each state q and each alphabet symbol a, there is a directed edge with the label "a" from q to each state in $\delta(q,a)$. For example, see Figure 4.1. Final states are represented by double circles.

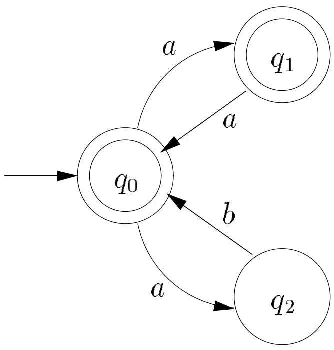

Figure 4.1: An NFA $M = (Q, \Sigma, \delta, q_0, F)$

In this example, $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $F = \{q_0, q_1\}$, and δ is defined as follows:

$$\begin{array}{rclrclrcl} \delta(q_0,a) &=& \{q_1,q_2\}\,; & \delta(q_1,a) &=& \{q_0\}\,; & \delta(q_2,a) &=& \emptyset; \\ \delta(q_0,b) &=& \emptyset; & \delta(q_1,b) &=& \emptyset; & \delta(q_2,b) &=& \{q_0\}\,. \end{array}$$

As was the case for DFAs, the definition of δ can be extended to define δ^*. If a is a letter in Σ, then $\delta^*(q,a) = \delta(q,a)$. Recursively, if w is a word in Σ^*, then $\delta^*(q,aw) = \bigcup_{q' \in \delta^*(q,w)} \{\delta(q',a)\}$.

The language accepted by an NFA $M = (Q, \Sigma, \delta, q_0, F)$ is denoted by $L(M)$

and defined as $\{x \in \Sigma^* \ : \ \delta^*(q_0, x) \cap F \neq \emptyset\}$. Two NFAs M and M' are said to be *equivalent* if $L(M) = L(M')$. For an NFA M, the *size* of M is defined as the number of states in M and is denoted by $|M|$. Furthermore, M is said to be *minimal* if there is no equivalent NFA that uses fewer states. However, unlike DFAs, two minimal NFAs for a language need not be isomorphic. For example, consider the regular language $L = \{a^i \ : \ i > 0\}$. Two non–isomorphic two–state NFAs that accept L are shown in Figure 4.2. Clearly, L cannot be accepted by a one–state NFA (since any NFA accepting L must contain a final state, but the start state cannot be final, as the empty word is not in L). So, any two–state NFA that accepts L must be minimal.

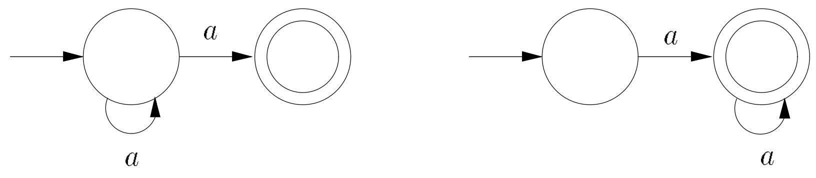

Figure 4.2: Two non–isomorphic minimal NFAs for $L = \{a^i \ : \ i > 0\}$.

We may also extend our definition of an NFA to include ϵ–transitions (that is, transitions that are labelled "ϵ"). We will refer to an NFA that includes ϵ–transitions as an NFA–ϵ.

Since ϵ represents the empty word over any alphabet, we extend δ to be a function from $Q \times (\Sigma \cup \{\epsilon\})$ to 2^Q. We wish for our definition of $L(M)$ to remain the same, that is, $L(M) = \{w : \delta(q_0, w) \cap F \neq \emptyset\}$. So, we must extend our definition of δ^* accordingly (remembering that, if w_1 and w_2 are words, then $w_1 w_2 = w_1 \epsilon w_2$). So, intuitively, $\delta^*(q, w)$ is the set of states r such that there is a path that is labelled

w (possibly including some edges that are labelled ϵ) leading from q to r.

To be precise, we will define $\delta^*(q, w)$ by induction on $|w|$. First, $\delta^*(q, \epsilon)$ is defined as the set of all states that can be reached from q by following a (graph–theoretic) path that contains *only* ϵ–transitions (that is, all of the edges in the path are labelled ϵ). Note that it is always true that $q \in \delta^*(q, \epsilon)$, since you can reach q from q by following the "empty path". We will proceed by induction. For any word $x \in \Sigma^*$ and and letter $a \in \Sigma$, we will define $\delta^*(q, wa)$ to be $\bigcup_{r \in \delta^*(q,w)} \delta^*(\delta(r, a), \epsilon)$.

At this point, it is important to note that an NFA–ϵ may be converted to an NFA without ϵ–transitions with no increase in the number of states. Formally, an NFA–ϵ $M = (Q, \Sigma, \delta, q_0, F)$ may be converted to an equivalent NFA $M' = (Q, \Sigma, \delta', q_0, F')$ as follows: For any $(q, a) \in Q \times \Sigma$, define $\delta'(q, a)$ to be $\delta^*(q, a)$. Also, if $\delta^*(q_0, \epsilon)$ contains any final states, then $F' = F \cup \{q_0\}$. Otherwise, $F' = F$. For a formal proof that this construction is correct, see Hopcroft & Ullman [12, Theorem 2.2]. Although their notation is slightly different, the concept is the same. For our purposes, the important point is that $|M'| = |M|$. This is clearly true, since they use the same set of states Q.

For every state q in a finite automaton $(Q, \Sigma, \delta, q_0, F)$, let $\mathcal{F}(q)$ denote the set of words $\{w : \delta^*(q, w) \in F\}$ (or, if the finite automaton is nondeterministic, $\mathcal{F}(q) = \{w : \delta^*(q, w) \cap F \neq \emptyset\}$). Furthermore, let $\mathcal{H}(q)$ denote the set of words $\{w : \delta^*(q_0, w) = q\}$ (or, in the nondeterministic case, $\{w : q \in \delta^*(q_0, w)\}$).

We are now ready to define another measure of descriptional complexity of a regular language L: its *nondeterministic state complexity* (denoted by $\mathrm{nsc}(L)$). The nondeterministic state complexity of a language is the number of states in a minimal

NFA for L. As shown above, an NFA–ϵ may be converted to an equivalent NFA (without ϵ–transitions) with no increase in the number of states. Therefore, when making arguments regarding nondeterministic state complexity, we may allow our NFAs to contain ϵ–transitions.

4.2 Deterministic and Nondeterministic State Complexity

Since a DFA is just a special case of an NFA, it is clear that if L is a regular language, then $\mathrm{sc}(L) \geq \mathrm{nsc}(L)$. In fact, we have a tight bound on how large $\mathrm{sc}(L)$ can be in comparison to $\mathrm{nsc}(L)$. The bound follows directly from the subset construction, which is originally due to Rabin & Scott [20]. The tightness of this bound was originally shown by Moore [19], although we cite a simpler construction in our proof.

Theorem 7 *If L is a regular language with nondeterministic state complexity n, then $\mathrm{sc}(L) \leq 2^n$. Furthermore, this bound is tight.*

Proof:

Suppose that M is a minimal NFA for L. Then we can use the standard subset construction (see Hopcroft & Ullman [12, Theorem 2.1] for a detailed description) to construct a DFA with at most 2^n states that accepts L. This establishes the bound. To see that the bound is tight, for each n, let L_n be the regular language specified by the regular expression $(a + (ab^*)^{n-1}a)^*$ (see Section 5.1 for a formal definition of regular expressions and the languages that they specify). As shown by Leung [17], L_n can be accepted by an NFA with n states, but cannot be accepted by a DFA with less than 2^n states. □

4.2.1 Unary Languages

The bound in Theorem 7 is only tight for regular languages over non–unary alphabets. For unary languages, we can get a better bound. Chrobak [5] noted that the following function is crucial to the study of minimal unary NFAs:

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

$F(n)$ represents the maximum order of an element in the symmetric group S_n (see any group theory text, for example Dummit & Foote [7], for a detailed discussion of these group–theoretic concepts). The exact value of $F(n)$ for any given n is related to the distribution of the prime numbers, and as a result we cannot expect to express $F(n)$ succinctly in terms of n. However, we have asymptotic results. Landau [14, 15] showed that

$$\lim_{n\to\infty} \frac{\log F(n)}{\sqrt{n \log n}} = 1.$$

(Unless otherwise stated, "log" denotes the natural logarithm.) Szalay [23] showed the following approximation to be true:

$$\log F(n) = \sqrt{n(\log n + \log\log n + \delta(n))},$$

where, asymptotically,

$$\delta(n) = -1 + o(1).$$

From these results, we can deduce:

Lemma 8 *$F(n) \in O(e^{\sqrt{n \log n}(1+o(1))})$,*

and

Lemma 9 *$F(n) \in \Omega(e^{\sqrt{n \log n}})$.*

However, we cannot deduce, as Chrobak [5] claimed, that $F(n) \in O(e^{\sqrt{n \log n}})$ (this error seems to have been caused by a typographical error in the statement of Szalay's approximation).

Now that we have good asymptotic bounds on $F(n)$, we are ready for the main theorems of this section:

Theorem 10 (Chrobak [5]) *If L is a unary regular language with nondeterministic state complexity n, then $\mathrm{sc}(L) \in O(F(n))$.*

And, in fact, this bound is asymptotically tight:

Theorem 11 (Chrobak [5]) *For each n there is a unary language L_n with $\mathrm{nsc}(L_n) \leq n$ and $\mathrm{sc}(L) = F(n-1)$.*

4.3 A Lower Bound

The problem of minimizing an NFA (that is, given an NFA, the problem of finding an equivalent minimal NFA) is known to be PSPACE–hard [13]. Furthermore, many non–isomorphic NFAs may be minimal for a particular language. As a result, we do not have a nondeterministic counterpart to the Myhill–Nerode theorem, which nicely classifies minimal DFAs. However, we have the following lower bound:

Lemma 12 (Birget [2]) *Let L be a regular language over the alphabet Σ, and suppose there exist n pairs of words $(x_1, w_1), (x_2, w_2), \dots, (x_n, w_n)$ such that:*

- *For all i with $1 \le i \le n$, $x_i w_i$ is in L, and*
- *For all i, j with $1 \le i, j \le n$ and $i \ne j$, at least one of $x_j w_i$ and $x_i w_j$ is not in L.*

Then $\text{nsc}(L) \ge n$.

Proof:

Suppose $M = (Q, \Sigma, \delta, q_0, F)$ is an NFA that accepts L. Then, we will associate a state q_i with each pair (x_i, w_i) as follows: we will choose q_i in such a way that $q_i \in \delta^*(q_0, x_i)$, and $\delta^*(q_i, w_i) \cap F \ne \emptyset$. Note that such a q_i must exist, since $x_i w_i \in L$.

Now, suppose that for $i \ne j$, $q_i = q_j$. Then $x_i w_j$ and $x_j w_i$ are both in L, (due to the way that q_i and q_j were chosen) which is a contradiction. Thus, $\{q_1, q_2, \dots, q_n\}$ is a set of n distinct states, and so $|M| \ge n$ as desired. □

4.4 Operations on Regular Languages

For many operations that preserve regularity, we have simple tight bounds on the increase in nondeterministic state complexity. Most of these results are original to this thesis. However, many of them were subsequently and independently published by Holzer and Kutrib [10, 11]. Those results that are not original to this thesis, but were first proven by Holzer and Kutrib are attributed accordingly. Similarly, a note is made where a result was original to this thesis, and subsequently published by Holzer and Kutrib.

4.4.1 Union

Theorem 13 *Suppose L_1 and L_2 are regular languages with nondeterministic state complexity m and n respectively. Then* $\mathrm{nsc}(L_1 \cup L_2) \leq m + n + 1$*, and this bound is tight.*

Proof:

Suppose that $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ are minimal NFAs for L_1 and L_2 respectively. Also assume that Q_1 and Q_2 are disjoint, and do not contain the state "q_0".

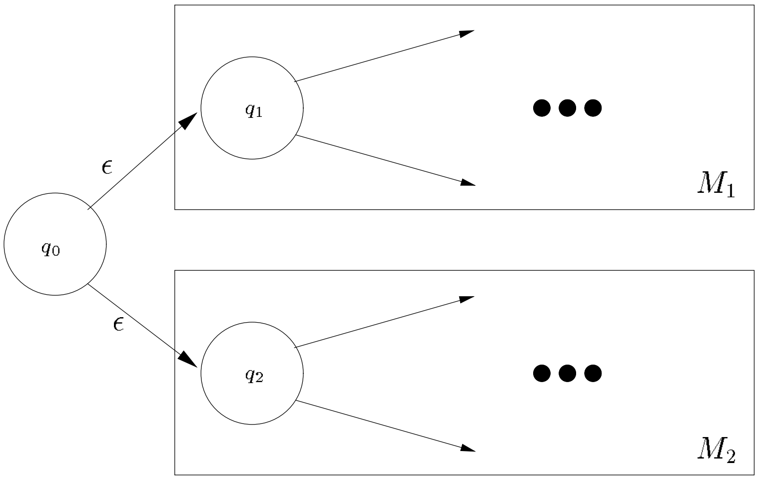

Figure 4.3: The construction that gives the bound on $\mathrm{nsc}(L_1 \cup L_2)$.

To establish the bound, we will use the following construction, as shown in Figure 4.3. Define $Q = Q_1 \cup Q_2 \cup \{q_0\}$ and $F = F_1 \cup F_2$. For each $a \in \Sigma$, define δ as follows:

$$\delta(q, a) = \begin{cases} \delta_1(q, a), & \text{when } q \in Q_1; \\ \delta_2(q, a), & \text{when } q \in Q_2; \\ \emptyset & \text{when } q = q_0. \end{cases}$$

Also, let $\delta(q_0, \epsilon) = \{q_1, q_2\}$. So, $M = (Q, \Sigma, \delta, q_0, F)$ is an NFA–ϵ with $m + n + 1$ states that accepts $L_1 \cup L_2$. This establishes the bound.

To see that the bound is tight, choose n and m arbitrarily, and let $L_1 = (a^n)^*$ and let $L_2 = (b^m)^*$. Clearly $\mathrm{nsc}(L_1) = n$ and $\mathrm{nsc}(L_2) = m$. Let $L = L_m \cup L_m$. So $L = (a^n)^* + (b^m)^*$. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a minimal NFA for L. For simplicity, assume that M contains no ϵ–transitions. From the bound that was proven above, $|M|$ can be no more than $m + n + 1$. We will show that M requires $m + n + 1$ states.

Recall the definitions of $\mathcal{F}(q)$ and $\mathcal{H}(q)$ from Section 4.1. Note that M is minimal, so, for any $q \in Q$, neither $\mathcal{F}(q)$ nor $\mathcal{H}(q)$ may be empty (for, if one of those sets were empty, removing q from M would result in a smaller equivalent NFA for L, which contradicts the minimality of M). Now, $\mathcal{F}(q_0) = L$ and so, in particular, $a^n \in \mathcal{F}(q_0)$ and $b^m \in \mathcal{F}(q_0)$. But, no word $w \in L$ may contain both a's and b's. Therefore, if $w \neq \epsilon$, then $q_0 \notin \delta^*(q_0, w)$. So, q_0 may not be part of any cycle in M. However, L is infinite so M must contain cycles.

Since no word w in L contains both a's and b's, and there are infinitely many words in L that contain a's, and infinitely many that contain b's, M must contain at least two disjoint cycles: one for inputs that contain a's, and one for inputs that contain b's. If $a^i \in L$, then $a^{i+j} \notin L$ for any $0 < j < n$. Thus, there must be a cycle that is accessible on inputs that contain a's which is of size at least n. Similarly, there must be a cycle of size m which is accessible on inputs that contain b's. Since these cycles must be disjoint, and the start state may not be a part of any cycle, $|M| \geq m + n + 1$ as desired. □

This result was also proven independently by Holzer and Kutrib [10]. Holzer and Kutrib [11, Theorem 4] also showed that this bound is achievable in the unary case, so long as m is not a divisor or multiple of n.

4.4.2 Intersection

Theorem 14 *Suppose L_1 and L_2 are regular languages with nondeterministic state complexity m and n respectively. Then $\mathrm{nsc}(L_1 \cap L_2) \leq mn$, and this bound is tight.*

Proof:

Suppose that $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ are minimal NFAs for L_1 and L_2 respectively. Also assume that Q_1 and Q_2 are disjoint, and do not contain the state "q_0".

Obviously, a word is in $L_1 \cap L_2$ if and only if it is in both L_1 and L_2. So, the idea behind our construction is to simulate M_1 and M_2 simultaneously on an input word x, and accept x if and only if both M_1 and M_2 would have accepted. Define $Q = Q_1 \times Q_2$ and $F = F_1 \times F_2$. Also, for $[q, r] \in Q_1 \times Q_2$ and $a \in \Sigma$, we define $\delta([q, r], a) = \delta_1(q, a) \times \delta_2(r, a)$. Finally, let $M = (Q, \Sigma, \delta, [q_1, q_2], F)$. Our new machine M simultaneously "simulates" both M_1 and M_2, since, for any word x, $\delta^*([q_1, q_2], x) = \delta_1^*(q_1, x) \times \delta_2^*(q_2, x)$. Thus, M accepts $L_1 \cap L_2$ and uses mn states, which establishes the bound.

To see that the bound is tight, consider the following example: Choose m and n arbitrarily, and fix $\Sigma = \{a, b\}$. Then, let $L_1 = \{w \in \Sigma^* \ : \ |w|_a \geq n - 1\}$ and let $L_2 = \{w \in \Sigma^* \ : \ |w|_b \geq m - 1\}$. Clearly, the nondeterministic state complexities of L_1 and L_2 are m and n respectively. To see that any NFA that accepts $L_1 \cap L_2$ requires at least mn states, we will use Lemma 12. Our set of pairs of words will be $W = \{(x_{i,j}, w_{i,j}) \ : \ 0 \leq i \leq n - 1; \ 0 \leq j \leq m - 1\}$, where each $x_{i,j} = a^i b^j$ and each $w_{i,j} = a^{n-i-1} b^{m-j-1}$. For any pair $(x_{i,j}, w_{i,j}) \in W$, it is clear that $x_{i,j} w_{i,j} \in L_1 \cap L_2$ since $|a^i b^j a^{n-i-1} b^{m-j-1}|_a = n - 1$ and $|a^i b^j a^{n-i-1} b^{m-j-1}|_b = m - 1$.

However, if we pick two *different* pairs $(x_{i,j}, w_{i,j})$ and $(x_{k,l}, w_{k,l})$ from W, then either $i < k$, $k < i$, $j < l$, or $l < j$ (since, if all of these statements are false, then $i = k$ and $j = l$ and the pairs that we chose were not different). We will examine each of the 4 cases:

Case 1: $i < k$, and so $x_{i,j} w_{k,l} \notin L_1$ (and therefore also not in $L_1 \cap L_2$).

Case 2: $k < i$, and so $x_{k,l} w_{i,j} \notin L_1$ (and therefore also not in $L_1 \cap L_2$).

Case 3: $j < l$, and so $x_{i,j} w_{k,l} \notin L_2$ (and therefore also not in $L_1 \cap L_2$).

Case 4: $l < j$, and so $x_{k,l}w_{i,j} \notin L_2$ (and therefore also not in $L_1 \cap L_2$).

Therefore, any NFA accepting $L_1 \cap L_2$ must have at least $|W| = mn$ states.

□

This result was also shown independently by Holzer and Kutrib [10].

The bound is also achievable in the unary case. For example, suppose that m and n are coprime. Then the languages $L_1 = (a^n)^*$ and $L_2 = (a^m)^*$ have nondeterministic state complexities m and n respectively. However, a word a^k is in $L_1 \cap L_2$ if and only if k is a multiple of m and n. Since m and n are coprime, $L_1 \cap L_2 = (a^{mn})^*$, and the nondeterministic state complexity of $L_1 \cap L_2$ is exactly mn. This was also shown independently by Holzer and Kutrib [11].

4.4.3 Concatenation

Theorem 15 *Suppose L_1 and L_2 are regular languages with nondeterministic state complexity m and n respectively. Then $\mathrm{nsc}(L_1L_2) \leq m + n$, and this bound is tight.*

Proof:

Suppose that $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ are minimal NFAs for L_1 and L_2 respectively. Also, assume that Q_1 and Q_2 are disjoint, and do not contain the state "q_0". To show that the bound holds, we use the following construction, shown in Figure 4.4: Let $Q = (Q_1 \cup Q_2)$. For each

$a \in \Sigma$, define δ as follows:

$$\delta(q, a) = \begin{cases} \delta_1(q, a), & \text{when } q \in Q_1; \\ \delta_2(q, a), & \text{when } q \in Q_2. \end{cases}$$

Additionally, for each $q \in F_1$, let $\delta(q, \epsilon) = \{q_2\}$. So $M = (Q, \Sigma, \delta, q_1, F_2)$ is an NFA–ϵ that uses $m + n$ states and accepts $L_1 L_2$, as desired.

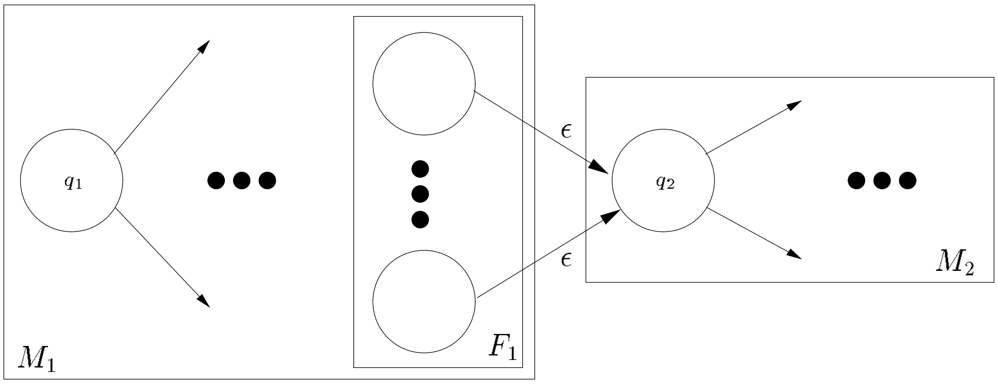

Figure 4.4: The construction that gives us the bound on $\text{nsc}(L_1 L_2)$.

To see that the bound is tight, we will use the two–letter alphabet $\Sigma = \{a, b\}$. Choose m and n arbitrarily. Let $L_1 = (a^m)^*$ and let $L_2 = (b^n)^*$. Then, clearly L_1 and L_2 have nondeterministic state complexities m and n respectively.

Consider any NFA $M = (Q, \Sigma, \delta, q_0, F)$ that accepts $L_1 L_2$. It is sufficient to show that M must have at least $m + n$ states, and we will do this using Lemma 12. For each $1 \leq i \leq m$, let $w_i = a^i$ and let $x_i = a^{2m-i}$. For each $1 \leq j \leq n$, let $w_{m+j} = b^j$ and let $x_{m+j} = b^{2n-j}$.

Each word $x_i y_i$ is either a^{2m} or b^{2n}, and thus clearly in $L_1 L_2$. However, if $i \neq j$, then there are four cases to consider:

Case 1: $1 \leq i, j \leq m$. Then $w_i x_j = a^i a^{2m-j} = a^{2m+i-j} \notin L_1 L_2$ (since $2m + i - j$ is not divisible by m).

Case 2: $1 \leq i \leq m < j \leq m + n$. Then $w_j x_i = b^{j-m} a^{2m-i} \notin L_1 L_2$ (since it is not in the form $a^* b^*$).

Case 3: $1 \leq j \leq m < i \leq m + n$. Then $w_i x_j = b^{i-m} a^{2m-j} \notin L_1 L_2$ (since it is not in the form $a^* b^*$).

Case 4: $m < i, j \leq m + n$. Then $w_i x_j = b^{i-m} b^{2n-j+m} = b^{2n+i-j} \notin L_1 L_2$ (since $2n + i - j$ is not divisible by n).

Therefore, by Lemma 12, M must have at least $m + n$ states, and the bound is tight. $\square$

This result was also shown independently by Holzer and Kutrib [10].

In the unary case, using the alphabet $\{a\}$, it is trivial to find languages L_m and L_n for any m and n such that $\text{nsc}(L_m) = m$, $\text{nsc}(L_n) = n$, and $\text{nsc}(L_m L_n) = m + n - 1$. Simply let $L_i = \{a^{i-1}\}$ for all i. Since $\text{nsc}(L_i) = i$ for all i, and $L_i L_j = L_{i+j-1}$, the result follows. This was shown independently by Holzer and Kutrib [11]. However, this leaves a gap (of size 1) between our upper and lower bounds. Therefore, we have the following open problem:

Open Problem 1 *Do there exist integers m and n, and unary languages L_m and L_n such that:*

- $\text{nsc}(L_m) = m$,
- $\text{nsc}(L_n) = n$, *and*
- $\text{nsc}(L_m L_n) = m + n$*?*

4.4.4 Kleene Closure

Theorem 16 *Suppose L is a regular language with nondeterministic state complexity n. Then* $\text{nsc}(L^*) \leq n + 1$.

Proof:

In order to see that the bound is correct, consider the following construction: Suppose that M is an NFA that accepts L. Then, modify M to create an NFA–ϵ M' as follows: simply add ϵ–transitions from each final state of M to the start state. Also, add a new final state q' with an epsilon transition from q to q' (this ensures that M' accepts the empty string). Then M' is the required NFA–ϵ that accepts L^*. □

In fact, as shown by Holzer and Kutrib [11], this bound is tight, even in the unary case.

Theorem 17 (Holzer & Kutrib, [11]) *For each $n \geq 2$, the unary language $L_n = \left\{a^k : k \equiv n - 1 \pmod{n}\right\}$ has nondeterministic state complexity n, but* $\text{nsc}(L_n^*) = n + 1$.

4.4.5 Reversal

Theorem 18 *Suppose L is a regular language with nondeterministic state complexity n. Then* $\mathrm{nsc}(L^R) \leq m + 1$.

Proof:

Suppose $M = (Q, \Sigma, \delta, q_0, F)$ is a minimal NFA for L. Intuitively, we can modify M by adding an additional state q_{new}, reversing each transition in M, and adding ϵ–transitions from q_{new} to each final state. Then, let the "old" start state be the only final state, and let q_{new} be the start state.

Formally, let $Q' = Q \cup \{q_{new}\}$ and for each $[q, a] \in Q \times \Sigma$ let $\delta'(q, a) = \{q' : q \in \delta(q', a)\}$. Furthermore, let $\delta'(q_{new}, \epsilon) = F$ and for each $a \in \Sigma$ let $\delta'(q_{new}, a) = \emptyset$. Then, $M' = (Q', \Sigma, \delta', q_{new}, \{q_0\})$ is an NFA–ϵ with $m + 1$ states that accepts L_1^R. This establishes the bound. □

This result was also shown independently by Holzer and Kutrib [10]. Additionally, they showed that the bound is tight for languages over an alphabet of size greater than or equal to 3:

Theorem 19 (Holzer & Kutrib [10]) *The bound in Theorem 18 is tight. Specifically, for every $k \geq 1$, if $L_k = a^k(a^{k+1})^*(b^* + c^*)$ then* $\mathrm{nsc}(L_k) = k + 3$, *and* $\mathrm{nsc}(L_k^R) = k + 4$.

In fact, a small modification of the above example shows that the bound is tight for languages over a two–letter alphabet.

Corollary 20 *For each $k \geq 1$, let $L_k = a^k(a^{k+1})^*((bb)^* + (bbb)^*)$. Then $\mathrm{nsc}(L_k) = k + 6$ and $\mathrm{nsc}(L_k^R) = k + 7$.*

In the unary case, this problem becomes trivial. If w is a unary word, then $w = w^R$. So, if L is a unary language, then $L = L^R$ and so $\mathrm{nsc}(L) = \mathrm{nsc}(L^R)$.

4.4.6 Complementation

Note that the subset construction (see Hopcroft & Ullman [12]) allows us to convert an NFA of size n to an equivalent DFA of size 2^n. Also, a DFA $M = (Q, \Sigma, \delta, q_0, F)$ may be complemented by replacing F with $Q - F$. Since a DFA is just a special case of an NFA, the following is a trivial upper bound:

Theorem 21 *Let L be a regular language with nondeterministic state complexity n. Then $\mathrm{nsc}(\overline{L}) \leq 2^n$.*

Although this bound seems quite large, it turns out that it is the best possible. Sakoda & Sipser [22] showed that for each n, there is a language L_n such that $\mathrm{nsc}(L_n) = n$ and $\mathrm{nsc}(\overline{L_n}) = 2^n$. However, they did not use a fixed alphabet to accomplish this. In fact, each L_n was over a different alphabet Σ_n of size 2^{n^2}.

Birget [3] published a result which claimed that, for each n, there exists a regular language L_n over an alphabet of size 3 such that L_n is accepted by an NFA of size n, but the smallest NFA that accepts $\overline{L_n}$ has size 2^n. However, this result was incorrect. Birget [4] corrected this error. The following theorem contains the corrected result:

Theorem 22 (Birget [3, 4]) *There is a fixed alphabet Σ of size 4 such that, for each n, there is a regular language L_n with $\mathrm{nsc}(L_n) = n$ and $\mathrm{nsc}(\overline{L_n}) = 2^n$.*

Proof:

Our alphabet will contain four letters: α, β, γ, and γ^{-1}. Intuitively, α, β, and γ represent functions from $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$ to $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$, as follows:

- α is the permutation $(\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n})$,
- β is the transposition $(\mathbf{1}, \mathbf{2})$, and
- γ maps both $\mathbf{1}$ and $\mathbf{2}$ to $\mathbf{1}$, and acts as the identity function on everything else.

Taken together, the set $\{\alpha, \beta, \gamma\}$ is a generating set for all total functions from $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$ to $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$. That is, for any total function $f : \{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\} \rightarrow \{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$, $f = f_1 \cdot f_2 \cdot \ldots \cdot f_k$ for some k, with each $f_i \in \{\alpha, \beta, \gamma\}$.

Note that the inverses of α and β are functions themselves, and are thus generated by $\{\alpha, \beta, \gamma\}$. In fact, $\alpha^{-1} = \alpha^{n-1}$, and $\beta^{-1} = \beta$. The inverse of γ is not, however, a function, and so it is included as the fourth letter in Σ. Furthermore, we may treat α, β, γ, and γ^{-1} as functions that map $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$ to $2^{\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}}$. (Note that the range of α, β and γ will be the set of singletons in $2^{\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}}$).

Suppose $f \in \Sigma^*$. Then $f = f_1 f_2 \ldots f_k$ with each $f_i \in \Sigma$. So we can allow f to represent the function $f_k \cdot f_{k-1} \cdot \ldots \cdot f_1$ which maps $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$ to $2^{\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}}$.

We are now ready to define L_n over the alphabet Σ. A word (function)

f is in L_n if and only if $\mathbf{2} \in f(\mathbf{1})$. First, we will construct an NFA $M = (Q, \Sigma, \delta, q_0, F)$ of size n which accepts L_n. First, we will set $Q = \{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$. For each $i \in Q$, and each $f \in \Sigma$, set $\delta(i, f) = f(i)$. Finally, $\mathbf{1}$ will be the start state and $\mathbf{2}$ will be the lone final state. It is easy to see that there is a path from $\mathbf{1}$ to $\mathbf{2}$ in M on input f if and only if $\mathbf{2} \in f(\mathbf{1})$. So M is an n–state NFA which accepts L_n.

To see that any NFA accepting $\overline{L_n}$ requires at least 2^n states, we will use Lemma 12. For each subset S of $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$, we will choose a pair of words (f_S, h_S) from Σ^* so that the following conditions hold:

- $f_S(\mathbf{1}) = S$,
- for each $i \in S$, $h_S(i) = \{\mathbf{2}\}$, and
- for each $i \notin S$, $h_S(i) = \{\mathbf{1}\}$.

To see that such f_S and h_S exist for each S, note that h_S is simply a total function whose domain is $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$ and whose range is $\{\mathbf{1}, \mathbf{2}\}$. Therefore, it is generated by $\{\alpha, \beta, \gamma\}$ and is in Σ^*.

Although f_S is not a function, it is the inverse of a function. In fact, setting f_S to be the inverse of h_S yields the desired property that $f_S(\mathbf{1}) = S$. Therefore, f_S is generated by $\{\alpha^{-1}, \beta^{-1}, \gamma^{-1}\}$, and thus is in Σ^* (Recall that α^{-1} and β^{-1} are generated by α and β respectively).

Since there are 2^n different subsets S of $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$, it is only left to show that the words (f_S, h_S) satisfy the requirements of Lemma 12. First, note that $h_S(f_S(\mathbf{1})) = \{\mathbf{1}\}$ and so $f_S h_S \in \overline{L_n}$. Now, consider two different subsets S and Z of $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$, and let i be in the symmetric difference of S and Z. If $i \in S \cap \overline{Z}$, then $i \in f_S(\mathbf{1})$ and $h_Z(i) = \{\mathbf{2}\}$. So $2 \in h_Z(f_S(\mathbf{1}))$ and so $f_S h_Z \in L_n$. Otherwise, $i \in Z \cap \overline{S}$, so $f_Z h_S \in L_n$. Thus, we see that at least one of $f_S h_Z$ and $f_Z h_S$ are in L_n, and thus not in $\overline{L_n}$, and so any NFA accepting $\overline{L_n}$, must have at least 2^n states, as desired. □

The question of whether we can achieve the same result (an increase from n to 2^n) over a smaller alphabet is currently open. However, using a two–letter alphabet, we can achieve an increase of $O(n)$ to 2^n in nondeterministic state complexity when complementing a regular language.

Theorem 23 *For each n, there exists a language L_n over a two–letter alphabet such that* $\text{nsc}(L_n) = O(n)$ *and* $\text{nsc}(\overline{L_n}) \geq 2^n$*.*

Proof:

Let $L_n = (a + b)^*(a(a + b)^{n-1}a + b(a + b)^{n-1}b)(a + b)^*$. So a word x is in L_n if and only if $x[i] = x[i + n]$ for some i. Since we have given a regular expression of size $O(n)$ for L_n, it must be that $\text{nsc}(L_n) \in O(n)$ (see Theorem 27).

To show that $\text{nsc}(\overline{L_n}) \geq 2^n$, we use Lemma 12. For each i with $1 \leq i \leq 2^n$, let w_i represent the ith word (in lexicographical order) of length n over the alphabet $\{a, b\}$. Also, let x_i represent the complement of w_i, that is, the image of w_i under the mapping $\{a \mapsto b;\ b \mapsto a\}$. Clearly $x_i w_i \in \overline{L_n}$ for each

i, since $w_i[k] \neq x_i[k] = (w_i x_i)[k+n]$ for all k with $1 \leq k \leq n$. Furthermore, if $i \neq j$, then $w_i[k] = x_i[k]$ for some k, that is, $w_i[k] = (w_i x_i)[k+n]$. Thus, by Lemma 12, $\mathrm{nsc}(\overline{L_n}) \geq 2^n$, as desired. □

Holzer and Kutrib [10] showed a stronger version of this result. They showed that for each $n > 2$ there exists an n–state NFA M_n over a two–letter alphabet such that any NFA which accepts the complement of $L(M_n)$ requires at least 2^{n-2} states.

In the unary case, we have a better bound. Recall Theorem 10, which states that if M is a unary NFA of size n accepting a language L, then the minimal DFA for L has size no more than $O(F(n))$. (See Section 4.2.1 for a discussion of $F(n)$.) Therefore, since DFAs can be complemented with no increase in size, we have the following bound:

Theorem 24 *If L is a unary regular language with nondeterministic state complexity n, then* $\mathrm{nsc}(\overline{L}) = O(F(n))$.

In fact, Holzer and Kutrib [11] show that this bound is, in fact, the best that we can do.

Theorem 25 (Holzer & Kutrib [10]) *For any $n > 1$ there exists a unary n–state NFA M such that the nondeterministic state complexity of the complement of $L(M)$ is $F(n-1)$.*

4.4.7 Quotients

In the unary case, we have tight bounds on the state complexities of the left and right quotients of a language. In the general case, we have a tight bound for the right quotient.

Theorem 26 *Let L be a regular language over Σ, and let w be a word in Σ^*. Then:*

(a) $\mathrm{nsc}(Lw^{-1}) \leq \mathrm{nsc}(L)$.

Also, if Σ is a unary alphabet, then

(b) $\mathrm{nsc}(w^{-1}L) \leq \mathrm{nsc}(L)$,

Otherwise,

(c) $\mathrm{nsc}(w^{-1}L) \leq \mathrm{nsc}(L) + 1$.

Furthermore, the bounds in (a) and (b) are tight.

Proof:

Suppose $\mathrm{nsc}(L) = n$. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a minimal NFA for L. Let $F' = \{q \in Q : \delta^*(q, w) \cap F \neq \emptyset\}$. That is, F' is the set of all states q from which there is a path labelled w that leads to some final state. Then $M' = (Q, \Sigma, \delta, q_0, F')$ accepts Lw^{-1}, and $|M'| = |M| = n$. This establishes the bound in (a). Since $Lw^{-1} = w^{-1}L$ in the unary case, this also establishes the bound in (b).

To establish the bound in part (c), let $Q' = Q \cup \{q'\}$ for some new state $q' \notin Q$. Define δ' as follows:

$$\begin{aligned}
\delta'(q, a) &= \delta(q, a) \; \forall q \in Q, \; a \in \Sigma; \\
\delta'(q', a) &= \emptyset \; \forall a \in \Sigma; \\
\delta'(q', \epsilon) &= \delta^*(q_0, w).
\end{aligned}$$

Then $M' = (Q', \Sigma, \delta', q', F)$ is an NFA–ϵ which accepts $w^{-1}L$, and $|M'| = |M| + 1 = n + 1$. This establishes the bound in (c).

To see that the bounds in (a) and (b) are tight (even in the unary case), simply let $L_n = \{x : |x| \equiv 0 \pmod{n}\}$. Then, clearly, L_n has nondeterministic state complexity n, and $L_n w^{-1} = w^{-1} L_n = \{x : |x| \equiv -k \pmod{n}\}$ (where $|w| = k$). Thus, $L_n w^{-1}$ also has nondeterministic state complexity n, and the bounds are tight. $\square$

However, we do not know if the bound in part (c) of the above theorem is tight.

Open Problem 2 *Is the bound in Theorem 26, part (c) tight? That is, does there exist an alphabet Σ, a regular language $L \subseteq \Sigma^*$, and a word $w \in \Sigma^*$ such that* $\mathrm{nsc}(w^{-1}L) = \mathrm{nsc}(L) + 1$*?*

Chapter 5

Regular Expression Size

In this section we will look at another, and perhaps more intuitive, way of specifying regular languages: regular expressions. We will study the size of the smallest regular expression that specifies a regular language. We will examine how this complexity measure compares to state complexity (both deterministic and nondeterministic), and what effect the application of certain regularity-preserving operations has on minimal regular expression size.

5.1 Definitions

A *regular expression* is a standard way of specifying a regular language. Regular expressions can be defined recursively. Suppose we wish to specify a regular language over Σ. Then, $\emptyset$ and ϵ are regular expressions, specifying the regular languages $\emptyset$ and $\{\epsilon\}$ respectively. Furthermore, for any $a \in \Sigma$, a is a regular expression specifying the language $\{a\}$. Now, suppose that r_1 and r_2 are regular expressions, specifying the regular languages L_1 and L_2 respectively. Then, $(r_1) + (r_2)$ is a regular expression that specifies $L_1 \cup L_2$. Also, $(r_1)(r_2)$ is a regular expression that specifies $L_1 L_2$. Finally, $(r_1)^*$ is a regular expression that specifies L_1^*. Note that the parentheses may be omitted if they are superfluous. For example, to specify the language containing the single word ab, we may simply use the regular expression "ab" rather than "$(a)(b)$" (although both would be acceptable). In the absence of parentheses, our "order of operations" has Kleene closure (star) binding most closely, followed by concatenation, followed by the union operator. Thus, the regular expression "$ab + cd^*$" is equivalent to "$(ab) + (c(d^*))$. We will refer to the language specified by the regular expression r as $L(r)$. Furthermore, as shown in Hopcroft & Ullman [12, Theorem 2.3], every regular language can be specified by a regular expression.

There are many ways to measure the complexity of a regular expression. For example, Ehrenfeucht and Zeiger [8] studied the *size*, *star height*, *width*, and *width* of regular expressions. We will use the *size* of, or number of alphabetic symbols in, a regular expression r (denoted by $|r|$) to measure its complexity. For example, the regular expressions $abab$ and $(a^*ba)^*b$ both have size 4. This measure was chosen because it is intuitively similar to the state complexity of a finite automaton.

A regular expression will be referred to as *minimal* if it is minimal with respect to size, that is, if there is no regular expression of smaller size that specifies the same language. The size of a regular language L, denoted by $\text{size}(L)$, refers to the size of a minimal regular expression for L.

5.2 Regular Expressions and Finite Automata

Regular expressions give us a fundamentally different way of representing regular languages than (deterministic or nondeterministic) finite automata. Therefore, we will consider the problem of converting from a regular expression to a finite automaton, and vice versa.

One of the reasons that we choose to measure the complexity of a regular expression by its size (as opposed to some other measure) is its similarity to state complexity, particularly, nondeterministic state complexity. Intuitively, a regular expression is easily transformed into an NFA. In fact, nondeterministic state complexity gives us a lower bound on regular expression size.

Theorem 27 (Leiss [16]) *Suppose that L is a regular language over an alphabet Σ. Then $\mathrm{nsc}(L) \leq \mathrm{size}(L) + 1$, and this bound is tight, even in the unary case.*

Proof:

First, define an NFA to be *non–returning* if there are no transitions leading to the start state. We will prove inductively that, for every regular expression r of size n, there there is a non–returning NFA–ϵ M_r of size $n+1$ that accepts $L(r)$. Since M_r can be converted to an NFA with no increase in the number of states, this is sufficient to prove our claim.

We will proceed by induction on the number of operations (union, concatenation, and Kleene closure) used in the construction of r. In the base case, there are no operations used, and so $r = \emptyset$ or $r = a$ for some $a \in \Sigma$. If $r = \emptyset$, then $|r| = 0$ and M_r is simply a single state (which is designated as the start state) with no transitions, and no final states. So, $|M_r| = 1 = |r| + 1$ as desired. If $r = a$ for some $a \in \Sigma$, then $|r| = 1$ and M_r consists of two states: a start state q_0 and a final state q_1, with a single transition from q_0 to q_1 on input a. Clearly, $L(M_r) = \{a\} = L(r)$ and $|M_r| = 2 = |r| + 1$ as desired.

For the inductive step, we must consider the operations of union, concatenation, and Kleene closure. First, consider the union operation. Suppose that $r = s + t$. So $|r| = |s| + |t|$. Suppose that $M_s = (Q_s, \Sigma, \delta_s, q_s, F_s)$ and $M_t = (Q_t, \Sigma, \delta_t, q_t, F_t)$. We can assume without loss of generality neither Q_s nor Q_t contain a state labelled q_0, and also that Q_s and Q_t are disjoint. Since M_s and M_t are non–returning, there are no transitions leading to q_s or q_t, so we can construct M_r as follows: Basically, we want to perform the usual union construction, except that we can "merge" the two start states (since our original NFAs are non–returning). Formally, define $Q = (Q_s \cup Q_t \cup \{q_0\}) - \{q_s, q_t\}$ and $F = (F_s \cup F_t)$. Also, for each $a \in \Sigma \cup \{\epsilon\}$, define δ as follows:

$$\delta(q, a) = \begin{cases} \delta_s(q, a), & \text{when } q \in Q_s; \\ \delta_t(q, a), & \text{when } q \in Q_t; \\ \delta_s(q_s, a) \cup \delta_t(q_t, a) & \text{when } q = q_0. \end{cases}$$

Then $M_r = (Q, \Sigma, \delta, q_0, F)$ accepts $L(s + t)$ and $|M_r| = |M_s| + |M_t| - 1 = |s| + 1 + |t| + 1 - 1 = |s + t| + 1$ as desired.

For concatenation, we can use a similar idea. If $r = st$ with $M_s = (Q_s, \Sigma, \delta_s, q_s, F_s)$ and $M_t = (Q_t, \Sigma, \delta_t, q_t, F_t)$, (with the usual assumption that Q_s and Q_t are disjoint) then we can perform the usual construction for concatenation, except that, since M_t is non–returning, there are no transitions leading into q_t and so we can remove that state. Formally, define $Q = Q_s \cup Q_t - \{q_t\}$ and $F = F_t$. For each $a \in \Sigma \cup \{\epsilon\}$, define δ as follows:

$$\delta(q,a) = \begin{cases} \delta_s(q,a), & \text{when } q \in Q_s - F_s; \\ \delta_s(q,a) \cup \delta_t(q_t,a), & \text{when } q \in F_s; \\ \delta_t(q,a), & \text{when } q \in Q_t; \end{cases}$$

Then $M_r = (Q, \Sigma, \delta, q_0, F)$ accepts $L(st)$ and $|M_r| = |M_s| + |M_t| - 1 = |s| + 1 + |t| + 1 - 1 = |st| + 1$ as desired.

Finally, for Kleene closure, suppose that $r = s^*$ with $M_s = (Q_s, \Sigma, \delta_s, q_s, F_s)$. Then for each $a \in \Sigma \cup \{\epsilon\}$, define δ as follows:

$$\delta(q,a) = \begin{cases} \delta_s(q,a), & \text{when } q \in Q_s - F_s; \\ \delta_s(q,a) \cup \delta_s(q_s,a), & \text{when } q \in F_s; \end{cases}$$

Then $M_r = (Q_s, \Sigma, \delta, q_s, F_s)$ is a non–returning NFA–ϵ that accepts $L(s^*)$ and $|M_r| = |M_s| = |s| + 1$ as desired.

Therefore, for any regular expression r, there is a non–returning NFA–ϵ M_r that accepts $L(r)$ and has size $|r| + 1$. Thus, there also exists an NFA of size $|r| + 1$ that accepts $L(r)$. Therefore, if L is a regular language, $\text{nsc}(L) \leq \text{size}(L) + 1$, as desired.

To see that the bound is tight, even in the unary case, simply choose some arbitrary n and let w be a word of length n. Then if $L = \{w\}$, it is clear that $\text{size}(L) = n$ and $\text{nsc}(L) = n + 1$. □

Corollary 28 *For any $n > 0$, the regular expression $(a^n)^*$ is minimal.*

Proof:

First, define L_n to be the language specified by the regular expression $(a^n)^*$. This result is similar to Holzer & Kutrib's result [11, Lemma 2] that the nondeterministic state complexity of L_n is n.

From the previous theorem, we know that if L_n is specified by a regular expression of size k, it is accepted by a non–returning NFA–ϵ of size $k + 1$. So, it is sufficient to show that any non–returning NFA–ϵ that accepts L_n requires $n + 1$ states.

Suppose $M = (Q, \{a\}, \delta, q_0, F)$ is a non–returning NFA–ϵ that accepts L_n. Since $a^n \in L_n$, there must be a sequence of states $\langle q_0, q_1, \ldots, q_n \rangle$ such that $q_n \in F$ and $q_{i+1} \in \delta^*(q_i, a)$ for each $0 \leq i \leq n - 1$. It is sufficient to show that these $n + 1$ states must all be unique.

First, note that since M is non–returning, it is not possible that $q_i = q_0$ for any $i > 0$. Now, choose i and j arbitrarily with $0 < i < j \leq n$. Note that $j = i + k$ for $0 < k < n$. Also, $q_i \in \delta^*(q_0, a^i)$ and $q_n \in \delta^*(q_j, a^{n-j})$. If $q_i = q_j$, then $q_n \in \delta^*(q_0, a^{i+n-j})$, that is, $a^{i+n-j} = a^{i+n-(i+k)} = a^{n-k} \in L_n$, which is a contradiction, since $0 < k < n$. Thus, $q_i \neq q_j$, and since i and j were chosen arbitrarily, we can conclude that each $q_i \in \langle q_0, q_1, \ldots, q_n \rangle$ is unique, and so M requires $n+1$ states. □

Although a regular expression can be transformed into an NFA with only an increase in size of an additive constant, the reverse is probably not true. Although we do not have a tight bound for this conversion, our best upper bound is exponential. This bound follows from an algorithm for converting a finite automaton to a regular expression, due to McNaughton and Yamada [18].

Theorem 29 (Ellul, Shallit, & Wang [9]) *If $M = (Q, \Sigma, \delta, q_0, F)$ is an finite automaton (deterministic or nondeterministic) of size n, and $|\Sigma| = k$, then there is a regular expression r of size at most $nk4^n$ that accepts $L(M)$.*

Although this is currently the best upper bound that is known, we do not have a matching lower bound. In fact, no non-trivial lower bounds are known. Ehrenfeucht and Zeiger [8] give examples of n–state finite automata that require exponentially large regular expressions, but their examples do not use a fixed alphabet. In fact, the alphabets that they use grow quadratically with n. Therefore, the following is currently an open problem:

Open Problem 3 *Does there exist a constant $c > 0$ such that, for arbitrarily large values of n, there exists a finite automaton M with $|M| = n$ and $\text{size}(L(M_n)) \geq 2^{cn}$?*

In the unary case, we can get a better bound for both deterministic and nondeterministic state complexity.

Theorem 30 (Ellul, Shallit, & Wang [9]) *If L is a unary regular language with* $\mathrm{sc}(L) = n$, *then* $\mathrm{size}(L) = O(n)$. *Furthermore, this bound is asymptotically tight, that is, for each n there is a regular language L_n with state complexity n and size* $\Omega(n)$.

Theorem 31 (Ellul, Shallit, & Wang [9]) *If L is a unary regular language with nondeterministic state complexity n, then* $\mathrm{size}(L) \leq 2n^2 + 4n$.

Proof:

A unary NFA $M = (Q, \{a\}, \delta, q_0, F)$ is said to be in *Chrobak normal form* if Q is the disjoint union of Q_T (the "tail") and Q_C (the "cycles") where $Q_T = \{q_0, q_1, \ldots, q_k\}$, $\delta(q_i, a) = q_{i+1}$ for each $i < k$, and Q_T is a disjoint union of cycles with exactly one transition from q_k to some state in each cycle. For an example of an NFA in Chrobak normal form, see Figure 5.1

Chrobak [5] showed that any n–state unary NFA M may be converted into an equivalent NFA M' in Chrobak normal form such that the "tail" of M' has size at most $n^2 + n$, and the cycles of M' use a total of n states. Our bound follows directly from this. $\square$

5.3 Operations on Regular Languages

As was the case with state complexity, we have simple tight bounds on the increase in regular expression size when these operations are applied.

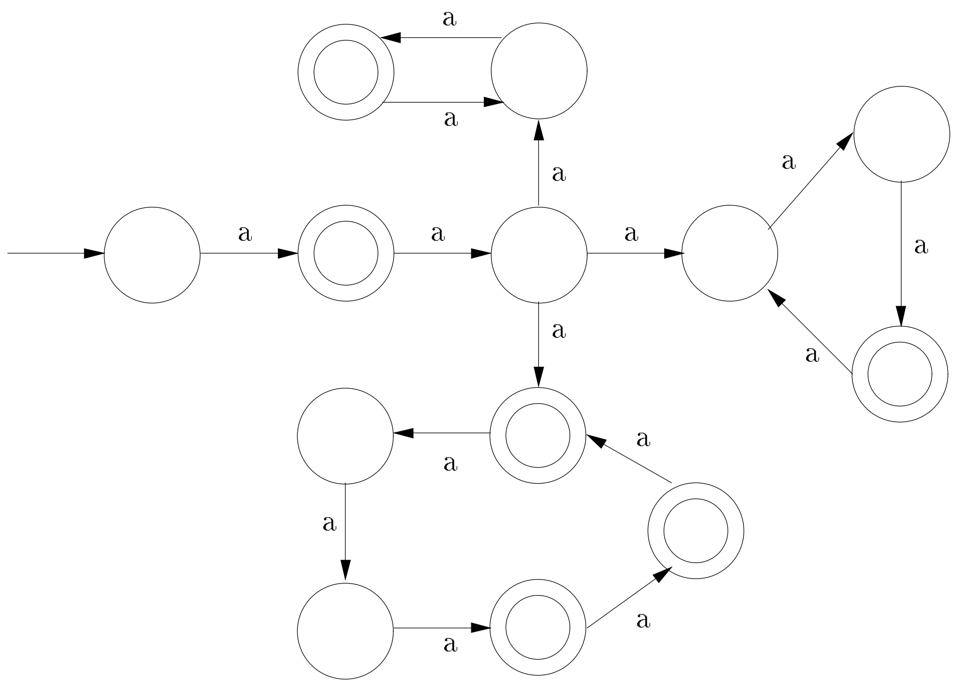

Figure 5.1: A nondeterministic finite automaton in Chrobak normal form.

5.3.1 Union

Theorem 32 *Suppose L_1 and L_2 are regular languages with minimal regular expressions r_1 and r_2 of sizes m and n respectively. Then* $\mathrm{size}(L_1 \cup L_2) \leq m + n$, *and this bound is tight.*

Proof:

To establish the bound, consider the regular expression $r = r_1 + r_2$. Clearly $L(r) = L_1 \cup L_2$ and $\mathrm{size}(r) = m + n$.

To see that the bound is tight, choose m and n arbitrarily, and consider the singleton regular languages $L_1 = \{a^m\}$ and $L_2 = \{b^n\}$. Clearly these languages have minimal regular expressions a^m and b^n respectively, and a minimal regular expression for $L_1 \cup L_2$ is $a^m + b^n$. Thus, the bound is tight.

□

In fact, this bound is also tight in the unary case. This is a direct result of the tightness of the same bound for nondeterministic state complexity.

Theorem 33 *For each i, let $r_i = (a^i)^*$, and let $L_i = L(r_i)$. Then r_i is the minimal regular expression for L_i (so $\text{size}(L_i) = |r_i| = i$) and for any choices of m, n such that m is neither a multiple nor a divisor of n, $\text{size}(L_m \cup L_n) = m + n$. Thus the bound in Theorem 32 is tight, even in the unary case.*

Proof:

The fact that r_i is minimal for L_i follows directly from Corollary 28.

To see that $\text{size}(L_m \cup L_n) = m + n$, note that Holzer & Kutrib [11, Theorem 4] showed that any NFA accepting $L_m \cup L_n$ requires $m + n + 1$ states, so long as m is neither a divisor or a multiple of n. Therefore, by Theorem 27, $(a^m)^* + (a^n)^*$ is a minimal regular expression, and so $\text{size}(L_m \cup L_n) = m + n$ as desired. □

5.3.2 Concatenation

Theorem 34 *Suppose L_1 and L_2 are regular languages with minimal regular expressions r_1 and r_2 of sizes m and n respectively. Then $\text{size}(L_1 L_2) \leq m + n$, and this bound is tight, even in the unary case.*

Proof:

To establish the bound, notice that the regular expression r_1r_2 specifies L_1L_2 and has size $m+n$.

To see that the bound is tight, choose m and n arbitrarily, and consider the singleton regular languages $L_1 = \{a^m\}$ and $L_2 = \{a^n\}$. These languages have minimal regular expressions a^m and a^n respectively, and a minimal regular expression for L_1L_2 is a^{m+n}. Thus, the bound is tight, even in the unary case. $\square$

5.3.3 Kleene Closure

Theorem 35 *Suppose L is a regular language with a minimal regular expression r of size n. Then* $\mathrm{size}(L^*) \leq n$*, and this bound is tight, even in the unary case.*

Proof:

A regular expression for L^* is r^*, so the bound is correct.

Choose an arbitrary n. Then let $L = (a^n)^*$. By Corollary 28, $\mathrm{size}(L) = n$, and, since $L = L^*$, it is also true that $\mathrm{size}(L^*) = n$. Thus, the bound is tight, even in the unary case. $\square$

5.3.4 Reversal

Theorem 36 *Suppose L is a regular language over any alphabet. Then* $\mathrm{size}(L) = \mathrm{size}(L^R)$.

Proof:

Suppose r is the minimal regular expression for L. We will show that there is some regular expression r^R such that $|r^R| = |r|$ and $L(r^R) = L(r)^R$. This will establish that size$(L^R) \leq$ size(L). However, by the symmetry of the reversal operation (ie, $(L^R)^R = L$) this also implies that size$(L) \geq$ size(L^R) and so size$(L) =$ size(L^R), which is our desired result.

In order to achieve this, we can simply let r^R be the reversal of r. To be more formal, we will use induction. First, consider the case where $r = a$ for some $a \in \Sigma$, or $r = \epsilon$, or $r = \emptyset$. In each of these cases, $L(r) = (L(r))^R$ and so taking $r^R = r$ achieves the desired result.

For the inductive step, there are 3 cases:

- If $r = r_1 + r_2$ for some regular expressions r_1 and r_2, then $r^R = r_2^R + r_1^R$.
- If $r = r_1 r_2$ for some regular expressions r_1 and r_2, then $r^R = r_2^R r_1^R$.
- If $r = (r_1)^*$ for some regular expression r_1, then $r^R = (r_1^R)^*$.

In each case, it is clear that $L(r^R) = (L(r))^R$, which gives us the desired result. $\square$

5.3.5 Complementation

In the unary case, we have a good bound on the increase in regular expression size when complementing a regular language. As was the case with nondeterministic state complexity in the unary case, we will need the function $F(n)$ for our analysis (see Section 4.2.1 for a detailed description of this function).

Theorem 37 (Ellul, Shallit, & Wang [9]) *If L is a unary language that is specified by a regular expression r of size n, then $\mathrm{size}(\overline{L}) \in O(F(n+1))$.*

Proof:

Since $\mathrm{size}(L) \leq n$, it follows from Theorem 27 that $\mathrm{nsc}(L) \leq n+1$. Therefore, by Theorem 10, $\mathrm{sc}(L) \in O(F(n+1))$. Note that $\mathrm{sc}(L) = \mathrm{sc}(\overline{L})$, since we can complement a DFA by interchanging final and non–final states, and so by Theorem 30 $\mathrm{size}(\overline{L}) \in O(F(n+1))$ as desired. $\square$

In fact, this bound is asymptotically tight.

Theorem 38 (Ellul, Shallit, & Wang [9]) *There are arbitrarily large integers n for which there exists a regular languages L_n of size $O(n)$ with $\mathrm{size}(\overline{L}) \in \Omega(F(n))$.*

In the general, non–unary case, we have a large gap between the best known upper and lower bounds on the increase in size when complementing a regular language. In fact, our best upper bound is doubly exponential:

Theorem 39 (Ellul, Shallit, & Wang [9]) *There exists a constant c such that for every regular language L over a fixed alphabet Σ of size k, if $\mathrm{size}(L) = n$, then $\mathrm{size}(\overline{L}) \leq c^{2^n}$.*

Proof:

Since size$(L) = n$, it follows from Theorem 27 that nsc$(L) \leq n+1$. So, by Theorem 7, sc$(L) \leq 2^{n+1}$. Therefore, since sc$(L) =$ sc$(\overline{L})$, by Theorem 29, size$(\overline{L}) \leq nk4^{2^{n+1}}$, where k is the size of the alphabet. This is sufficient to show that the bound is correct, since:

$$\begin{aligned} nk4^{2^{n+1}} &= nk(2^2)^{2^{n+1}} \\ &= nk2^{4(2^n)} \\ &\leq k2^{2^n}2^{4(2^n)} \\ &= k2^{5(2^n)} \\ &= k32^{2^n} \\ &\leq (32k)^{2^n}. \end{aligned}$$

Thus, the bound holds, with $c \leq 32k$. $\square$

It seems unlikely that this bound is achievable. However, complementation can cause an exponential increase in minimum regular expression size:

Theorem 40 (Ellul, Shallit, & Wang [9]) *For each n, let $L_n \subseteq \{a,b\}^*$ be the regular language $\{w \;:\; \exists i \; w[i] = w[i+n] = a\}$. Then:*

(a) size$(L_n) \leq 2n+4$, *and*

(b) size$(\overline{L_n}) \geq 2^n - 1$.

Proof:

L_n is specified by the regular expression $(a+b)^*a(a+b)^{n-1}a(a+b)^*$. Thus, the bound in part (a) is satisfied.

For part (b), we will use Lemma 12 to find a lower bound on the nondeterministic state complexity of $\overline{L_n}$. For a word $w \in \{a,b\}^*$, we will denote its image under the morphism $\{a \to b;\ b \to a\}$ by $\overline{w}$.

Let S be the set of ordered pairs $\{(w, \overline{w})\ :\ w \in \{a,b\}^n\}$ Note that $w\overline{w} \in \overline{L_n}$ for each $w \in \{a,b\}^n$, since $(w\overline{w})[i] \neq (w\overline{w})[i+n]$ for each $1 \leq i \leq n$.

Now, suppose that $x, y \in \{a,b\}^n$ with $x \neq y$. Then $x[i] \neq y[i]$ for some $1 \leq i \leq n$. So $x[i] = \overline{y}[i]$ and $y[i] = \overline{x}[i]$. If $x[i] = \overline{y}[i] = a$ then $x\overline{y} \in \overline{L}$. Otherwise, $y[i] = \overline{x}[i] = a$ and $y\overline{x} \in \overline{L}$. So, the premises for Lemma 12 are satisfied, and $\text{nsc}(\overline{L}) \geq 2^n$. Therefore, by Theorem 27, $\text{size}(L) \geq 2^n - 1$, as desired. □

The large gap between the upper and lower bounds leads to the following open problem:

Open Problem 4 *What is the (asymptotically) largest function $f(n)$ such that, for arbitrarily large values of n, there exists a regular language L_n with:*

(a) $\text{size}(L_n) \leq n$, *and*

(b) $\text{size}(\overline{L_n}) \in \Omega(f(n))$*?*

5.3.6 Intersection

In the unary case, we can use our nondeterministic state complexity results to get a bound on the increase in regular expression size when taking the intersection of two regular languages.

Theorem 41 (Ellul, Shallit, & Wang [9]) *Suppose that L_1 and L_2 are unary regular languages of size m and n respectively. Then* $\mathrm{size}(L_1 \cap L_2) \in O((mn)^2)$.

However, we do not know if this bound is tight.

Open Problem 5 *Does there exist a constant $c > 0$ such that, for arbitrarily large values of m and n, there exist unary regular languages L_1 and L_2 of size m and n respectively, such that* $\mathrm{size}(L_1 \cap L_2) \geq c(mn)^2$*?*

In the non–unary case, our bound is different.

Theorem 42 (Ellul, Shallit, & Wang [9]) *There is a constant c such that, for all regular languages L_1 and L_2, if* $\mathrm{size}(L_1) = m$ *and* $\mathrm{size}(L_2) = n$*, then* $\mathrm{size}(L_1 \cap L_2) \leq c^{(m+1)(n+1)}$.

However, we do not know if this bound is tight.

Open Problem 6 *Does there exist a constant $c > 0$ such that, for arbitrarily large values of m and n, there exist unary regular languages L_1 and L_2 of size m and n respectively, such that* $\mathrm{size}(L_1 \cap L_2) \geq 2^{cmn}$*?*

5.3.7 Quotients

Theorem 43 *Let r be a regular expression of size n that specifies a language L over Σ, and let w be a word in Σ^*. Then:*

(a) $\mathrm{size}(Lw^{-1}) \in 2^{O(n)}$*, and*

(b) $\mathrm{size}(w^{-1}L) \in 2^{O(n)}$*.*

Proof:

By Theorem 27, $\mathrm{nsc}(L) \leq n+1$. So, by Theorem 26, $\mathrm{nsc}(Lw^{-1}) \leq n+1$, and $\mathrm{nsc}(w^{-1}L) \leq n+2$. Thus, by Theorem 29, there are regular expressions r' and r'' specifying Lw^{-1} and $w^{-1}L$ respectively, with:

(a) $|r'| \leq |\Sigma|(n+1)4^{n+1} \in 2^{O(n)}$, and

(b) $|r''| \leq |\Sigma|(n+2)4^{n+2} \in 2^{O(n)}$,

as desired. □

However, we do not know if these bounds are tight.

Open Problem 7 *Does there exist a constant $c > 0$ such that, for arbitrarily large values of n, there exists a regular language L of size n and a word w such that* $\mathrm{size}(Lw^{-1}) \geq 2^{cn}$*?*

Open Problem 8 *Does there exist a constant $c > 0$ such that, for arbitrarily large values of n, there exists a regular language L of size n and a word w such that* $\mathrm{size}(w^{-1}L) \geq 2^{cn}$*?*

Chapter 6

Radius

So far, we have studied both deterministic and nondeterministic finite automata, as well as regular expressions. In each case, we were concerned with the size of the smallest representation of a regular language.

In this chapter, we will study the radius of a regular language, which is a measure of the shape of a finite automaton that accepts it, rather than its size. We will examine how radius is related to other measures of descriptional complexity, and we will give bounds on the increase in radius when certain regularity-preserving operations are applied.

6.1 Definitions

For every state q in a finite automaton $M = (Q, \Sigma, \delta, q_0, F)$, we define the *depth* of q, denoted by $\mathrm{depth}(q)$, to be the (graph–theoretic) distance from the start state to q. Formally, if M is a DFA, then

$$\operatorname{depth}(q) = \min_{x \in \Sigma^*} \{|x| : \delta^*(q_0, x) = q\},$$

otherwise, if M is an NFA,

$$\operatorname{depth}(q) = \min_{x \in \Sigma^*} \{|x| : q \in \delta^*(q_0, x)\}.$$

If q is not reachable from q_0 then we define $\operatorname{depth}(q)$ to be infinite.

We are now ready to define the *radius* of a finite automaton. It should be noted that this concept is *not* the same as the graph–theoretic concept of radius. Informally, we may consider the start state of a finite automaton to be the "center" of the automaton, and so the "radius" of this automaton is the distance from the start state to the point that is furthest away. Formally, if $M = (Q, \Sigma, \delta, q_0, F)$ is a finite automaton, then the radius of M, denoted by $\operatorname{rad}(M)$, is $\max\{\operatorname{depth}(q) : q \in Q\}$. For example, the radius of the NFA in Figure 6.1 is 1. In this example, the depth of the start state is 0, and the depth of each of the other two states in 1. Therefore, the radius of the NFA is 1. We can extend this to define the radius of a regular language, which is simply the minimum radius of all DFAs which accept that language. Formally, if L is a regular language, then the *radius* of L, is denoted by $\operatorname{rad}(L)$, and is equal to $\min\{\operatorname{rad}(M) : L(M) = L,\ M \in \text{DFA}\}$.

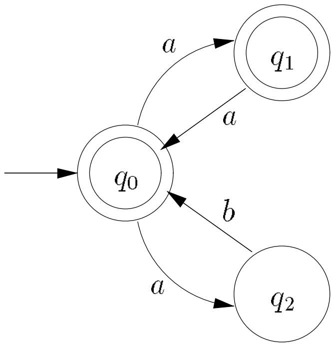

Figure 6.1: An NFA of radius 1

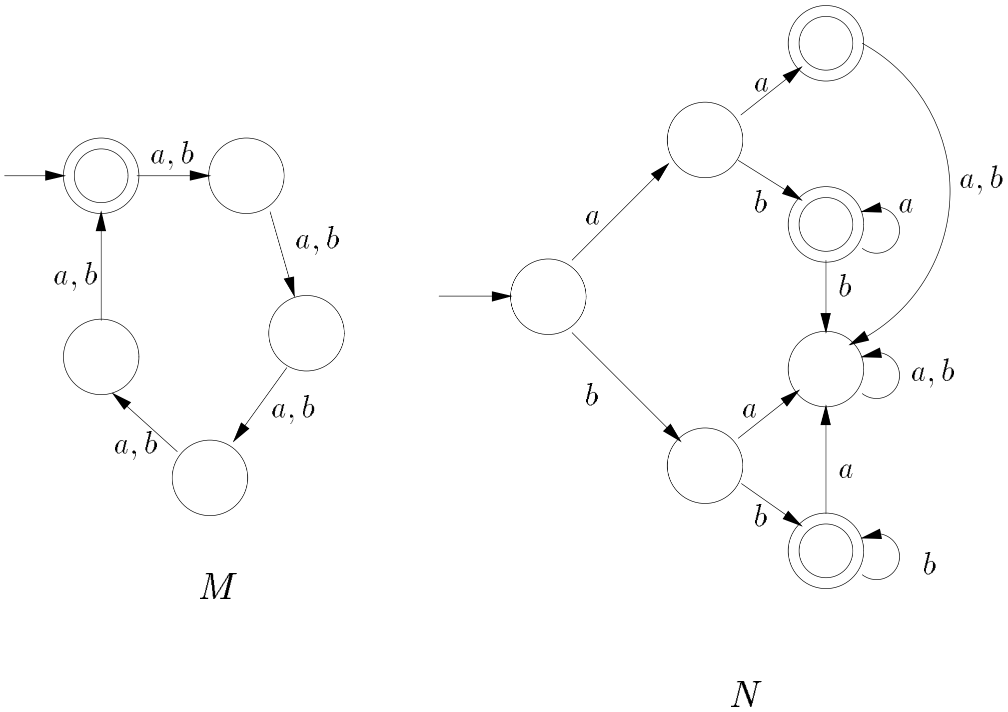

Figure 6.2: The minimal DFAs M and N

6.2 Minimal DFAs

It should be noted that radius and state complexity are two completely different measures. It is possible to have two minimal DFAs M and N with $|M| < |N|$ but $\mathrm{rad}(M) > \mathrm{rad}(N)$. For example, see Figure 6.2 on the previous page. N is the minimal DFA for the language $aa + aba^* + bbb^*$, and M is the minimal DFA for the language $((a+b)^5)^*$. N has radius 2 and size 7, while M has radius 4 and size 5. Therefore, it is not true that $|M| < |N| \Rightarrow \mathrm{rad}(M) \leq \mathrm{rad}(N)$, even if M and N are minimal. However, despite this fact, it is true that, for any regular language L, the minimal DFA for L is also a DFA of minimum radius for L.

Theorem 44 *Suppose L is a regular language with minimal DFA M. Then* $\mathrm{rad}(L) = \mathrm{rad}(M)$.

Proof:

Consider an arbitrary regular language L whose minimal DFA is $M = (Q, \Sigma, \delta, q_0, F)$. Suppose that $\mathrm{rad}(M) = n$. Then clearly $\mathrm{rad}(L) \leq n$.

Since $\mathrm{rad}(M) = n$, there is some state $q_{max} \in Q$ of depth n. Thus, there is some word x of length n such that $\delta^*(q_0, x) = q_{max}$. Additionally, whenever $\delta^*(q_0, w) = q_{max}$, it must be that $|w| \geq |x|$. Thus, in the computation on input x, each state in the path from q_0 to q_{max} is visited only once, otherwise we could remove the loop and find a shorter word leading to q_{max}.

Now, let $x_{[1..k]}$ signify the first k letters of x. Since each state in the path from q to q_{max} on input x was visited only once, we have that $\delta^*(q_0, x_{[1..i]}) \neq$

$\delta^*(q_0, x_{[1..j]})$ for $i \neq j$. So, since M is a minimal DFA for L, the words $x_{[1..i]}$ and $x_{[1..j]}$ must lie in different equivalence classes under the standard Myhill–Nerode equivalence relation.

Let $M' = (Q', \Sigma, \delta', q_0', F')$ be a DFA that accepts L. Then $\delta'^*(q_0', x_{[1..i]}) \neq \delta'^*(q_0', x_{[1..j]})$ when $1 \leq i < j \leq n$. So the path in M' from q_0' to $\delta'^*(q_o', x)$ on input x does not visit the same state twice, and so $\mathrm{depth}(\delta'^*(q_0', x)) = |x| = n$. Thus $\mathrm{rad}(M') \geq n$ and so $\mathrm{rad}(L) \geq n$.

Therefore $\mathrm{rad}(L) = n = \mathrm{rad}(M)$. □

6.3 State Complexity

Since the minimal DFA for a regular language is also the DFA of minimum radius for that language, there are tight upper and lower bounds relating state complexity to radius.

Theorem 45 *For a regular language L over an alphabet Σ, where $|\Sigma| = k \geq 2$:*

$$\mathrm{rad}(L) + 1 \leq \mathrm{sc}(L) \leq \frac{k^{\mathrm{rad}(L)+1} - 1}{k - 1},$$

and, in fact, these bounds are tight.

Proof:

Let L be an arbitrary regular language of radius n, and $M = (Q, \Sigma, \delta, q_0, F)$ be the minimal DFA for L. Then, by Theorem 44, M also has radius n. So, there is some state $q \in Q$ such that $\text{depth}(q) = n$, that is, the shortest path from the start state to q has length n. Therefore, the DFA must contain at least $n + 1$ states (since a path of length n has n edges and $n + 1$ nodes). Thus, $\text{sc}(L) - 1$ is an upper bound on $\text{rad}(L)$.

Since M has radius n, every state in M has a depth of no more than n. However, since M is a DFA and therefore, by definition, complete, each state in M has outdegree equal to $k = |\Sigma|$, there can be at most i^k distinct paths of length i, and therefore there can be at most i^k distinct states of depth i. So, since each state has depth less than or equal to n, the total number of states can be no more than:

$$\sum_{i=0}^{n} i^k = \frac{k^{n+1} - 1}{k - 1}.$$

Thus, the lower bound on $\text{sc}(L)$ holds as well.

It remains to be shown that the bounds are tight. Consider arbitrary natural numbers k and n. Let $\Sigma = \{0, 1, \ldots, k - 1\}$. It is sufficient to show that, given any such k and n, regular languages L_1 and L_2 over Σ can be constructed such that:

1. $\text{rad}(L_1) = \text{rad}(L_2) = n$,

2. $\mathrm{sc}(L_1) = n + 1$, and
3. $\mathrm{sc}(L_2) = \frac{k^{n+1}-1}{k-1}$.

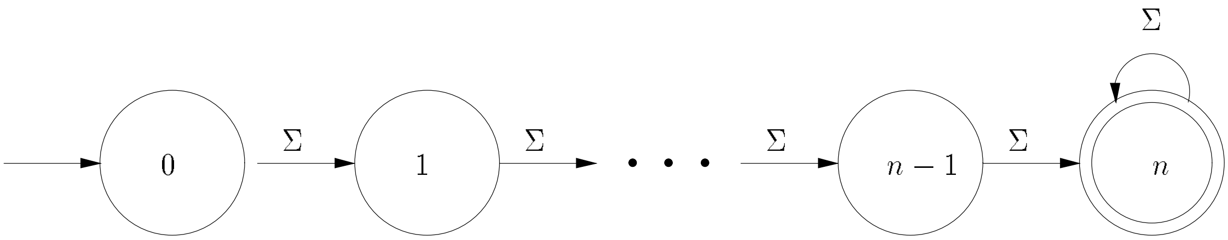

Figure 6.3: The minimal DFA for L_1.

Define $L_1 = \{x \in \Sigma^* : |x| \geq n\}$. Since x is the only word in L, it is clear that the DFA shown in Figure 6.3 is minimal for L_1. So, the radius of L_1 is n, and the state complexity of L_1 is $n + 1$, as desired.

Now let $w_1, w_2, \ldots, w_{k^n}$ represent the k^n words of length n over Σ, in lexicographic order. Let L_2 be the language $\{w_i x : x \in \Sigma^*, i + |x| \geq k^n\}$. Define the DFA $M_2 = (Q, \Sigma, \delta, q_0, F)$ as follows:

$$\begin{aligned} Q &= \left\{q_y : y \in \Sigma^{\leq n}\right\}; \\ \forall a \in \Sigma,\ \delta(q_z, a) &= \begin{cases} q_{za}, & \text{when } |z| < n; \\ q_{w_{i+1}}, & \text{when } |z| = n \text{ and so } z = w_i \text{ for some } i; \end{cases} \\ q_0 &= q_\epsilon; \\ F &= \{q_{w_{k^n}}\}. \end{aligned}$$

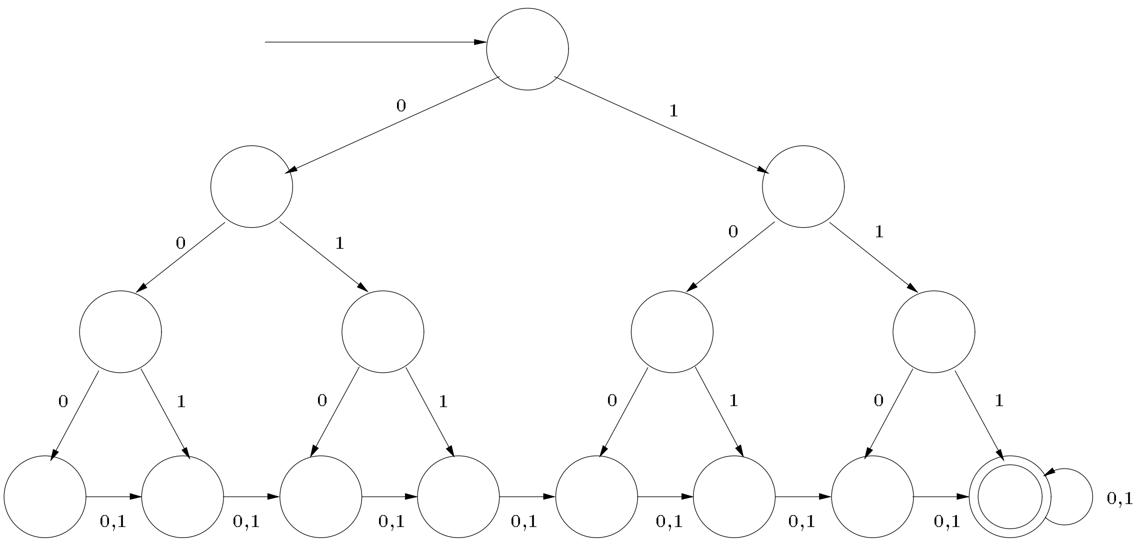

Figure 6.4: A transition diagram for M_2, with $k = 2$ and $n = 3$.

A transition diagram for this DFA with $k = 2$ and $n = 3$ is given in Figure 6.4. The language accepted by M_2 is clearly L_2; however, it remains to be shown that M_2 is the minimal DFA for L_2. It is sufficient to show that any two distinct words z_1, z_2 of length less than or equal to n lie in different Myhill-Nerode equivalence classes.

There are two cases to consider: either the lengths of z_1 and z_2 are the same, or they are different. Suppose first that both words have the same length l, which is less than or equal to n. Let $a \in \Sigma$. Since z_1 and z_2 are different, $z_1 a^{n-l}$ and $z_2 a^{n-l}$ are also different words, both of length n. So, $z_1 a^{n-l} = w_i$ and $z_2 a^{n-l} = w_j$ for some $i \neq j$. Assume without loss of generality that $i < j$. Then $w_i a^{k^n - i} = z_1 a^{n-l+k^n-i}$ is in L_2, but $w_j a^{k^n-i} = z_2 a^{n-l+k^n-i}$ is not in L_2.

So, z_1 and z_2 lie in different equivalence classes.

It is only left to show that if z_1 and z_2 are of different lengths (less than or equal to n), then they lie in different equivalence classes. First, note the following property of L_2: For any word x with $|x| \geq n$, and for any words y_1 and y_2 with $|y_1| = |y_2|$, xy_1 and xy_2 lie in the same equivalence class. That is, if two words of the same length agree on their first n symbols, they are in the same equivalence class.

Now, assume that z_1 and z_2 are of different lengths l and m, both less than or equal to n. Assume without loss of generality that $l < m \leq n$. Now, let a and b be distinct members of Σ. Consider the following words:

$$x_1 = a^{n-m},$$
$$x_2 = a^{m-l},$$
$$x_3 = b^{m-l}.$$

Note that $|x_1x_2| = |x_1x_3| = n - l$. So, $z_1x_1x_2$ and $z_1x_1x_3$ are different words of length n, and thus (as shown above) lie in different equivalence classes.

Finally, note that $z_2x_1x_2$ and $z_2x_1x_3$ have the same length, and agree on the first n symbols (z_2x_1). Therefore, they lie in the same equivalence class. So, when the strings x_1x_2 and x_1x_3 are appended to z_1, they produce words in different equivalence classes. But, when they are appended to z_2, they produce words in the same equivalence class. Thus, z_1 and z_2 must lie in different equivalence classes, and so M_2 is the minimal DFA for L_2. Therefore,

L_2 has radius n and state complexity $\frac{k^{n+1}-1}{k-1}$, and the lower bound is tight. □

In the unary case, we have a much simpler result, due to the structure of unary DFAs.

Theorem 46 *Suppose L_1 is a unary language. Then* $\text{rad}(L) = \text{sc}(L) - 1$.

Proof:

Suppose M is a unary DFA. Since M is unary and deterministic, the outdegree of each state is exactly 1. Thus, there is at most one state of depth k for any particular k. So, suppose $|M| = n$. Then for each $0 \leq k \leq n-1$ there is exactly one state of depth k. Thus, $\text{rad}(M) = n - 1$. So, if M is a unary DFA, $\text{rad}(M) = |M| - 1$. In particular, if M is the minimal DFA for L, then $\text{rad}(L) = \text{rad}(M) = |M| - 1 = \text{sc}(L) - 1$, as desired. □

6.4 Operations on Regular Languages

As was the case for both deterministic and nondeterministic state complexity, and regular expression size, we would like to find tight upper bounds on the increase in radius when the usual regularity-preserving operations are applied. However, for most of these operations, we only have trivial upper bounds, and there is quite a large gap between our best upper and lower bounds. In the unary case, these problems are much simpler. Their results follow directly from Theorem 46 and our results for unary state complexity (see Section 3.2).

6.4.1 Union and Intersection

Since the radius of a regular language is the same as the radius of its minimal DFA (recall Theorem 44), it seems intuitive that the upper bounds for union and intersection should be similar. Given two regular languages L_1 and L_2, the constructions used to create a DFA for $L_1 \cup L_2$ and $L_1 \cap L_2$ are almost identical: the only difference between the two DFAs is the set of final states. Thus, the constructed DFAs will have the same size and radius. In fact, as shown in Section 3.3, the tight upper bounds on state complexity are the same for union and intersection.

For both union and intersection, the best upper bound known on the increase in radius is trivial:

Theorem 47 *Suppose L_1 and L_2 are regular languages over the alphabet Σ, where $|\Sigma| = k \geq 2$. Furthermore suppose that* $\mathrm{rad}(L_1) = m$ *and* $\mathrm{rad}(L_2) = n$*. Then:*

$$\mathrm{rad}(L_1 \cup L_2) \leq \frac{k^{m+n+2} - k^{m+1} - k^{n+1} + 1}{k^2 - 2k + 1} - 1,$$

and

$$\mathrm{rad}(L_1 \cap L_2) \leq \frac{k^{m+n+2} - k^{m+1} - k^{n+1} + 1}{k^2 - 2k + 1} - 1.$$

Proof:

By Theorem 45,

$$\mathrm{sc}(L_1) \leq \frac{k^{m+1} - 1}{k - 1}$$

and

$$\mathrm{sc}(L_1) \leq \frac{k^{m+1} - 1}{k - 1}.$$

Thus,

$$\begin{aligned} \mathrm{sc}(L_1 \cup L_2) &\leq \left(\frac{k^{m+1}-1}{k-1}\right)\left(\frac{k^{n+1}-1}{k-1}\right) \\ &= \frac{k^{m+n+2}-k^{m+1}-k^{n+1}+1}{k^2-2k+1}, \end{aligned}$$

and

$$\begin{aligned} \mathrm{sc}(L_1 \cap L_2) &\leq \left(\frac{k^{m+1}-1}{k-1}\right)\left(\frac{k^{n+1}-1}{k-1}\right) \\ &= \frac{k^{m+n+2}-k^{m+1}-k^{n+1}+1}{k^2-2k+1}. \end{aligned}$$

Therefore, by Theorem 45,

$$\mathrm{rad}(L_1 \cup L_2) \leq \frac{k^{m+n+2}-k^{m+1}-k^{n+1}+1}{k^2-2k+1} - 1,$$

and

$$\mathrm{rad}(L_1 \cap L_2) \leq \frac{k^{m+n+2}-k^{m+1}-k^{n+1}+1}{k^2-2k+1} - 1,$$

as desired. □

However, we do not know if either of these bounds are tight.

Open Problem 9 *What is the (asymptotically) largest function $f(m,n)$ such that, for arbitrarily large values of m and n, there exist regular languages L_m and L'_n with:*

(a) $\mathrm{rad}(L_m) = m$,

(b) $\mathrm{rad}(L'_n) = n$, *and*

(c) $\mathrm{rad}(L_m \cup L'_n) \in \Omega(f(m,n))$*?*

Open Problem 10 *What is the (asymptotically) largest function $f(m,n)$ such that, for arbitrarily large values of m and n, there exist regular languages L_m and L'_n with:*

(a) $\mathrm{rad}(L_m) = m$*,*

(b) $\mathrm{rad}(L'_n) = n$*, and*

(c) $\mathrm{rad}(L_m \cap L'_n) \in \Omega(f(m,n))$*?*

However, we do know that a multiplicative increase is achievable, at least for small instances. That is, there exist languages L_1 and L_2 such that $\mathrm{rad}(L_1 \cup L_2)$ and $\mathrm{rad}(L_1 \cap L_2)$ are both larger than $\mathrm{rad}(L_1)\mathrm{rad}(L_2)$. For example:

Example 48 *For each $n > 1$, define $M_n = (Q_n, \Sigma, \delta_n, 1, F_n)$, where:*

- $Q_n = \{1, \ldots, 2^{n+1} - 1\}$*,*
- $\Sigma = \{a, b\}$*,*
- $\delta_n(i, a) = 2i$ *for all* $1 \le i \le 2^n - 1$*,*
- $\delta_n(i, b) = 2i + 1$ *for all* $1 \le i \le 2^n - 1$*,*
- $\delta_n(i, a) = \delta_n(i, b) = i + 1$ *for all* $2^n \le i \le 2^{n+1} - 2$*,*
- $\delta_n(2^{n+1} - 1, a) = \delta_n(2^{n+1} - 1, b) = 1$*, and*
- $F_n = \{2^{n+1} - 1\}$*.*

For example, see Figure 6.5 for a transition diagram of M_2.

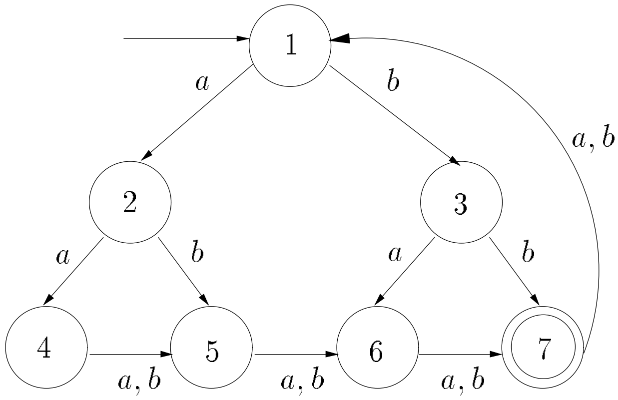

Figure 6.5: The DFA M_2.

Also, define $L_n = L(M_n)$. Then:

(a) $\mathrm{rad}(M_n) = n$,

(b) M_n is minimal, and so $\mathrm{rad}(L_n) = n$,

(c) $\mathrm{rad}(L_n \cap L_m) > mn$ *for some values of m and n, and*

(d) $\mathrm{rad}(L_n \cup L_m) > mn$ *for some values of m and n.*

Proof:

Part (a) is obvious. The minimality of M_n follows from the Myhill–Nerode theorem, and so part (b) is a direct result of Theorem 45. For parts (c) and (d), we chose some small values of m and n and used the Grail software package to examine the effects that union and intersection have on radius for these particular languages. The Grail software package, including source code, is available at `http://www.csd.uwo.ca/research/grail/`. Also, see

Appendix A for the source code that was written and added into Grail to compute radius.

Our results for values of m and n between 2 and 4 follow:

The DFAs M_2, M_3, and M_4:

```
$ cat DFA2
(START) |- 1
1 a 2
1 b 3
2 a 4
2 b 5
3 a 6
3 b 7
4 a 5
4 b 5
5 a 6
5 b 6
6 a 7
6 b 7
7 a 1
7 b 1
7 -| (FINAL)

$ cat DFA3
(START) |- 1
1 a 2
1 b 3
2 a 4
2 b 5
3 a 6
3 b 7
4 a 8
4 b 9
5 a 10
5 b 11
```

```
6 a 12
6 b 13
7 a 14
7 b 15
8 a 9
8 b 9
9 a 10
9 b 10
10 a 11
10 b 11
11 a 12
11 b 12
12 a 13
12 b 13
13 a 14
13 b 14
14 a 15
14 b 15
15 a 1
15 b 1
15 -| (FINAL)

$ cat DFA4
(START) |- 1
1 a 2
1 b 3
2 a 4
2 b 5
3 a 6
3 b 7
4 a 8
4 b 9
5 a 10
5 b 11
6 a 12
6 b 13
7 a 14
7 b 15
8 a 16
```

```
8 b 17
9 a 18
9 b 19
10 a 20
10 b 21
11 a 22
11 b 23
12 a 24
12 b 25
13 a 26
13 b 27
14 a 28
14 b 29
15 a 30
15 b 31
16 a 17
17 a 18
18 a 19
19 a 20
20 a 21
21 a 22
22 a 23
23 a 24
24 a 25
25 a 26
26 a 27
27 a 28
28 a 29
29 a 30
30 a 31
16 b 17
17 b 18
18 b 19
19 b 20
20 b 21
21 b 22
22 b 23
23 b 24
24 b 25
```

```
25 b 26
26 b 27
27 b 28
28 b 29
29 b 30
30 b 31
31 -| (FINAL)
```

The radius of each language:

```
$ fmrad < DFA2
2

$ fmrad < DFA3
3

$ fmrad < DFA4
4
```

The radius of each union of languages:

```
$ fmunion DFA2 DFA3 | fmdeterm | fmmin | fmcomp | fmrad
9

$ fmunion DFA2 DFA4 | fmdeterm | fmmin | fmcomp | fmrad
9

$ fmunion DFA3 DFA4 | fmdeterm | fmmin | fmcomp | fmrad
14
```

The radius of each intersection of languages:

```
$ fmcross DFA2 DFA3 | fmdeterm | fmmin | fmcomp | fmrad
9

$ fmcross DFA2 DFA4 | fmdeterm | fmmin | fmcomp | fmrad
8

$ fmcross DFA3 DFA4 | fmdeterm | fmmin | fmcomp | fmrad
```

14

So:

$$\text{rad}(L_2 \cup L_3) = \text{rad}(L_2 \cap L_3) = 9 > \text{rad}(L_2)\text{rad}(L_3),$$

and

$$\text{rad}(L_3 \cup L_4) = \text{rad}(L_3 \cap L_4) = 14 > \text{rad}(L_3)\text{rad}(L_4),$$

as desired. □

In the unary case, tight upper bounds for both union and intersection follow directly from Theorem 46 and our results on unary state complexity.

Theorem 49 *Suppose L_1 and L_2 are unary regular languages, with* $\text{rad}(L_1) = m$ *and* $\text{rad}(L_2) = n$*. Then:*

- $\text{rad}(L_1 \cup L_2) \leq mn + m + n$, *and*
- $\text{rad}(L_1 \cap L_2) \leq mn + m + n$.

Furthermore, these bounds are tight.

Proof:

Since L_1 has radius m, it has state complexity $m + 1$. Similarly, L_2 has state complexity $n + 1$. So the state complexities of $L_1 \cup L_2$ and $L_1 \cap L_2$ are no more than $(m + 1)(n + 1)$. Thus, the radius of $L_1 \cup L_2$ (or $L_1 \cap L_2$) is no more than $(m + 1)(n + 1) - 1 = mn + m + n$.

The tightness of the bounds follow directly from the tightness of the state complexity bounds. □

6.4.2 Concatenation

Once again, a trivial bound follows from Theorem 45, along with our results on state complexity.

Theorem 50 *Suppose L_1 and L_2 are regular languages over the alphabet Σ, where $|\Sigma| = k \geq 2$. Furthermore suppose that the radius of L_1 is m, the radius of L_2 is n, and the minimal DFA for L_1 has j final states. Then:*

$$\mathrm{rad}(L_1L_2) \leq \frac{(k^{m+1}-1)\,2^{\frac{k^{n+1}-1}{k-1}}}{k-1} - j2^{\frac{k^{n+1}-k}{k-1}}.$$

Proof:

By Theorem 45,

$$\mathrm{sc}(L_1) \leq \frac{k^{m+1}-1}{k-1}$$

and

$$\mathrm{sc}(L_2) \leq \frac{k^{n+1}-1}{k-1}.$$

Thus,

$$\begin{aligned} \mathrm{sc}(L_1L_2) &\leq \frac{k^{m+1}-1}{k-1}2^{\frac{k^{n+1}-1}{k-1}} - j2^{\frac{k^{n+1}-1}{k-1}-1} \\ &= \frac{(k^{m+1}-1)\,2^{\frac{k^{n+1}-1}{k-1}}}{k-1} - j2^{\frac{k^{n+1}-k}{k-1}}. \end{aligned}$$

Therefore, by Theorem 45,

$$\mathrm{rad}(L_1L_2) \leq \frac{(k^{m+1}-1)\,2^{\frac{k^{n+1}-1}{k-1}}}{k-1} - j2^{\frac{k^{n+1}-k}{k-1}},$$

as desired.

□

However, we do not know if this bounds is tight.

Open Problem 11 *What is the (asymptotically) largest function $f(m, n)$ such that, for arbitrarily large values of m and n, there exist regular languages L_m and L'_n with:*

(a) $\mathrm{rad}(L_m) = m$,

(b) $\mathrm{rad}(L'_n) = n$, *and*

(c) $\mathrm{rad}(L_m L'_n) \in \Omega(f(m, n))$*?*

In the unary case, a tight upper bound follows directly from Theorem 46 and our results on unary state complexity.

Theorem 51 *Suppose L_1 and L_2 are unary regular languages, with* $\mathrm{rad}(L_1) = m$ *and* $\mathrm{rad}(L_2) = n$. *Then* $\mathrm{rad}(L_1 L_2) \leq mn + m + n$ *Furthermore, this bound is tight.*

Proof:

Since L_1 has radius m, it has state complexity $m + 1$. Similarly, L_2 has state complexity $n + 1$. So the state complexity of $L_1 L_2$ is no more than $(m+1)(n+1)$. Thus, the radius of $L_1 L_2$ is no more than $(m+1)(n+1) - 1 = mn + m + n$.

The tightness of the bound follows directly from the tightness of the state complexity bound. □

6.4.3 Kleene Closure

Once again, a trivial bound follows from Theorem 45, along with our results on state complexity.

Theorem 52 *Suppose L is a regular language over the alphabet Σ of size $k \geq 2$. Furthermore suppose that* $\mathrm{rad}(L) = m$*. Then:*

$$\mathrm{rad}(L^*) \leq 2^{\frac{k^{m+1}-k}{k-1}} + 2^{\frac{k^{m+1}-2k+1}{k-1}} - 1.$$

Proof:

By Theorem 45,

$$\mathrm{sc}(L) \leq \frac{k^{m+1}-1}{k-1}.$$

Thus,

$$\begin{aligned} \mathrm{sc}(L^*) &\leq 2^{\frac{k^{m+1}-1}{k-1}-1} + 2^{\frac{k^{m+1}-1}{k-1}-2} \\ &= 2^{\frac{k^{m+1}-k}{k-1}} + 2^{\frac{k^{m+1}-2k+1}{k-1}}. \end{aligned}$$

Therefore, by Theorem 45,

$$\mathrm{rad}(L^*) \leq 2^{\frac{k^{m+1}-k}{k-1}} + 2^{\frac{k^{m+1}-2k+1}{k-1}} - 1,$$

as desired. □

However, we do not know if this bound is tight.

Open Problem 12 *What is the (asymptotically) largest function $f(n)$ such that, for arbitrarily large values of n, there exists a regular language L_n with $\mathrm{rad}(L_n) = n$ and $\mathrm{rad}(L_n^*) \in \Omega(f(n))$?*

In the unary case, a tight upper bound follows directly from Theorem 46 and our results on unary state complexity.

Theorem 53 *Suppose L_1 is a unary regular language of radius m. Then $\mathrm{rad}(L_1^*) \leq m^2$, and this bound is tight.*

Proof:

Since L_1 has radius m, it has state complexity $m+1$. So the state complexity of L_1^* is no more than $((m+1)-1)^2+1 = m^2+1$. Thus, the radius of L_1^* is no more than $m^2+1-1 = m^2$.

The tightness of the bound follows directly from the tightness of the state complexity bound. □

6.4.4 Reversal

Once again, a trivial bound follows from Theorem 45, along with our results on state complexity.

Theorem 54 *Suppose L is a regular language over the alphabet Σ of size $k \geq 2$. Furthermore suppose that $\mathrm{rad}(L) = m$. Then:*

$$\mathrm{rad}(L^*) \leq 2^{\frac{k^{m+1}-1}{k-1}} - 1.$$

Proof:

By Theorem 45,

$$\mathrm{sc}(L) \leq \frac{k^{m+1}-1}{k-1}.$$

Thus,

$$\mathrm{sc}(L^R) \leq 2^{\frac{k^{m+1}-1}{k-1}}.$$

Therefore, by Theorem 45,

$$\mathrm{rad}(L^*) \leq 2^{\frac{k^{m+1}-1}{k-1}} - 1,$$

as desired. □

If we allow our alphabet size to grow arbitrarily large, we can achieve a large increase in radius when reversing a regular language.

Example 55 *Let Σ_k represent the k–letter alphabet $\{a_1, a_2, \ldots, a_k\}$, and, for each i, let p_i be the ith prime. Define the language L_k over the alphabet Σ_k to be:*

$$\bigcup_{i=1}^{k} a_i(\Sigma_k^{p_i})^*.$$

Then:

(a) $\mathrm{rad}(L_k) = p_k$, *and*

(b) $\mathrm{rad}(L_k^R) = \prod_{i=1}^{k} p_i - 1$.

Proof:

To see that the radius of L_k is p_k, consider the DFA $M_k = (Q_k, \Sigma_k, \delta_k, q_0, F_k)$, where:

- $Q = \{q_0\} \cup \{q_{i,j} \ : \ 1 \leq i \leq k; \ 1 \leq j \leq p_k\}$,
- $\delta_k(q_0, a_i) = q_{i,1}$ for all $1 \leq i \leq k$,
- $\delta_k(q_{i,j}, a) = q_{i,j+1}$ for all $1 \leq i \leq k$; $1 \leq j \leq p_i - 1$; $a \in \Sigma_k$,
- $\delta_k(q_{i,p_i}, a) = q_{i,1}$ for all $1 \leq i \leq k$; $a \in \Sigma_k$, and
- $F_k = \{q_{i,1} \ : \ 1 \leq i \leq k\}$,

as shown in Figure 6.6.

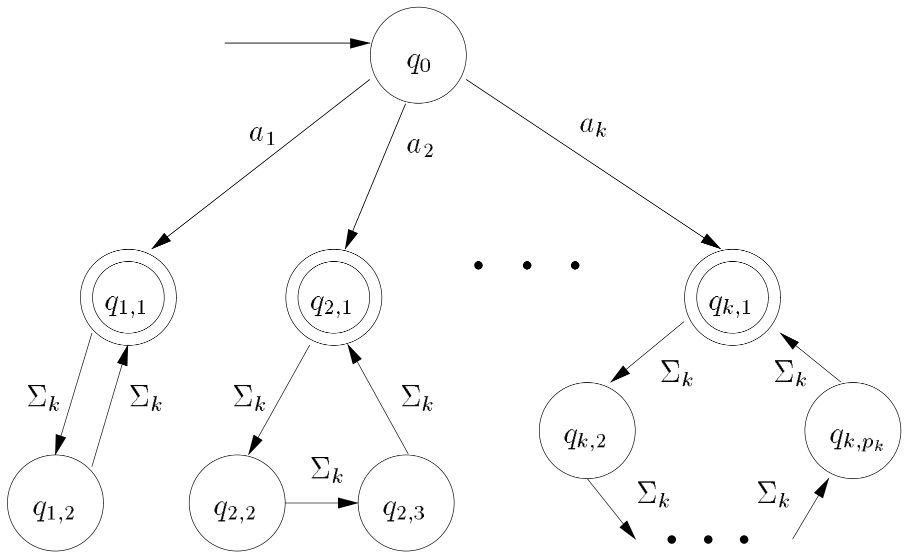

Figure 6.6: The DFA that accepts L_k.

This DFA accepts L_k, and, by the theorem of Myhill–Nerode, it is clearly a minimal DFA. Thus, $\mathrm{rad}(L_k) = \mathrm{rad}(M_k) = p_k$.

It is left to show that $\mathrm{rad}(L_k^R) = \prod_{i=1}^{k} p_i - 1$. First, note that:

$$L_k^R \;=\; \bigcup_{i=1}^{k} (\Sigma_k^{p_i})^* a_i \;=\; \{wa_i \;:\; w \in \Sigma_k^*,\ |w| \equiv 0 \pmod{p_i}\}\,.$$

So, consider some fixed $w \in {\Sigma_k}^*$. Now, let m be the product of the first k primes, that is, $m = \prod_{i=1}^{k} p_i$. Then $w\Sigma_k \subseteq L_k^R$ if and only if $|w| \equiv 0 \pmod{m}$. Therefore, any Myhill–Nerode equivalence class that contains some word of a length divisible by m must contain *only* words of length divisible by m. Also, note that, for any word x, the following three statements are equivalent:

1. $\exists y, |y| \equiv 0 \pmod{m}$ such that $yx \in L_k^R$
2. $x \in L_k^R$
3. $\forall y$ such that $|y| \equiv 0 \pmod{m}$, $yx \in L_k^R$

Thus, it follows that all words of length divisible by m are in the same Myhill–Nerode equivalence class, and therefore there must be a single Myhill-Nerode equivalence class that contains only the words of length divisible by m. In the minimal DFA for L_k^R, this class will be represented by the start state (since 0 is divisible by m). Also, any path of length m that starts at the start state must also end at that state (since the equivalence class contains all words of length m) and so there can be no state that has a depth greater than $m-1$. Thus, the radius of the language can be no more than $m-1$. Also, there may be no shorter path from the start state to itself (otherwise there would be a shorter word in the equivalence class). So, consider the state (in the minimal DFA) that represents the equivalence class containing a_1^{m-1}. It follows that this state must have depth $m-1$, and so the radius of the language can be no less than that. Thus, the radius of L_k^R is exactly $\prod_{i=1}^k p_i - 1$, as desired.

□

In fact, we can achieve a similar result over a fixed–size alphabet

Corollary 56 *Define Σ_k, p_i, and L_k as in Example 55. Additionally, for each k, define $\varphi_k : \Sigma^k \to \{0,1\}^{\lceil \log_2 k \rceil}$ as follows: $\varphi_k(a_i) = [i-1]_2$, that is, $\varphi_k(a_i)$ is the base–2 representation of $i-1$, padded on the left with 0's (if necessary) to ensure that $|\varphi_k(a_i)| = \lceil \log_2 k \rceil$.*

For each k, define L'_k (over the alphabet $\{0,1,2\}$) to be

$$\bigcup_{i=1}^{k} \varphi(a_i)(2^{p_i})^*.$$

Then:

(a) $\mathrm{rad}(L_k) = p_k + \lceil \log_2 k \rceil - 1$, *and*

(b) $\mathrm{rad}(L_k^R) = \prod_{i=1}^k p_i - 1$.

Proof:

The proof is almost identical to the proof of Example 55, since $w\{0,1\}^{\lceil \log_2 k \rceil} \subseteq L_k$ if and only if $|w| \equiv 0 \pmod{\prod_{i=1}^k p_i}$. □

The large gap between the best known increase and the best known upper bound leads to the following open problem.

Open Problem 13 *Suppose we fix the size of an alphabet Σ to be constant. Then, what is the (asymptotically) largest function $f(n)$ such that, for arbitrarily large values of n, there exists a regular language L_n with:*

(a) $\mathrm{rad}(L_n) = n$, *and*

(b) $\mathrm{rad}(L_n^R) \in \Omega(f(n))$*?*

Of course, in the unary case, the problem is trivial, since $L = L^R$ and so $\mathrm{rad}(L) = \mathrm{rad}(L^R)$ for any unary language L.

6.4.5 Complementation

Lemma 57 *If L is a regular language, then the radius of $\overline{L}$ is the same as the radius of L.*

Proof:

If $M = (Q, \Sigma, \delta, q_0, F)$ is the minimal DFA for L, then $\overline{M} = (Q, \Sigma, \delta, q_0, Q - F)$ is the minimal DFA for $\overline{L}$. Since M and $\overline{M}$ have the same set of states, the same transition function, and the same start state, they must also have the same radius, and so as a result of Theorem 44, L and $\overline{L}$ must also have the same radius. □

6.4.6 Quotients

Theorem 58 *Suppose L is a regular language of radius n, and w is a word. Then the radius of Lw^{-1} is no more than n. Furthermore, this bound is tight, even in the unary case. That is, there are cases where* $\mathrm{rad}(Lw^{-1}) = \mathrm{rad}(L)$.

Proof:

Let $M = (Q, \Sigma, \delta, q_0, F)$ be the minimal DFA for L. So the radius of M is n. Define F' to be the following subset of Q: for each $q \in Q$, $q \in F'$ if and only if $\delta^*(q, w) \in F$. Then clearly $M' = (Q, \Sigma, \delta, q_0, F')$ accepts Lw^{-1}, and the radius of M' is the same as the radius of M (since the two automata have the same set of states, the same transition function, and the same start state). Thus, the radius of Lw^{-1} can be no more than n.

To see that this bound is tight, even in the unary case, let $\Sigma = \{a\}$, and choose n and k arbitrarily. Let $L = (a^{n+1})^*$ and let let $w = a^k$. Then the radius of L is n, since the graph of its minimal DFA is simply a directed $(n+1)$–cycle with each edge labelled "a" and a single final state, also designated as the start state.

Let $j = (n + 1) - k \pmod{n+1}$. Then $Lw^{-1} = a^j(a^{n+1})^*$, and the minimal DFA for Lw^{-1} has the same set of states and transition function as the minimal DFA for L (the only difference is the final state, which is the unique state of depth j). Therefore, the radius of Lw^{-1} is the same as the radius of L. For an example, see Figure 6.7. □

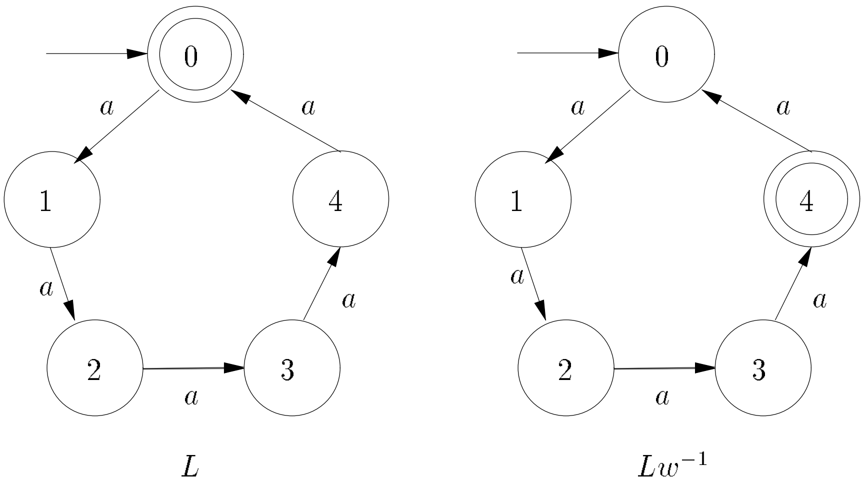

Figure 6.7: The minimal DFAs for L and Lw^{-1} when $n = 4$ and $k = 6$.

Chapter 7

Nondeterministic Radius

In this chapter, we will extend the concept of radius and apply it to nondeterministic finite automata. This will lead us to study the nondeterministic radius of a regular language, and examine how nondeterministic radius is related to other measures of descriptional complexity. Also, we will give bounds on the increase in radius when certain regularity-preserving operations are applied.

7.1 Definitions

The nondeterministic radius of a regular language is the minimum radius of all NFAs which accept that language. Formally, if L is a regular language, then the **nondeterministic radius** of L is denoted by $\mathrm{nrad}(L)$, and is equal to $\min\{\mathrm{rad}(M) : L(M) = L,\ M \in \mathrm{NFA}\}$.

In Section 4.1, we showed how an NFA–ϵ may be converted to an NFA (without ϵ–transitions) with no increase in the number of states used. Unfortunately, we cannot make the same statement concerning radius. However, we can use a similar construction to show that an NFA–ϵ may be converted to an NFA with an increase in radius of at most 1.

Lemma 59 *Suppose M is an NFA–ϵ of radius n. Then there exists an NFA (without ϵ–transitions) M' that is equivalent to M and has a radius of no more than $n + 1$.*

Proof:

Consider an NFA–ϵ $M = (Q, \Sigma, \delta, q_0, F)$, and suppose that the radius of M is n. We will construct an equivalent NFA $M' = (Q', \Sigma, \delta', q_0, F')$ as follows:

- $Q' = \left(\bigcup_{x\in\Sigma^+} \delta^*(q_0, x)\right) \cup \{q_0\}$. That is, Q' contains the start state, along with the set of all states in Q that are reachable from the start state by following a path that is not completely made up of ϵ–transitions.
- For any $(q, a) \in Q' \times \Sigma$, $\delta'(q, a) = \delta^*(q, a)$.
- If $\delta^*(q_0, \epsilon)$ contains any final states, then $F' = F \cup \{q_0\}$. Otherwise, $F' = F$.

This construction is almost identical to the one in Section 4.1, with the only difference being that unreachable states are removed. Also, this construction has the property that for any $w \in \Sigma^+$, $\delta'^*(q_0, w) = \delta^*(q_0, w)$. (Since there are no ϵ–transitions in M', it is necessarily true that $\delta'^*(q_0, \epsilon) = \{q_0\}$.)

It is left to show that the radius of M' is no more than $n + 1$. To avoid confusion, it is necessary to distinguish between the depth of a state $q \in Q'$

in the machine M, and its depth in M'. We will refer to these quantities as $\text{depth}_M(q)$ and $\text{depth}_{M'}(q)$ respectively.

Consider a state $q \in Q'$. Let w be a minimum–length word from Σ^* such that $q \in \delta^*(q_0, w)$. So, by definition, $\text{depth}_M(q) = |w|$. There are two cases to consider:

- If $|w| > 0$, then, by the property noted above, $\delta'^*(q_0, w) = \delta^*(q_0, w)$ and so $q \in \delta'^*(q_0, w)$. So by definition, $\text{depth}_{M'}(q) \leq |w|$, that is, $\text{depth}_{M'}(q) \leq \text{depth}_M(q)$ and so it follows that $\text{depth}_{M'}(q) \leq \text{rad}(M)$.
- Otherwise, if $|w| = 0$, that is, if $w = \epsilon$, then choose some $(q', a) \in Q' \times \Sigma$ such that $q \in \delta'(q', a)$. Since each state in Q' is reachable in M', such a pair (q', a) must exist. Now, choose a minimal–length word $w' \in \Sigma^*$ such that $q' \in \delta^*(q_0, w)$. Note that, by definition, $\text{depth}_M(q') = |w|$. Also, $q \in \delta^*(q_0, wa)$, and so $q \in \delta'^*(q_0, wa)$. Thus, $\text{depth}_{M'}(q) \leq |w| + 1 = \text{depth}_M(q') + 1$. So it follows that $\text{depth}_{M'}(q) \leq \text{rad}(M) + 1$.

In either case, we have come to the conclusion that $\text{depth}_{M'}(q) \leq \text{rad}(M)+1$. Since the choice of $q \in Q'$ was arbitrary, we can conclude that $\text{rad}(M') \leq \text{rad}(M) + 1$. □

It should be noted that this bound is the best possible. That is, it is not always possible to convert an NFA–ϵ into an NFA without having the radius increase by 1. Examples of such NFA–ϵ's will be seen throughout this section (for example, see the proof of Theorem 67). So, it is important that, in our discussions of nondeterministic radius, we limit ourselves to NFAs without ϵ–transitions (unlike our discussions of nondeterministic state complexity, where we were free to use NFAs and NFA–ϵ's interchangeably).

7.2 Lower Bounds

As was the case with deterministic radius, we have some trivial lower bounds on the nondeterministic radius of a language

Lemma 60 *Suppose L is a nonempty regular language, and w is the shortest word in L. Then* $\mathrm{nrad}(L) \geq |w|$. *Furthermore, this bound is tight.*

Proof:

Suppose $M = (Q, \Sigma, \delta, q_0, F)$ is a minimum–radius NFA that accepts L. Since L is nonempty, F must be nonempty. Let $q_F \in F$. If $\mathrm{depth}(q_F) < |w|$, then there is some word w' with $|w'| < |w|$ and $q_F \in \delta^*(q_0, w')$. So $w' \in L$, since $q_F \in F$. This is a contradiction, since w is the shortest word in L. Thus, the depth of q_F is at least $|w|$ and so $\mathrm{rad}(M) = \mathrm{nrad}(L) \geq |w|$.

To see that the bound is tight, simply let $L = \{w\}$ for any word w. The nondeterministic radius of L can be no more than $|w|$ since it is accepted by a $|w| + 1$–state NFA of radius $|w|$. □

Lemma 61 *Suppose L is nonempty regular language such that:*

- *$\epsilon \in L$,*
- *$L \cap \Sigma^+ \neq \emptyset$, and*
- *w is the shortest word in $L \cap \Sigma^+$.*

Then the nondeterministic radius of L is at least $|w| - 1$. Furthermore, this bound is tight.

Proof:

Suppose $M = (Q, \Sigma, \delta, q_0, F)$ is a minimum–radius NFA for L. Since $w \in L$, there must be some $q_F \in F$ with $q_F \in \delta^*(q_0, w)$. Choose some state q such that, for some $a \in \Sigma$, $q_F \in \delta(q, a)$. Let w' be the minimum–length word such that $q \in \delta(q_0, w')$. So, depth$(q) = |w'|$. Note that $w'a \in L$, so $|w'| \geq |w| - 1$ (by the minimality of w). So depth$(q) \geq |w| - 1$ and so rad$(M) \geq |w| - 1$, which establishes the bound.

To see that the bound is tight, let $L = (a^n)^*$ for some n. Then the shortest word in $L \cap \Sigma^+$ is a^n, and the nondeterministic radius of L is $n - 1$, as will be shown in Corollary 65 in the section on unary languages. $\square$

Lemma 62 *Suppose L is a regular language over the alphabet Σ. The languages $\emptyset$, $\{\epsilon\}$, and Δ^* (for any finite set of letters $\Delta \subseteq \Sigma$) all have nondeterministic radius 0. All other regular languages L have a larger nondeterministic radius.*

Proof:

Note that any NFA that has radius 0 must contain only 1 state, and any 1–state NFA has radius 0. Thus, the languages $\emptyset$, $\{\epsilon\}$, and Δ^* clearly have nondeterministic radius 0 since they are accepted by 1–state NFAs.

Suppose some regular language is accepted by a 1–state NFA $M = (\{q\}, \Sigma, \delta, q, F)$. We will show that this language must be either $\emptyset$, $\{\epsilon\}$, and Δ^*.

If $F = \emptyset$ then $L(M) = \emptyset$. Otherwise, $F = \{q\} = Q$. In this case, if there are no transitions, then $L = \{\epsilon\}$. Otherwise, the only possible transitions are in the form $\delta(q, a) = \{q\}$. So, for each $a \in \Sigma$, either $\delta(q, a) = \{q\}$ or $\delta(q, a) = \emptyset$. If we define $\Delta = \{a \in \Sigma \ : \ \delta(q, a) = \{q\}\}$, then $L = \Delta^*$. $\square$

7.3 Deterministic and Nondeterministic Radius

Note that, since a DFA is simply a special case of an NFA, the nondeterministic radius of a language can be no more than its deterministic radius. However, there is no bound on the possible blowup from nondeterministic to deterministic radius.

Theorem 63 *For each $n > 0$, there exists a unary regular language L_n whose deterministic radius is n and whose nondeterministic radius is 1.*

Proof:

Let L_n be the language $\left\{a^k : k < n\right\}$.

This can be accepted by an NFA $M = (Q, \{a\}, \delta, q_0, F)$ of radius 1, where $Q = \{0, 1, \ldots, n-1\}$, $q_0 = 0$, $F = Q$, and δ is defined as follows:

$$\delta(i, a) = \begin{cases} \{1, 2, \ldots, n\}, & \text{when } i = 0; \\ \{i+1\}, & \text{when } 1 \leq i \leq n-2; \\ \emptyset, & \text{when } i = n-1. \end{cases}$$

See Figure 7.1 for a transition diagram for this NFA.

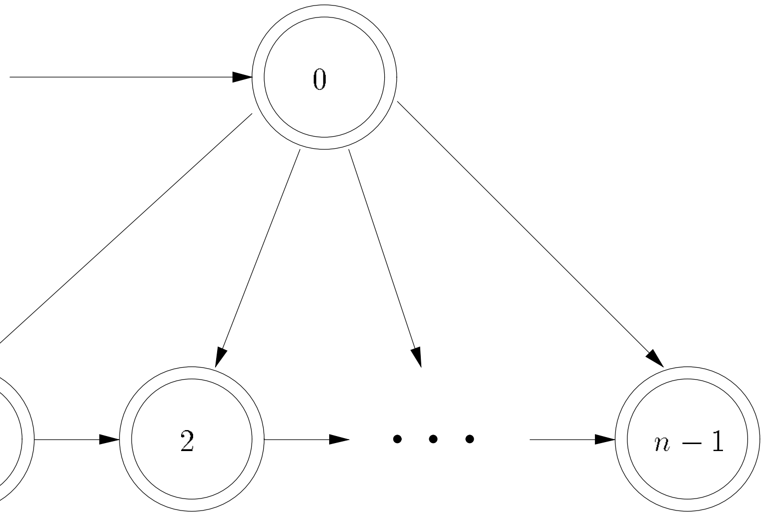

Figure 7.1: A radius–1 NFA that accepts L_n.

Since every state is final, this NFA accepts a word a^k if and only if there is a walk of length k from 0 to some state. (Note that a walk is simply a path that may use the same state more than once.) Clearly, this is true if and only if $k \leq n$. So the nondeterministic radius of L_n is 1.

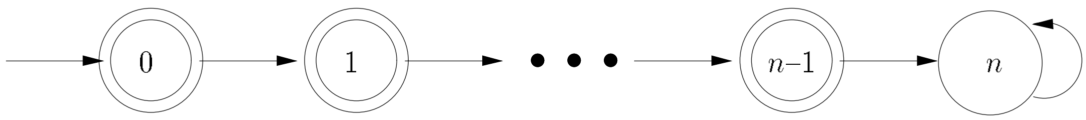

Figure 7.2: The minimal DFA for L_n.

However, consider the minimal DFA M_{min} which accepts L_n. The single non–final state in M_{min} can be reached from the initial state on any input $\{a^k : k \geq n\}$. This non–final state cannot be reached on a shorter input (since all shorter words are in L_n) so the machine has radius n, hence, the deterministic radius of L_n is n. $\square$

7.4 Unary Languages

For a regular language L over the unary alphabet $\{a\}$, define $\mathrm{gap}(L)$ to be the length n of the longest sequence $S = \{a^i, a^{i+1}, \ldots, a^{i+n-1}\}$ such that $S \cap L = \emptyset$ but $a^{i+n} \in L$. Then, $\mathrm{gap}(L)$ provides a lower bound for the nondeterministic radius of L. Specifically:

Lemma 64 *Let L be a unary regular language. Then* $\mathrm{gap}(L) \leq \mathrm{nrad}(L)$.

Proof:

Suppose that there is a unary language $L \subseteq \{a\}^*$ such that $\mathrm{gap}(L) = n$. So, there is some i such that $a^{i+n} \in L$ but $\{a^i, a^{i+1}, \ldots, a^{i+n-1}\} \cap L = \emptyset$. Consider an NFA $M = (Q, \Sigma, \delta, q_0, F)$ that accepts L. Since $a^{i+n} \in L$, there is some walk $q_0, q_1, \ldots, q_{i+n}$ where $q_{i+n} \in F$.

Suppose $\mathrm{nrad}(L) < n$. Then, by definition, for each $q \in Q$, $\mathrm{depth}(q) < n$. In particular, $\mathrm{depth}(q_n) < n$, and so there is a walk of length k (where $0 \leq k \leq n-1$) from q_0 to q_n. Also, we know that there is a walk $q_n, q_{n+1}, \ldots, q_{i+n}$ of length i from q_n to q_{i+n}, so, by concatenating these walks together, we get a walk of length $i+k$ (where $0 \leq k \leq n-1$) from q_0 to q_{i+n}, and so, since $q_{i+n} \in F$, it must be true that $a^{i+k} \in L$ for some $0 \leq k \leq n-1$, which is a contradiction. Thus, the nondeterministic radius of L is no less than $\mathrm{gap}(L)$.

□

Corollary 65 *For any choices of $k \geq 0$ and $n > 0$, the unary language $L = \{a^i : i \equiv k \pmod{n}\}$ has nondeterministic radius $n-1$.*

Proof:

From Lemma 64 we know that $\mathrm{nrad}(L) \geq n-1$ since $\{a^{k+1}, \ldots, a^{k+n-1}\} \cap L = \emptyset$ but $a^{k+n} \in L$.

Also, L is accepted by a single cycle of length n (with the kth state in the cycle being the only final state), and this has radius $n-1$. Thus, $\mathrm{nrad}(L) = n-1$ as desired. □

In the case where L is a finite language, $\mathrm{gap}(L)$ also gives us a tight upper bound on the nondeterministic radius of L.

Theorem 66 *Let L be a finite unary regular language over the alphabet $\Sigma = \{a\}$.*

Then there is an NFA M that accepts L such that:

$$\begin{aligned} |M| &= \mathrm{nsc}(L), \text{ and} \\ \mathrm{nrad}(M) &= \mathrm{gap}(L) + 1. \end{aligned}$$

Proof:

Suppose $L = \{a^{n_1}, a^{n_2}, \ldots, a^{n_k}\}$, where $n_1 < n_2 < \ldots < n_k$. Define $M = (Q, \{a\}, \delta, q_0, F)$ as follows:

$$\begin{aligned} Q &= \{q_0, q_1, \ldots, q_{n_k}\} \\ \text{for } i > 0,\ \delta(q_i, a) &= \{q_{i+1}\} \\ \delta(q_0, a) &= \{q_m : m = n_k - n_i + 1,\ 1 \le i \le k\} \\ F &= \begin{cases} \{q_{n_k}\}, & \text{if } \epsilon \notin L; \\ \{q_0, q_{n_k}\}, & \text{if } \epsilon \in L. \end{cases} \end{aligned}$$

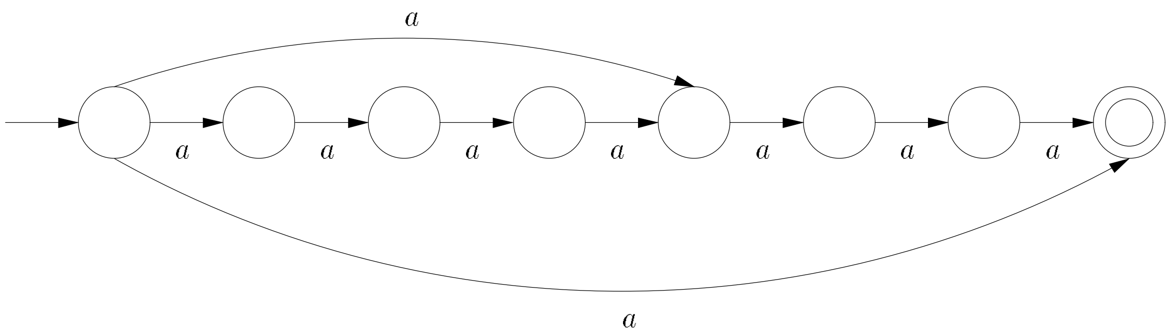

Figure 7.3: A minimal NFA for $L = \{a, a^4, a^7\}$.

A transition diagram representing this NFA for the language $\{a, a^4, a^7\}$ is given in Figure 7.3. To see that M is a minimal NFA for L, note that a^{n_k} is the largest word accepted by L. Thus, there must be a length n_k walk from the start state to some final state (since there is a word of length n_k in the language) and, there can be no cycles in this walk (since, if there was a cycle of length m in the walk, then a^{n_k+m} would also be in the language. Thus, this walk is actually a path, and thus contains $n_k + 1$ nodes. So, any NFA that accepts L requires at least $n_k + 1$ states, and so M is minimal.

Furthermore, note that for each state $q_i \in Q$ with $i > 0$, $\text{depth}(q_i) = \text{depth}(q_{i+1}) + 1$, unless if $i = n_k - n_j + 1$ for some $j < k$, in which case, $\text{depth}(q_i) = 1$. Therefore, since $\text{depth}(q_0) = 0$, the maximum value of $\text{depth}(q_i)$ is the maximum of $\{(n_k - n_j + 1) - (n_k - n_{j-1} + 1) : 1 \leq j \leq k\}$ where $n_0 = 0$. However, $(n_k - n_j + 1) - (n_k - n_{j-1} + 1) = n_j - n_{j-1}$ and so the maximum value of $\text{depth}(q_i)$ is (by definition) $\text{gap}(L) + 1$. Therefore M is the required minimal NFA with $\text{nrad}(M) = \text{gap}(L) + 1$. □

7.5 Operations on Regular Languages

As was the case for the other descriptional complexity measures that were studied in previous chapters, we would like to find tight upper bounds on the increase in nondeterministic radius when the usual regularity-preserving operations are applied.

7.5.1 Union

Theorem 67 *If L_1 and L_2 are regular languages with* $\text{nrad}(L_1) = m$ *and* $\text{nrad}(L_2) = n$, *then* $\text{nrad}(L_1 \cup L_2) \leq \max\{m, n\} + 1$.

Proof:

Suppose that $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ are NFAs of minimum radius for L_1 and L_2 respectively. Also assume that Q_1 and Q_2 are disjoint, and do not contain the state "q_0".

To establish the bound, we will use a construction similar to that used in the proof of Lemma 13 (see Figure 4.3). Define $Q = Q_1 \cup Q_2 \cup \{q_0\}$ and $F = F_1 \cup F_2$. For each $a \in \Sigma$, define δ as follows:

$$\delta(q, a) = \begin{cases} \delta_1(q, a), & \text{when } q \in Q_1; \\ \delta_2(q, a), & \text{when } q \in Q_2; \\ \emptyset & \text{when } q = q_0. \end{cases}$$

Also, let $\delta(q_0, \epsilon) = \{q_1, q_2\}$. So, $M = (Q, \Sigma, \delta, q_0, F)$ is an NFA–ϵ that accepts $L_1 \cup L_2$.

Note that for any word $w \in \Sigma^*$, $\delta^*(q_0, w) = \delta_1^*(q_1, w) \cup \delta_2^*(q_2, w)$. Thus, for any state q in Q_1 (or Q_2), the depth of q in M is the same as the depth of q in Q_1 (or Q_2). So it follows that $\text{rad}(M) = \max\{\text{rad}(M_1), \text{rad}(M_2)\}$. Since M is an NFA–ϵ, it follows from Lemma 59 that there exists an NFA M' that accepts $L_1 \cup L_2$ and has radius $\max\{\text{rad}(M_1), \text{rad}(M_2)\} + 1$. $\square$

In fact, this bound is the best that we can do, even in the unary case.

Theorem 68 *For some arbitrary $n > 1$, let L_1 be the unary language $(a^n)^*$, and let L_2 be the finite unary language $\{a\}$. Also, let $L = L_1 \cup L_2$. Then:*

(a) $\mathrm{nrad}(L_1) = n - 1$,

(b) $\mathrm{nrad}(L_2) = 1$, *and*

(c) $\mathrm{nrad}(L) \geq n$.

Proof:

Parts (a) and (b) follow from Corollary 65 and Theorem 66, respectively.

For part (c), suppose that $M = (Q, \Sigma, \delta, q_0, F)$ is an NFA of minimum radius that accepts L. We will show that the radius of M must be at least n.

Suppose that the radius of M is less than n. Then, for each state $q \in Q$, the depth of q must be less than n. Since $a^{2n} \in L$, there must be some state q and some final state q_F such that there is a walk of length n from q_0 to q, and a walk of length n from q to q_F. That is, $q \in \delta^*(q_0, a^n)$ and $q_F \in \delta^*(q, a^n)$.

However, since the depth of q must be less than n, there must also be a shorter walk from q_0 to q, that is, $q \in \delta^*(q_0, a^k)$ for some $k < n$. So, $q_F \in \delta^*(q_0, a^{n+k})$ for some $k < n$. Since q_F is a final state, that means that $a^{n+k} \in L$. Note that the 3 shortest words in L are a, a^n, and a^{2n}. Thus, since $k < n$, it must be that $k = 0$ and so $q = q_0$. That is, $q_0 \in \delta^*(q_0, a^n)$. But, $a \in L$ so there is another final state q_{F2} with $q_{F2} \in \delta(q_0, a)$. So, since $q_0 \in \delta^*(q_0, a^n)$, it is also true that $q_{F2} \in \delta^*(q_0, a^{n+1})$, and so $a^{n+1} \in L$, which is a contradiction. Thus, it is not true that the depth of q is less than n, and so the radius of M is at least n, as desired. $\square$

7.5.2 Intersection

There is no upper bound on the possible increase in nondeterministic radius when intersecting two regular languages.

Theorem 69 *For each $n > 0$, there are regular languages L_1 and L_2 over a two–letter alphabet such that:*

(a) $\mathrm{nrad}(L_1) = 0$,

(b) $\mathrm{nrad}(L_2) = 1$, *and*

(c) $\mathrm{nrad}(L_1 \cap L_2) = n$.

Proof:

For an arbitrarily chosen $n > 0$, let $L_1 = a^*$, and let $L_2 = \left\{ba^k \ : \ k < n\right\} \cup \{a^n\}$.

By Lemma 62, $\mathrm{nrad}(L_1) = 0$, so condition (a) holds.

To see that condition (b) holds, let $M = (Q, \{a, b\}, \delta, q_0, F)$ be the NFA shown in Figure 7.4. Formally:

- $Q = \{0, 1, \ldots, n\}$,

- $\delta(i, a) = \{i + 1\}$ for all $0 \leq i \leq n - 1$,
- $\delta(n, a) = \emptyset$,
- $\delta(0, b) = \{1, \ldots, n\}$,
- $\delta(i, b) = \emptyset$ for all $1 \leq i \leq n$,
- $q_0 = 0$, and
- $F = \{n\}$.

.

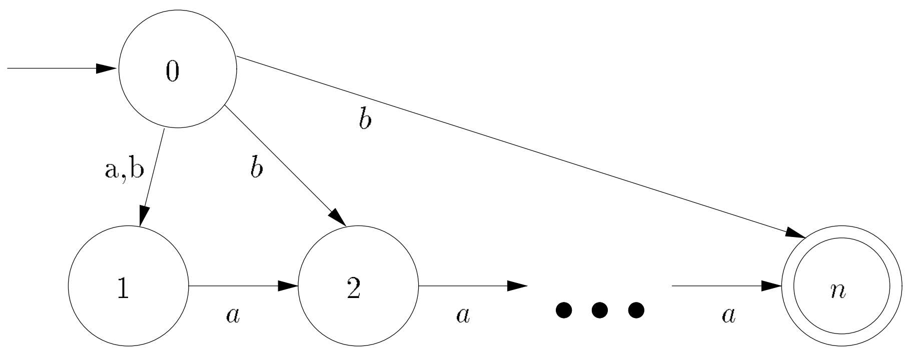

Figure 7.4: The radius–1 NFA M that accepts L_2.

Clearly, the radius of M is 1 and M accepts L_2. So, condition (b) is satisfied.

Note that $L_1 \cap L_2 = \{a^n\}$. So, in any NFA accepting $L_1 \cap L_2$, any final state must have depth at least n (since there are no words of length less than n in the language). Thus, condition (c) holds. $\square$

7.5.3 Concatenation

Theorem 70 *If* L_1 *and* L_2 *are regular languages with* $\text{nrad}(L_1) = m$ *and* $\text{nrad}(L_2) = n$*, then* $\text{nrad}(L_1 L_2) \leq m + n + 1$.

Proof:

To see that the bound holds, note that we can use the construction from Theorem 15 to create an NFA–ϵ of radius $m + n$ for $L_1 L_2$. So, by Lemma 59, there is an NFA of radius $m + n + 1$ that accepts $L_1 L_2$. □

And, as it turns out, this bound cannot be improved upon in general.

Theorem 71 *Let* $L_1 = a^*$ *and* $L_2 = b^*$*. Then:*

(a) $\text{nrad}(L_1) = \text{nrad}(L_2) = 0$*, and*

(b) $\text{nrad}(L_1 L_2) = 1$.

Proof:

Part (a) is a direct result of Corollary 65.

Since $a^* b^*$ cannot be accepted by a 1–state NFA, its nondeterministic radius must be at least 1. By Theorem 70 its nondeterministic radius is at most 1. This establishes part (b). □

And, in fact, this seemingly trivial example is the best that we can do. If the nondeterministic radius of one of the languages is non–zero, we have a better bound on the nondeterministic radius of their concatenation.

Theorem 72 *Suppose that L_1 and L_2 are regular languages of nondeterministic radius m and n respectively, with $m > 0$ or $n > 0$. Then* $\mathrm{nrad}(L_1 L_2) \leq m + n$, *and this bound is tight, even in the unary case.*

Proof:

Suppose that $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ are minimum–radius NFAs for L_1 and L_2 respectively. So $\mathrm{nrad}(M_1) = m$ and $\mathrm{nrad}(M_2) = n$. We will construct an NFA of radius $m + n$ that accepts $L_1 L_2$.

First, use the standard construction as described in Theorem 15 to construct an NFA–ϵ $M = (Q, \Sigma, \delta, q_0, F)$ that has radius $m + n$ and accepts $L_1 L_2$ (see Figure 7.5).

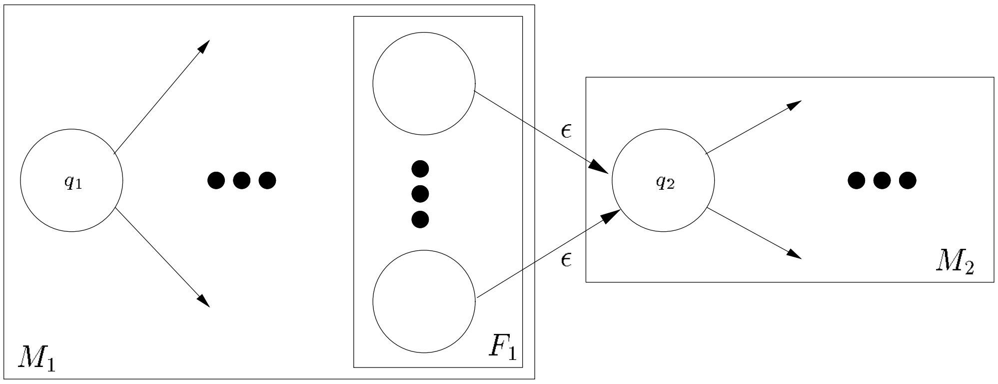

Figure 7.5: M, which has radius $m + n$ and accepts $L_1 L_2$

Formally,

- $Q = Q_1 \cup Q_2$,
- $\delta(q, a) = \delta_1(q, a)$ when $q \in Q_1$ and $a \in \Sigma$,

- $\delta(q, a) = \delta_2(q, a)$ when $q \in Q_2$ and $a \in \Sigma$,
- $\delta(q, \epsilon) = \{q_2\}$ when $q \in F_1$,
- $q_0 = q_1$, and
- $F = F_2$.

Now, we will construct an NFA $M' = (Q', \Sigma, \delta', q_0', F')$ of radius $m + n$ that accepts $L_1 L_2$. We can assume without loss of generality that $L_1 \cap \Sigma^+ \neq \emptyset$, and $L_2 \cap \Sigma^+ \neq \emptyset$. That is, each language contains a word of length at least 1. If this were not true then either:

(a) L_1 or L_2 is equal to $\emptyset$, and so $L_1 L_2 = \emptyset$, or

(b) $L_1 = \{\epsilon\}$ and so $L_1 L_2 = L_2$, or

(c) $L_2 = \{\epsilon\}$ and so $L_1 L_2 = L_1$.

In each case, the result follows trivially.

We can construct M' as follows:

- $Q' = Q$,
- $\delta'(q, a) = \delta^*(q, a)$ when $q \in Q'$ and $a \in \Sigma$,
- $q_0' = q_0$, and
- $F' = \begin{cases} F \cup \{q_0\}, & \text{when } \epsilon \in L_1 L_2; \\ F, & \text{otherwise.} \end{cases}$

Note that this is identical to the construction used in the proof of Lemma 59. As a result, it has the property that, for each word $w \in \Sigma^+$, $\delta'^*(q_0, w) = \delta^*(q_0, w)$. So, for a state $q \in Q = Q'$, we will denote the depth of q in M by $\mathrm{depth}_M(q)$, and the depth of q in M' by $\mathrm{depth}_{M'}(q)$.

Now, we will show that the nondeterministic radius of M' is no more than the nondeterministic radius of M, which is $m + n$. Consider a state $q \in Q'$. Let w be a minimum–length word from Σ^* such that $q \in \delta^*(q_0, w)$. So, by definition, $\mathrm{depth}_M(q) = |w|$.

There are two cases to consider. If $|w| > 0$ then $w \in \Sigma^+$ and so $q \in \delta'^*(q_0, w)$ and so $\mathrm{depth}_{M'}(q) \leq |w| = \mathrm{depth}_M(q) \leq \mathrm{nrad}(M) = m + n$.

Otherwise, $|w| = 0$ and so $w = \epsilon$. So $q \in \delta^*(q_0, \epsilon)$. However, we can see from the construction that the only ϵ–transitions in M are transitions from each $q_F \in F_1$ to q_2. Thus, either $q = q_0$ or $q = q_2$. If $q = q_0$ then $\mathrm{depth}_{M'}(q) = \mathrm{depth}_M(q) = 0 \leq m + n$. Otherwise $q = q_2$.

Now, there are 3 cases to consider. If $m = 0$ then $L_1 = \Delta^*$ for some nonempty $\Delta \subseteq \Sigma$ (since L_1 contains some word in Σ^+, and by Lemma 62). So choose some $a \in \Delta$, and $q \in \delta'(q_0, a)$, and so $\mathrm{depth}_{M'}(q) = 1 \leq m + n$ (since it is not true that $m = n = 0$). Similarly, if $n = 0$ then $L_2 = \Delta^*$ for some nonempty $\Delta \subseteq \Sigma$. So choose some $a \in \Delta$, and $q \in \delta'(q_0, a)$, and so $\mathrm{depth}_{M'}(q) = 1 \leq m + n$.

Finally, if $m, n > 0$ then let w' be the shortest word in $L_1 \cap \Sigma^+$. Recall that $\delta(q_F, \epsilon) = \{q\}$ for each $q_F \in F_1$. So, $q \in \delta^*(q_0, w')$, and $q \in \delta'^*(q_0, w')$. Therefore $\text{depth}_{M'}(q) \leq |w'|$. But, by Lemma 61, $|w'| \leq m + 1$. Thus, $\text{depth}_{M'}(q) \leq m + 1 \leq m + n$, since $m, n > 0$.

Therefore, for any state $q \in Q'$, the depth of q is no more than $m + n$. Thus, the radius of M' is no more than $m + n$ and so $\text{nrad}(L_1 L_2) \leq m + n$, as desired.

To see that the bound is tight, even in the unary case, choose m and n arbitrarily and let $L_1 = \{a^m\}$ and $L_2 = \{a^n\}$. So $L_1 L_2 = a^{m+n}$ and by Theorem 66:

- $\text{nrad}(L_1) = m$,
- $\text{nrad}(L_2) = n$, and
- $\text{nrad}(L_1 L_2) = m + n$,

which establishes that the bound is tight. □

In the unary case, we do not have to worry about the special case where both languages have a nondeterministic radius of 0.

Corollary 73 *If L_1 and L_2 are unary regular languages with $\text{nrad}(L_1) = m$ and $\text{nrad}(L_2) = n$, then $\text{nrad}(L_1 L_2) \leq m + n$.*

Proof:

Suppose L_1 and L_2 are regular languages over the unary alphabet Σ. If either $m > 0$ or $n > 0$ then by Theorem 72, the nondeterministic radius of L_1L_2 is no more than $m + n$.

Otherwise, $m = n = 0$. Then by Lemma 62, both L_1 and L_2 are one of $\emptyset$, $\{\epsilon\}$, or a^*. If one of L_1 or L_2 are $\emptyset$, then $L_1L_2 = \emptyset$ and $\mathrm{nrad}(L_1L_2) = 0 = m + n$. If $L_1 = \epsilon$ then $L_1L_2 = L_2$, which has nondeterministic radius $n = 0 = m+n$. Similarly, if If $L_2 = \epsilon$ then $L_1L_2 = L_1$, which has nondeterministic radius $m = 0 = m + n$. Finally, if both L_1 and L_2 are equal to a^*, then $L_1L_2 = a^*a^* = a^*$, which has nondeterministic radius $0 = m + n$. $\Box$

7.5.4 Kleene Closure

Theorem 74 *Suppose L is a regular language with nondeterministic radius n. Then* $\mathrm{nrad}(L^*) \leq n$*, and this bound is tight.*

Proof:

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a minimal-radius NFA for L. Then we use the same construction as was used in Lemma 16. That is, we modify M to create an NFA–ϵ $M' = (Q, \Sigma, \delta', q_0, F)$ by adding ϵ–transitions from each state in F to q_0. So, it is clear that M' accepts L^*, and has radius n. However, M' is an NFA–ϵ, not an NFA. We must show that we can convert M' to an NFA M'' with no increase in radius.

Note that the only ϵ–transitions in M' are transitions that lead to the start state. Thus, the only state of depth 0 is the start state, q_0. So, when we apply the standard construction to M' to create an equivalent NFA $M'' = (Q'', \Sigma, \delta'', q_0, F'')$ (see the proof of Lemma 59 for this construction) we see that every state in Q'' (except for the start state) had non-zero depth in M', and thus its depth is unchanged in M''. Of course, the depth of the start state is always 0, so that remains unchanged. Since each state in M'' has the same depth as it had in M', the radius of M'' cannot be more than the radius of M' (which is the same as the radius of M).

Thus, $\text{nrad}(L^*) \leq \text{nrad}(L)$ as desired.

To see that the bound is tight, simply note that there are languages L of arbitrary nondeterministic radius with $L = L^*$. For example, choose $L = (a^n)^*$ for some arbitrary n. So, since $L = L^*$, it follows that $\text{nrad}(L) = \text{nrad}(L^*)$. □

7.5.5 Reversal

Suppose L is a regular language. Then there is no upper bound for $\text{nrad}(L^R)$ in terms of $\text{nrad}(L)$.

Theorem 75 *For each $n > 0$, there is a regular language L_n such that:*

(a) the nondeterministic radius of L_n is 1, and

(b) the nondeterministic radius of L_n^R is n.

Proof:

Choose $n > 0$ arbitrarily, and let $L_n = \{a^i \ : \ 1 \leq i \leq n\} \cup \{ba^n\}$. Then the nondeterministic radius of L_n is 1, since L_n is accepted by the radius–1 NFA $M = (Q, \{a, b\}, \delta, q_0, F)$ (shown in Figure 7.6) where:

- $Q = \{0, 1, \ldots, n, n+1\}$,
- $\delta(0, a) = \{2, \ldots, n+1\}$,
- $\delta(0, b) = \{1\}$,
- $\delta(k, a) = \{k+1\}$ for all $1 \leq k \leq n$,
- $\delta(k, b) = \emptyset$ for all $1 \leq k \leq n+1$,
- $\delta(n+1, a) = \emptyset$,
- $q_0 = 0$, and
- $F = \{n+1\}$.

To see that the nondeterministic radius of L^R is n, note that applying the standard reversal construction to M (ie, reverse the direction of all the transitions and interchange the start and final state: see M_R in Figure 7.6) yields an NFA of radius n, so the nondeterministic radius of L^R is at most n. To see that the nondeterministic radius of L^R is at least n, note that:

(a) $a^n b \in L^R$, and

(b) $a^n b$ is the **only** word in L^R that ends in "b".

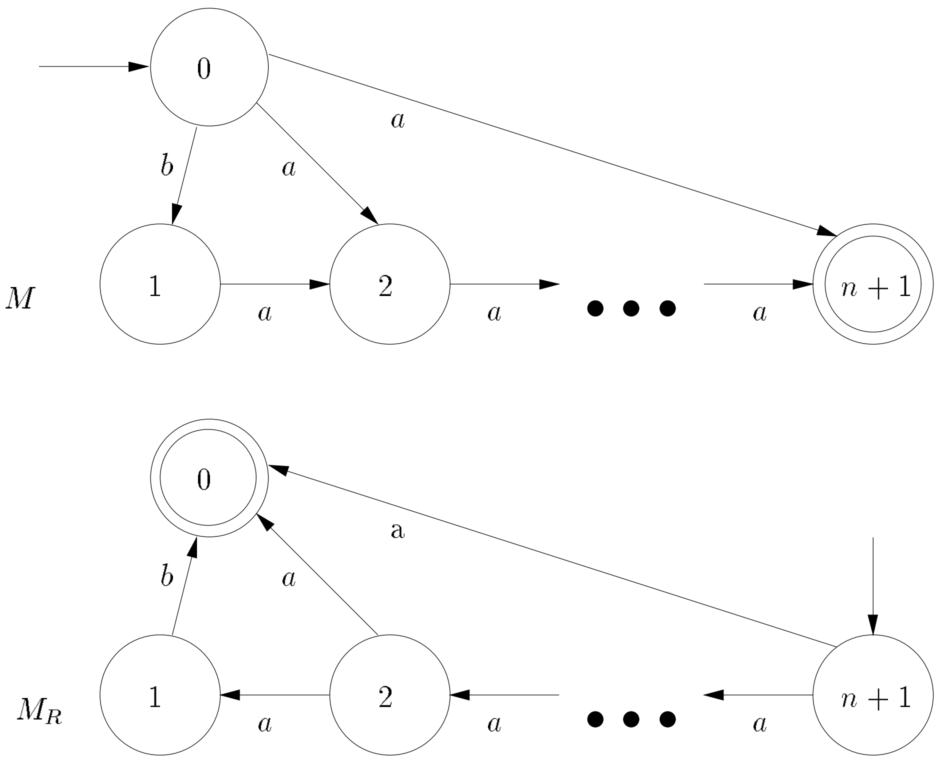

Figure 7.6: M has radius 1, and M_R has radius n.

So, let $M' = (Q', \{a, b\}, \delta', q_0', F')$ be the minimum–radius NFA that accepts L^R. Because of (a) above, there must be some $q \in Q'$ such that $q \in \delta'^*(q_0', a^n)$ and $\delta'(q, b) \cap F \neq \emptyset$. So, if the depth of q is less than n, there must be some word w with $|w| < n$ and $wb \in L$, which contradicts (b) above. Therefore the depth of q must be at least n, and so $\mathrm{nrad}(M') \geq n$, so $\mathrm{nrad}(L^R) \geq n$, as desired. $\square$

Of course, this is not true in the unary case since $L = L^R$ for any unary language L. Thus, if L is unary, $\mathrm{nrad}(L) = \mathrm{nrad}(L^R)$.

7.5.6 Complementation

It turns out that the possible increase in nondeterministic radius when complementing a regular language is unbounded, even in the unary case. This result is similar to Theorem 63.

Theorem 76 *For each n there is a unary regular language L_n of nondeterministic radius 1 such that the nondeterministic radius of $\overline{L_n}$ is n.*

Proof:

For each n, let $L_n = \{a^i \ : \ i < n\}$. As shown in Theorem 63, the nondeterministic radius of L_n is 1.

Note that $\overline{L_n} = \{a^i \ : \ i \geq n\}$. Since the shortest word in $\overline{L_n}$ is of length n, any final state in an NFA that accepts $\overline{L_n}$ must have depth at least n. Thus, the nondeterministic radius of $\overline{L_n}$ is at least n. □

7.6 Computability

For the first time, we will examine the topic of computability. Every other complexity measure that we have studied is trivially (even if inefficiently) computable. However, the problem is not trivial for nondeterministic radius. Since, for each $k > 0$, there are infinitely many languages of nondeterministic radius k (this is just a generalization of Theorem 63), we cannot simply determine the nondeterministic radius of a regular language by iterating through all NFAs of each radius until we find one that accepts our language. This leads to the following open problem:

Open Problem 14 *For a regular language L, is the nondeterministic radius of L computable?*

Chapter 8

Conclusions and Open Problems

8.1 State Complexity

State complexity has been extensively studied by many people. As a result, tight bounds on the increase in state complexity when many standard operations are applied to regular langauges are already known, for both the unary and non–unary cases [2, 24, 25, 26].

8.2 Nondeterministic State Complexity

Relationships between deterministic and nondeterministic state complexity have already been studied in both the unary case [5] and general case, [19, 20], and were quite well–known. However, we corrected a slight error in the generally accepted upper bound for the possible increase from nondeterministic to deterministic state complexity in the unary case.

There was a known tight bound on the increase in nondeterministic state complexity when taking the complement of a regular language (due to Birget [3]), although there was an error [4] in the proof that appeared in the original paper. For many other operations, we gave tight upper bounds in this thesis. Many of these results were simultaneously and independently discovered by Holzer and Kutrib [10, 11]. However, there are still some open problems. In particular, it is unknown whether or not the bound for concatenation in the unary case is tight. Also, we do not know whether the upper bound for complementation is achievable when the alphabet size is less than 4. Finally, it is an open problem whether the upper bound for left quotients is tight.

8.3 Regular Expression Size

We studied relationships between regular expression size and state complexity (both deterministic and nondeterministic). We gave a tight lower bound on regular expression size, in terms of nondeterministic state complexity (this result was due to Leiss [16]). We also gave an upper bound. However, it is an open problem whether this upper bound is acheivable. In the unary case, we gave a tight lower bound on regular expression size in terms of state complexity. We also gave an upper bound, but, once again, it is an open problem whether this bound is acheivable.

We gave tight upper bounds on the increase in regular expression size the operations of union, concatenation, and Kleene closure. For complementation, we gave a bound that is asymptotically tight in the unary case. However, for the general case, the bound that we gave is trivial, and there is an exponential gap between this bound and the largest acheivable increase in regular expression size. Therefore, it is an open problem whether the bound is tight, or whether there is a better bound. Similarly, we gave upper bounds for both the unary and general cases for intersection. However, it is currently unknown whether these bounds are tight. Finally, we gave exponential upper bounds for both left and right quotients, but it is an open problem whether these bounds are tight.

8.4 Radius

We studied the relationship between radius and state complexity. We showed that the DFA of minimum radius that accepts a language is, in fact, the minimal DFA for that language. Also, we gave tight upper and lower bounds on radius (in terms of state complexity).

Also, we gave upper bounds on the increase in radius for various operations. However, the bounds for union, intersection, concatenation, Kleene closure, and reversal are seemingly trivial. For each of these operations, it is an open problem whether the bound associated with that operation is tight. Complementation leaves radius unchanged, and we gave a tight bound for quotients.

8.5 Nondeterministic Radius

We studied relationships between deterministic and nondeterministic radius. We showed that the possible increase from nondeterministic radius to deterministic radius of a regular language is unbounded.

Additionally, we gave tight upper bounds on the increase in nondeterministic state complexity for the operations of union, concatenation, Kleene closure. We also showed that no such bounds exist for the operations of intersection, reversal, and complementation. Additionally, it is an open problem whether the nondeterministic radius of a regular language is computable.

Appendix A

Source Code

The following C++ code can be added into Grail (available at `http://www.csd.uwo.ca/research/grail/`) to allow it to calculate the radius of a finite automaton. This file can be named "`radius.src`". The documentation (which is included with the Grail software package) explains how this code may be integrated into Grail.

```
//  Compute the radius of a finite automaton.
//
//  Written by Keith Ellul

template <class Label>
int
fm<Label>::radius() const
{
    // Now copy code from reachable_states(), but count the
    // number of "waves"

    int     i;
    int     radius = 0;
```

```
set<state> wave = start_states;
set<state> s = wave;

s.sort();
arcs.sort();
for (;;)
{
    // find all targets reachable in one instruction

    set<state>      targets;
    set<state>      temp2;
    list<state>     temp3;
    fm<Label>       temp;

    temp3.clear();

    for (i=0; i<wave.size(); ++i)
    {
        select(wave[i], SOURCE, temp);
        temp.sinks(temp2);
        temp2.sort();
        temp3 += temp2;
    }

    targets.from_list(temp3);

    // if we've found new states, they will be used on
    // the next iteration

    wave.clear();
    for (i=0; i<targets.size(); ++i)
        if (s.member(targets[i]) < 0)
            wave.disjoint_union(targets[i]);

    if (wave.size() == 0)
        break;

    wave.sort();
```

```
        s.merge(wave);
        radius++;
    }

    return radius;
}
```

Bibliography

[1] J. Barzdin and A. Koršunov, *On the diameter of reduced finite automata*, Diskret. Analiz, 9 (1967), pp. 3–45.

[2] J.-C. Birget, *Intersection and union of regular languages and state complexity*, Information Processing Letters, 43 (1992), pp. 185–190.

[3] ——, *Partial orders on words, minimal elements of regular languages, and state complexity*, Theoretical Computer Science, 119 (1993), pp. 267–291.

[4] ——, *Erratum: Partial orders on words, minimal elements of regular languages, and state complexity.* personal communication, March 2002.

[5] M. Chrobak, *Finite automata and unary languages*, Theoretical Computer Science, 47 (1986), pp. 149–158.

[6] M. Domaratzki, *State complexity and proportional removals*, Proceedings of Descriptional Complexity of Automata, Grammars, and Related Structures, (2000), pp. 55–66.

[7] D. S. Dummit and R. M. Foote, *Abstract Algebra*, Prentice Hall, 2nd ed., 1999.

[8] A. Ehrenfeucht and P. Zeiger, *Complexity measures for regular expressions*, Journal of Computer and System Sciences, 12 (1976), pp. 134–146.

[9] K. Ellul, J. Shallit, and M.-W. Wang, *Regular expressions: New results and open problems*, Pre–Proceedings of Descriptional Complexity and Formal Systems, (2002), pp. 17–34.

[10] M. Holzer and M. Kutrib, *State complexity of basic operations on nondeterministic finite automata*, IFIG Research Report 0103, (2001).

[11] ——, *Unary language operations and their nondeterministic state complexity*, To Appear, DLT, (2002).

[12] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.

[13] T. Jiang and B. Ravikumar, *Minimal NFA problems are hard*, SIAM J. Comput., 22 (1993), pp. 1117–1141.

[14] E. Landau, *Über die Maximalordung der Permutationen gegebenen Grades*, Archiv. der Math. und Phys., 3 (1903), pp. 92–103.

[15] ——, *Handbuch der Lehre von der Verteilung der Primzahlen I*, Teubner, 1909.

[16] E. Leiss, *Constructing a finite automaton for a given regular expression*, SIGACT News, 12(3) (Fall 1980), pp. 81–87.

[17] H. Leung, *Separating exponentially ambiguous finite automata from polynomially ambiguous finite automata*, SIAM J. Comput., 27 (1998), pp. 1073–1082.

[18] R. McNaughton and H. Yamada, *Regular expressions and state graphs for automata*, IRE Trans. Electron. Comput., EC–9 (1960), pp. 39–47.

[19] F. 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), pp. 1211–1214.

[20] M. Rabin and D. Scott, *Finite automata and their decision problems*, IBM J. Res. Develop., 3 (1959), pp. 114–125.

[21] G. Rozenberg and A. Salomaa, eds., *Handbook of Formal Languages*, Springer, 1997.

[22] W. Sakoda and M. Sipser, *Nondeterminism and the size of two-way finite automata*, Proceedings of the Tenth ACM Symposium on the Theory of Computing, (1978), pp. 275–286.

[23] M. Szalay, *On the maximal order in S_n and S_n^**, Acta Arithmetica, 37 (1980), pp. 320–328.

[24] S. Yu, *State complexity of regular languages*, Proceedings of Descriptional Complexity of Automata, Grammars, and Related Structures, (1999), pp. 77–88.

[25] S. Yu and Q. Zhuang, *On the state complexity of intersection of regular languages*, SIGACT News, 22(3) (Summer 1991), pp. 52–54.

[26] S. Yu, Q. Zhuang, and K. Salomaa, *The state complexities of some basic*

operations on regular languages, Theoretical Computer Science, 125 (1994), pp. 315–328.