Michael Domaratzki | Alexander Okhotin

State complexity of power

Turku Centre *for* Computer Science

TUCS Technical Report
No 845, January 2008

State complexity of power

Michael Domaratzki
Department of Computer Science, University of Manitoba
Winnipeg R3T 2N2, Canada
mdomarat@cs.umanitoba.ca

Alexander Okhotin
Academy of Finland, *and*
Department of Mathematics, University of Turku, *and*
Turku Centre for Computer Science
Turku FIN–20014, Finland
alexander.okhotin@utu.fi

TUCS Technical Report
No 845, January 2008

Abstract

The number of states in a deterministic finite automaton (DFA) recognizing the language L^k, where L is regular language recognized by an n-state DFA, and $k \geqslant 2$ is a constant, is shown to be at most $n2^{(k-1)n}$ and at least $(n-k)2^{(k-1)(n-k)}$ in the worst case, for every $n > k$ and for every alphabet of at least six letters. Thus, the state complexity of L^k is $\Theta(n2^{(k-1)n})$. In the case $k = 3$ the corresponding state complexity function for L^3 is determined as $\frac{6n-3}{8}4^n - (n-1)2^n - n$ with the lower bound witnessed by automata over a four-letter alphabet. The nondeterministic state complexity of L^k is demonstrated to be nk. This bound is shown to be tight over a two-letter alphabet.

Keywords: descriptional complexity, finite automata, state complexity, combined operations, concatenation

TUCS Laboratory
Discrete Mathematics for Information Technology

1 Introduction

State complexity, which is the measure of the minimal number of states in any DFA accepting a given regular language, is one of the most well-studied descriptional complexity measures for formal languages; the topic has been an active research area for over ten years. Many results related to the state complexity of various operations on formal languages have been examined. For a recent survey of results related to this area, see Yu [15]. We note in particular that the state complexity of concatenation was obtained by Maslov [9] and further studied by Yu *et al.* [16] and Jirásek *et al.* [6], who determined the effect of the number of final states on the state complexity. The state complexity of concatenation over a unary alphabet was considered by Yu *et al.* [16] and subsequently by Pighizzini and Shallit [10], while Holzer and Kutrib [5] have studied the state complexity of concatenation with respect to nondeterministic finite automata (NFA).

The state complexity of combinations of basic operations was recently studied by A. Salomaa *et al.* [13] and subsequently examined in several papers [4, 7, 8, 14, 16]. In each result, a certain combination of operations over independent arguments is examined to determine its exact state complexity; in many cases, the state complexity of the combined operation is less than the direct composition of the deterministic state complexities of the individual operations.

As noted by K. Salomaa and Yu [14], an interesting research topic is the state complexity of combined operations with "non-linear variables", i.e., combined operations in which one or more operands are used in several positions in the expression. Rampersad [11] gives results on non-linear combined operations by studying the state complexity of powers of a language: L^k for $k \geqslant 2$. In particular, Rampersad shows that if the state complexity of L is n, then L^2 has state complexity at most $n2^n - 2^{n-1}$, and this bound can be reached for any $n \geqslant 3$ over an alphabet of size two. Rampersad also addresses the problem of the state complexity of L^k for $k \geqslant 3$ and unary languages L, but leaves the state complexity of L^k for $k \geqslant 3$ and arbitrary alphabets open.

In this paper, we consider this problem of the state complexity of L^k for L over an alphabet of size at least two. In particular, we show a general bound for the L^k which holds for any $k \geqslant 2$. A lower bound which is optimal up to a constant factor (with the constant depending on k) is given over a six-letter alphabet. For the state complexity of L^3, we show an improved upper bound and a matching lower bound over a four-letter alphabet.

Finally, we address the problem of nondeterministic state complexity of power. We show that if the nondeterministic state complexity of L is n, then the nondeterministic state complexity of L^k is nk for all $k \geqslant 2$, and give a matching lower bound over a binary alphabet.

2 Definitions

For additional background in formal language and automata theory, see Rozenberg and A. Salomaa [12]. Let Σ be a finite set of symbols, called letters. The set Σ is called an alphabet. A string over Σ is any finite sequence of letters from Σ. The empty string, which contains no letters, is denoted ε. The set Σ^* is the set of all strings over Σ. A language L is any subset of Σ^*. If $x = a_1 a_2 \cdots a_n$ is a string, with $a_i \in \Sigma$, then the length of x, denoted by $|x|$, is n.

Given languages $L_1, L_2 \subseteq \Sigma^*$, $L_1 L_2 = \{xy \ : \ x \in L_1, y \in L_2\}$ is the concatenation of L_1 and L_2. The k-th power of a language L is defined recursively as $L^1 = L$ and $L^k = LL^{k-1}$ for all $k \geqslant 2$.

A *deterministic finite automaton* (DFA) is a quintuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ is an alphabet, $\delta : Q \times \Sigma \to Q$ is the transition function, $q_0 \in Q$ is the distinguished start state and $F \subseteq Q$ is the set of final states. We extend δ to a function acting on $Q \times \Sigma^*$ in the usual way: $\delta(q, \varepsilon) = q$ for all $q \in Q$, and $\delta(q, wa) = \delta(\delta(q, w), a)$ for any $q \in Q$, $w \in \Sigma^*$ and $a \in \Sigma$.

A DFA $A = (Q, \Sigma, \delta, q_0, F)$ is said to be complete if δ is defined for all pairs $(q, a) \in Q \times \Sigma$. In this paper, we assume that all DFAs are complete.

A string w is accepted by A if $\delta(q_0, w) \in F$. The language $L(A)$ is the set of all strings accepted by A: $L(A) = \{w \in \Sigma^* \ : \ \delta(q_0, w) \in F\}$. A language L is *regular* if there exists a DFA A such that $L(A) = L$.

A *nondeterministic finite automaton* (NFA) is a quintuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q, Σ, q_0 and F are as in the deterministic case, but $\delta : Q \times \Sigma \to 2^Q$. The extension of δ to $Q \times \Sigma^*$ is accomplished by $\delta(q, \varepsilon) = q$ and $\delta(q, wa) = \cup_{q' \in \delta(q,w)} \delta(q', a)$. For an NFA A, $L(A) = \{w \in \Sigma^* \ : \ \delta(q_0, w) \cap F \neq \varnothing\}$. It is known that NFAs accept exactly the regular languages.

The *(deterministic) state complexity* of a regular language L, denoted $\mathrm{sc}(L)$, is the minimum number of states in any DFA which accepts L. Similarly, the *nondeterministic state complexity* of L is the minimum number of states in any NFA which accepts L, and is denoted by $\mathrm{nsc}(L)$.

Given a DFA $A = (Q, \Sigma, \delta, q_0, F)$, a state $q \in Q$ is said to be *reachable* if there exists a string $w \in \Sigma^*$ such that $\delta(q_0, w) = q$. Given two states $q_1, q_2 \in Q$, we say that they are *equivalent* if $\delta(q_1, w) \in F$ if and only if $\delta(q_2, w) \in F$ for all $w \in \Sigma^*$. If a pair of states is not equivalent, we say that they are *inequivalent*.

3 State complexity of L^k

In this section, we consider the state complexity of L^k, while treating the value k as a constant. We show an upper bound which is based on reachability

of states, while an explicit lower bound with respect to an alphabet of size six is also given. The upper and lower bound differ by a multiplicative factor of $2^{k(k-1)}\frac{n}{n-k} = \Theta(1)$.

3.1 Upper bound

Let L be a regular language such that $\mathrm{sc}(L) = n$ and every minimal DFA for L has f final states. Note that the construction of Yu *et al.* [16, Thm. 2.3] for concatenation gives the following upper bound on L^k for an arbitrary $k \geqslant 2$:

$$n2^{(k-1)n} - \frac{f(2^{nk}-1)}{2(2^n-1)} - \frac{f}{2}.$$

We now describe the construction of a DFA for L^k, which we use throughout what follows. Let $A = (Q, \Sigma, \delta, 0, F)$ be an arbitrary DFA. Assume without loss of generality that $Q = \{0, 1, \ldots, n-1\}$.

For a subset $P \subseteq Q$ and for $w \in \Sigma^*$, we use the notation $\delta(P, w) = \{\delta(p, w) \mid p \in P\}$.

The DFA for $L(A)^k$ is defined as $A^k = (Q_k, \Sigma, \delta_k, (0, \varnothing, \varnothing, \ldots, \varnothing), F_k)$, with the set of states $Q_k = Q \times (2^Q)^{k-1}$, and F_k consisting of all states $(i, P_1, P_2, \ldots, P_{k-1}) \in Q_k$ such that $P_{k-1} \cap F \neq \varnothing$.

The transition function $\delta_k : Q_k \times \Sigma \to Q_k$ is defined as follows: $\delta_k((i, P_1, P_2, \ldots, P_{k-1}), a) = (i', P_1', P_2', \ldots, P_{k-1}')$ where:

1. $i' = \delta(i, a)$.
2. if $i' \in F$, then $P_1' = \{0\} \cup \delta(P_1, a)$. Otherwise, $P_1' = \delta(P_1, a)$.
3. for all $1 \leqslant j \leqslant k-2$, if $P_j' \cap F \neq \varnothing$, then $P_{j+1}' = \{0\} \cup \delta(P_{j+1}, a)$. Otherwise, $P_{j+1}' = \delta(P_{j+1}, a)$.

According to this definition, it is easy to see that if $\delta_k((0, \varnothing, \ldots, \varnothing)) = (i, P_1, \ldots, P_{k-1})$, then $\delta(0, w) = i$ and further $\ell \in P_j$ if and only if there exists a factorization $w = u_0 u_1 \ldots u_{j_1} v$ with $u_0, u_1, \ldots, u_{j-1} \in L(A)$ and with $\delta(0, v) = \ell$. It follows that $L(A^k) = L(A)^k$.

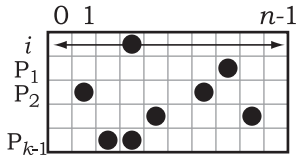

Figure 1: Representing states from Q_k as diagrams.

The above construction of A^k will be used throughout this paper. States from Q_k will be represented by diagrams as in Figure 1. Each row represents one of the k components of Q_k. The j-th row represents the j-th component;

accordingly, the top row is an element of Q, and all other rows represent a subset of Q. A solid dot will represent that a particular state is an element of the component: the left-most column represents state 0, the next left-most state 1, etc.

Since $|Q_k| = n2^{n(k-1)}$, the following upper bound on the state complexity of k-th power can be inferred:

Lemma 1. *Let L be a regular language with $sc(L) = n$ and let $k \geqslant 2$. Then the state complexity of L^k is at most $n2^{(k-1)n}$.*

3.2 Lower bound

In order to establish a close lower bound on the state complexity of k-th power, it is sufficient to present a sequence of automata $A_{k,n}$ $(2 \leqslant k < n)$ over the alphabet $\Sigma = \{a, b, c, d, e, f\}$, with every $A_{k,n}$ using n states, so that $L(A_{k,n})^k$ requires $\Omega(n2^{(k-1)n})$ states.

Let each $A_{k,n}$ have a set of states $Q = \{0, 1, \ldots, n-1\}$, of which 0 is the initial state, $n-1$ is the sole accepting state, and where the transitions are defined as follows:

$$\delta(j,a) = \begin{cases} j+1 & \text{if } 1 \leqslant j \leqslant n-k-1, \\ 1 & \text{if } j = n-k, \\ j & \text{otherwise,} \end{cases}$$

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

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

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

We now construct a DFA $(A_{k,n})^k$ for the language $L(A_{k,n})^k$ as described in Section 3.1. Its set of states is $Q_k = Q \times (2^Q)^{k-1}$.

Figure 2 shows the effect of the letters from Σ on states from Q_k. In particular, the letter a rotates the elements in the range $\{1, \ldots, n-k\}$ forward, and leaves the remaining states unchanged. The letter b rotates those states in the range $\{n-k+1, \ldots, n-1\}$ forward, and leaves all remaining states unchanged. An occurrence of the letter c swaps the states 0 and 1, leaving all others unchanged, while d collapses the state $n-k+1$ onto state 0, leaving all other elements unchanged. The letter e maps the state 0 onto the state $n-1$, as well as shifts those states in the range $\{n-k+1, \ldots, n-1\}$ back by

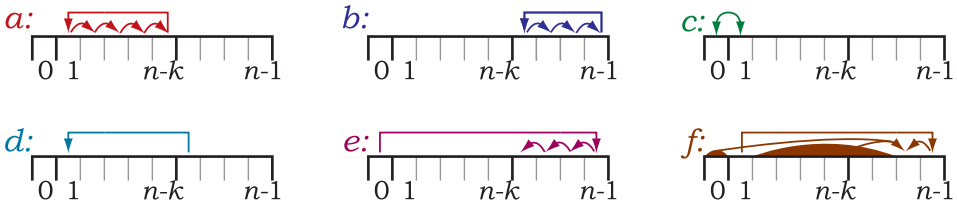

Figure 2: Transition table of $A_{k,n}$ and its action on $(A_{k,n})^k$. States with no arrow originating from them are unchanged by the letter.

one. Finally, the letter f collapses all states except 1 onto $n-2$, and maps 1 to $n-1$.

We also recall that, according to the construction of $(A_{k,n})^k$, $n-1 \in P_i$ implies $0 \in P_{i+1}$ for all $1 \leqslant i < k-1$. In the diagrams, this means that a state at the end of one row implies the existence of a state at the beginning of the next row (if such a row is present).

We now show the reachability and inequivalence of a large subset of states which will establish the lower bound. Lemmata 2 and 3 will establish that the states are reachable, and Lemma 4 shows that all such states are inequivalent.

Lemma 2. *Every state of the form* $(n-k+1, P_1, \ldots, P_{k-1})$*, where* $P_i \setminus \{1, \ldots, n-k\} = \{0, n-k+i+1\}$ *for all* $1 \leqslant i < k-1$ *and* $P_{k-1} \setminus \{1, \ldots, n-k\} = \{0\}$*, is reachable from the initial state.*

There are $2^{(k-1)(n-k)}$ such states, and their general form is presented in the diagram in Figure 3(b). In these diagrams, white areas without a dot indicate regions that are empty: no states are present in these regions. Grey areas in the diagram represent regions which may or may not be filled: any state in P_i in a grey region may or may not be present.

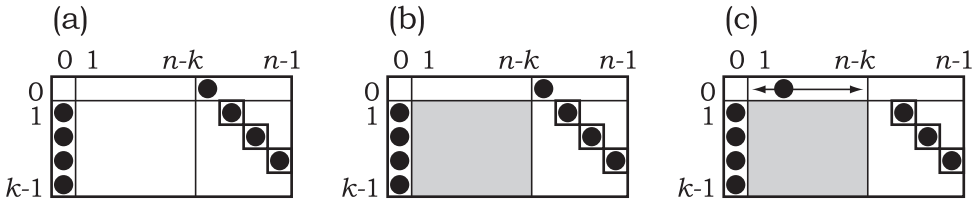

Figure 3: Outline of the reachability proof for L^k.

Proof. All states of this form will be reached by induction on the number of elements in the grey area, $\sum_{i=1}^{k-1} |P_i|$.

Basis: $\sum_{i=1}^{k-1} |P_i| = 2(k-2)+1$, that is, $P_i = \{0, n-k+i+1\}$ for all $i < k-1$ and $P_{k-1} = \{0\}$, see Figure 3(a). Then the state is $(n-k+1, \{0, n-k+2\}, \{0, n-k+3\}, \ldots, \{0, n-1\}, \{0\})$, and it is reachable from the initial state $(0, \varnothing, \ldots, \varnothing)$ by $e^{k-1}b^{k-1}$.

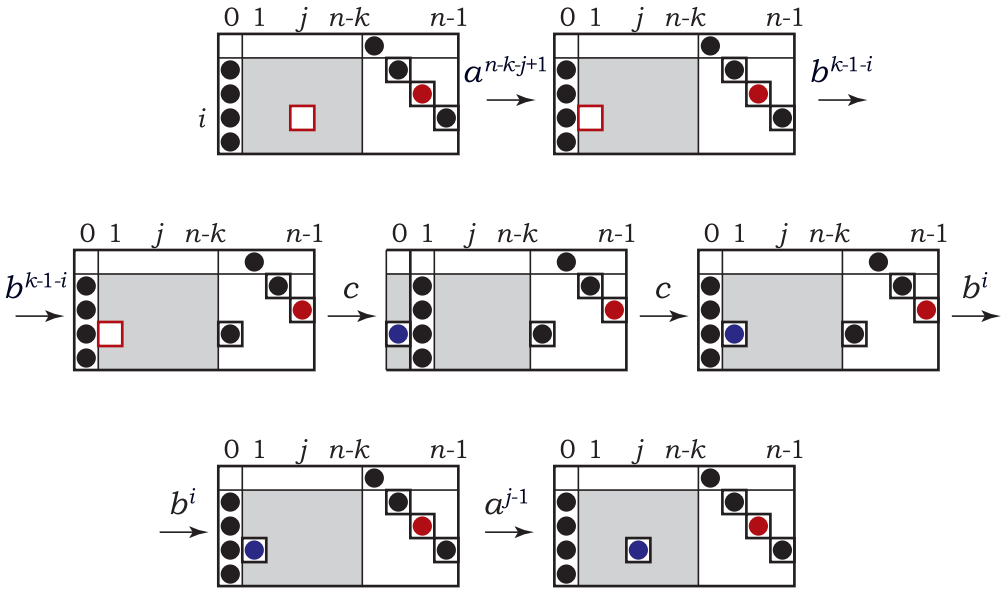

Figure 4: Adding j to P_i using the string $a^{n-k-j+1}b^{k-1-i}ccb^ia^{j-1}$.

Induction step: Let $(j_0, P_1, P_2, \ldots, P_{k-1})$ be an arbitrary state and let $j \notin P_i$ be arbitrary. In order to add j to P_i, we apply the string $a^{n-k-j+1}b^{k-1-i}ccb^ia^{j-1}$, as shown in Figure 4. The prefix $a^{n-k-j+1}$ rotates the empty square to column 1, the next substring b^{k-1-i} rotates the element $n-k+i-2 \in P_{i-1}$ to column $n-1$, then cc swaps columns 0 and 1 twice, effectively filling the empty square, and the suffix b^ia^{j-1} rotates the columns back to their original order. □

It remains to move the solid dot in the top row to any column among $1, \ldots, n-k$.

Lemma 3. *Every state of the form $(j_0, P_1, \ldots, P_{k-1})$, where $1 \leqslant j_0 \leqslant n-k$, $P_i \setminus \{1, \ldots, n-k\} = \{0, n-k+i+1\}$ for all $1 \leqslant i < k-1$ and $P_{k-1} \setminus \{1, \ldots, n-k\} = \{0\}$, is reachable from the initial state.*

There are $(n-k)2^{(k-1)(n-k)}$ such states, illustrated in the diagram in Figure 3(c), in which the arrow represents the range of j_0.

Proof. Let $(j_0, P_1, \ldots, P_{k-1})$ be an arbitrary state which satisfies the conditions of the lemma. We claim that there exists a reachable state $(n-k+1, P'_1, \ldots, P'_{k-1})$ such that, after reading da^{j_0-1}, we arrive in the state $(j_0, P_1, \ldots, P_{k-1})$. This will establish the lemma.

Define P'_i as follows. Take $P'_i \setminus \{1, \ldots, n-k\} = \{0, n-k+i+1\}$ for all $1 \leqslant i < k-1$ and $P'_{k-1} \setminus \{1, \ldots, n-k\} = \{0\}$, Next, $P'_i \cap \{1, \ldots, n-k\} = \{x - j_0 + 1 \ : \ x \in P_i\}$ where subtraction and addition is taken modulo $n-k$ in the range $\{1, \ldots, n-k\}$.

Then the state $(n-k+1, P'_1, \ldots, P'_{k-1})$ is reachable by Lemma 2. By d, the automaton goes from this state to $(1, P'_1, \ldots, P'_{k-1})$. Next, after application

of a^{j_0-1}, each P'_i is properly rotated (in the range $\{1,\ldots,n-k\}$) to P_i, that is, the automaton proceeds to $(j_0, P_1, \ldots, P_{k-1})$. □

Lemma 4. *All states of the above form are pairwise inequivalent.*

Proof. We first require the following three claims:

Claim 1. *Let $(j, P_1, \ldots, P_{k-1})$ be an arbitrary state and $1 \leqslant i \leqslant k-1$. After reading the string $(cf)^{k-i}$, the automaton $(A_{k,n})^k$ is in a final state if and only if $0 \in P_i$.*

Proof. The proof is by induction on i, starting with $i = k-1$.

For $i = k-1$, first suppose that $0 \in P_{k-1}$. Then after reading c, the automaton is in the state $(j', P'_1, \ldots, P'_{k-1})$ where $1 \in P'_{k-1}$. After reading f, the automaton is in state $(j'', P''_1, \ldots, P''_{k-1})$ where $n-1 \in P''_{k-1}$. This is a final state, as required.

Now, suppose that $0 \notin P_{k-1}$. After reading c, we are in the state $(j', P'_1, \ldots, P'_{k-1})$ where $1 \notin P'_{k-1}$. After reading f, the automaton is in state $(j'', P''_1, \ldots, P''_{k-1})$ where $n-1 \notin P''_{k-1}$, as f maps all states but 1 to the state $n-2$. Thus, as $n-1 \notin P''_{k-1}$, the state is not final.

Assume that the statement holds for all i with $\ell < i \leqslant k-1$. We now establish it for $i = \ell < k-1$. Assume first that $0 \notin P_\ell$. Again, after reading cf, we are in a state $(j', P'_1, \ldots, P'_{k-1})$ where $n-1 \notin P'_\ell$. Thus, cf does not add 0 to $P'_{\ell+1}$. On the other hand, the application of cf ensures that $P'_{\ell+1} \subseteq \{n-2, n-1\}$, since f maps all states into that pair, and 0 is not added to $P'_{\ell+1}$ after reading cf. Thus, $0 \notin P'_{\ell+1}$. By induction, after reading $(cf)^{k-\ell-1}$ from $(j', P'_1, \ldots, P'_{k-1})$ we are not in a final state.

Now assume that $0 \in P_\ell$. After reading cf, we can verify that we are in a state $(j', P'_1, \ldots, P'_{k-1})$ where $n-1 \in P_\ell$, and thus $0 \in P_{\ell+1}$. Now, by induction, after reading $(cf)^{k-\ell-1}$ we arrive in a final state. □

Claim 2. *For all j ($0 \leqslant j \leqslant n-1$), the string $a^{n-k-j+1}f(cf)^{k-1}$ is accepted from $(j_0, P_1, \ldots, P_{k-1})$ if and only if $j = j_0$.*

Proof. To establish this claim, first note that if $j = j_0$, then $a^{n-k-j+1}$ moves to a state $(1, P'_1, \ldots, P'_{k-1})$, which is then mapped to $(n-1, P''_1, \ldots, P''_{k-1})$ by f. Thus, after reading $a^{n-k-j+1}f$, we have $0 \in P''_1$. By Claim 1, after reading $(cf)^{k-1}$, we arrive in a final state.

On the other hand, if $j \neq j_0$, then $a^{n-k-j+1}$ maps j_0 to a state which is mapped to $n-2$ after reading f. Thus, after reading $a^{n-k-j+1}f$, the DFA is in a state $(n-2, P'_1, P'_2, \ldots, P'_{k-1})$ where $0 \notin P'_1$: we have just read f, which maps all elements to either $n-1$ or $n-2$, and the first component is not $n-1$, which would add 0 to P'_1. Again, using Claim 1, we can establish that upon reading $(cf)^{k-1}$, such a state $(n-2, P'_1, \ldots, P'_{k-1})$ proceeds to a non-final state. □

Claim 3. *For all i, j, with $1 \leqslant i \leqslant k-1$ and $0 \leqslant j \leqslant n-1$, the string $a^{n-k-j+1}f(cf)^{k-1-i}$ is accepted from a state $(j_0, P_1, \ldots, P_{k-1})$ if and only if $j \in P_i$.*

Proof. If $j \in P_i$, then reading $a^{n-k-j+1}$ moves to a state $(j'_0, P'_1, \ldots, P'_{k-1})$ where $1 \in P'_i$, which is subsequently mapped to a state $(j''_0, P''_1, \ldots, P''_{k-1})$ where $n-1 \in P''_i$ by f. Thus, if $i < k-1$, then $0 \in P''_{i+1}$. By Claim 1, after reading $(cf)^{k-i-1}$, $(j''_0, P''_0, \ldots, P''_{k-1})$ proceeds to a final state. Otherwise, if $i = k-1$, then $n-1 \in P''_{k-1}$, which is again a final state.

If $j \notin P_i$, then after reading $a^{n-k-j+1}f$, we move to a state $(j'_0, P'_1, \ldots, P'_{k-1})$ such that $n-1 \notin P'_i$. If $i = k-1$, the string is not accepted and we are finished. Otherwise, if $i < k-1$, as we have just read an f, we must have that $0 \notin P'_{i+1}$: $n-1 \notin P'_i$ and f maps every element to either $n-1$ or $n-2$. Again, by Claim 1, after reading $(cf)^{k-i-1}$, we arrive in a non-final state. □

With these three claims, we can easily establish that if $(j_0, P_1, P_2, \ldots, P_{k-1}) \neq (j'_0, P'_1, P'_2, \ldots, P'_{k-1})$, then there exists a string $w \in a^*f(cf)^*$ such that exactly one of the states leads to a final state on reading w. □

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

Proof. By Lemmata 1, 3 and 4. □

Corollary 1. *For every constant $k \geqslant 2$, the state complexity of L^k is $\Theta(n2^{(k-1)n})$.*

4 State complexity of L^3

The state complexity of L^2 is known precisely from Rampersad [11]. For the next power, the cube, Corollary 1 asserts that the state complexity of L^3 is $\Theta(n4^n)$, and Theorem 1 states in particular that it lies between $(n-3)4^{(n-3)}$ and $n4^n$ for each $n \geqslant 4$. We now obtain a precise expression for this function.

4.1 Upper bound

Let $A = (Q, \Sigma, \delta, 0, F)$ be an arbitrary DFA. Assume without loss of generality that $Q = \{0, 1, \ldots, n-1\}$. Recall from Section 3.1 the construction for A^k for $k = 3$. In particular, $A^3 = (Q_3, \Sigma, \delta_3, (0, \varnothing, \varnothing), F_3)$ with the set of states $Q_3 = Q \times 2^Q \times 2^Q$ and F_3 consisting of all states $(i, P, R) \in Q_3$ such that $R \cap F \neq \varnothing$.

The transition function $\delta_3 : Q_3 \times \Sigma \to Q_3$ is defined as follows: $\delta_3((i, P, R), a) = (i', P', R')$ where:

1. $i' = \delta(i, a)$.
2. if $i' \in F$, then $P' = \{0\} \cup \delta(P, a)$. Otherwise, $P' = \delta(P, a)$.
3. if $P' \cap F \neq \varnothing$, then $R' = \{0\} \cup \delta(R, a)$. Otherwise, $R' = \delta(R, a)$.

We now give an improved description of unreachable states in A^3. We will again use diagrams as in the case of L^k in Section 3 to represent states; in this case, as we are considering the cube, the diagrams will have three rows.

Lemma 5. *The following states in Q_3 are unreachable:*

(a) (i, P, R) *such that* $i \in F$ *and* $0 \notin P$.

(b) (i, P, R) *such that* $P \cap F \neq \varnothing$ *and* $0 \notin R$.

(c) $(i, \varnothing, R)$ *where* $R \neq \varnothing$.

Additionally, when there is only one final state and this final state is not initial (assume without loss of generality that it is state $n-1$), the following states are also unreachable:

(d) $(i, \{i\}, Q)$ *where* $0 \leqslant i < n - 1$.

(e) $(i, \{i\}, Q \setminus \{i\})$ *where* $0 \leqslant i < n - 1$.

(f) $(0, Q, \{0\})$.

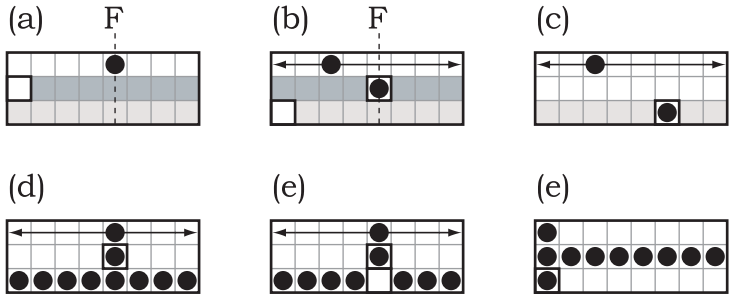

Figure 5: Unreachable states in Lemma 5.

Proof. Cases (a) and (b) follow immediately from the definition of δ_3: if a final state appears in a component, 0 must be added to the next component. Case (c) also follows from the definition of δ_3: elements of R can only be added when elements of P are already present, and once some states appear in P, they will never completely disappear, since the DFA is complete.

We now turn to case (d). Assume that $\delta_3((i', P', R'), a) = (i, \{i\}, Q)$ for some state (i', P', R') and some letter a. If $\delta(Q, a) = Q$, then we must have that $R' = Q$ and every state has a unique inverse image, and thus $P' = \{i'\}$. Thus, the preceding state (i', P', R') is $(i', \{i'\}, Q)$, which is of the same form.

Thus, assume that $\delta(Q, a) \neq Q$. Then $\delta(R', a) \neq Q$, so in order for $\delta_3((i', P', R'), a) = (i, \{i\}, Q)$, we must have that $i = n - 1$, contrary to our assumption on i.

Thus, the states $(i, \{i\}, Q)$ are unreachable from the start state: every state which leads into them is of the same form and thus also not reachable.

Case (e) is similar to case (d). Assume that $\delta_3((i', P', R'), a) = (i, \{i\}, Q \setminus \{i\})$ for some state (i', P', R') and some letter a. If $\delta(Q, a) = Q$, then we must have that $P' = \{i'\}$ and that $R' = Q \setminus \{i'\}$, which is again a state of the same form.

Thus, assume that $\delta(Q, a) \neq Q$. Note that $i' \notin R'$, otherwise $i \in \delta(R', a) = Q \setminus \{i\}$. Thus, $|R'| \leqslant n - 1$. Notice that $\delta(R', a) \neq Q \setminus \{i\}$ implies that $i = n - 1$, contrary to our choice of i. Thus, $\delta(R', a) = Q \setminus \{i\}$, we must actually have $|R'| = n - 1$. Therefore, $\delta(i', a) = i$ and thus, taken with $\delta(R', a) = Q \setminus \{i\}$, we have that $\delta(Q, a) = Q$, contrary to our assumption.

Finally, for case (f), consider the state $(0, Q, \{0\})$. Let $(j, P, R) \in Q_3$ and $a \in \Sigma$ be such that $\delta_3((j, P, R), a) = (0, Q, \{0\})$. As $0 \notin F$, we have that $Q = \delta(P, a)$. Thus, it must be that $P = Q$. But now, j is the unique state such that $\delta(j, a) = 0$ and $R = \{j\}$. Thus, $\delta_3((j, Q, \{j\}), a) = (0, Q, \{0\})$. If $j \neq 0$, then the state $(j, Q, \{j\})$ is already unreachable by case (b). Thus, the only other possibly reachable state leading to $(0, Q, \{0\})$ is itself, and the state is unreachable. □

Note that Lemma 5 does not consider the case of the initial state being the unique accepting state. This case is in fact trivial in terms of state complexity, which will be discussed in the proof of Lemma 6 below.

Lemma 6. *The state complexity of L^3 is at most*

$$\frac{6n - 3}{8} 4^n - (n - 1)2^n - n \quad (\textit{for } n \geqslant 3). \tag{1}$$

Proof. Let A be a DFA with n states and f final states.

We first note that if A has only one final state, we may assume without loss of generality that it is not the initial state. Indeed, if the lone final state is also the initial state, then $L(A) = L(A)^*$. Thus $L(A)^k = L(A)^*$ for all $k \geqslant 1$, and the state complexity is unaffected by taking powers (and (1) obviously holds). Therefore, in what follows, in the cases where A has only one final state we assume that it is not the initial state.

Consider first the case for more than one final state. Then the conditions (a), (b) and (c) from Lemma 5 are applicable. The total number of states is $n4^n$. We can also count the number of unreachable states:

(a) $f2^{2n-1}$ states of the form (i, P, R) such that $i \in F$ and $0 \notin P$.

(b) If $0 \notin F$, there are $n(2^f - 1)2^{2n-f-1} - f(2^f - 1)2^{2n-f-2}$ states of the form (i, P, R) such that $P \cap F \neq \varnothing$ and $0 \notin R$ not already excluded by (a).

On the other hand, if $0 \in F$, there are $n(2^f - 1)2^{2n-f-1} - f(2^{f-1} - 1)2^{2n-f-1}$ states of the form (i, P, R) such that $P \cap F \neq \varnothing$ and $0 \notin R$ not already excluded by (a).

(c) $(n-f)(2^n-1)$ states of the form $(i, \varnothing, R)$ not already excluded by (a).

The refined total of reachable states in the case that $0 \notin F$ is:

$$n4^n - f2^{2n-1} - (2^f - 1)(2^{2n-f-2})(2n - f) - (n - f)(2^n - 1). \qquad (2)$$

In the case where $0 \in F$, it is

$$n4^n - f2^{2n-1} - ((2n - f)(2^{f-1}) - (n - f))(2^{2n-f-1}) - (n - f)(2^n - 1). \quad (3)$$

For one final state, cases (d),(e) and (f) of Lemma 5 yield an additional $2(n-1)+1 = 2n-1$ states which are unreachable. Thus, the total for one final state (which is not the initial state by assumption) is, using (2),

$$n4^n - 2^{2n-1} - 2^{2n-3}(2n - 1) - (n - 1)(2^n - 1) - 2n + 1. \qquad (4)$$

Simplifying the above, we get the expression $\frac{6n-3}{8}4^n - (n-1)2^n - n$.

Now, consider the case of $f \geqslant 2$: we can easily verify that $f(2^{f-1} - 1)2^{2n-f-1} < f(2^f - 1)2^{2n-f-2}$, and hence the expression in (3) is larger than (2). Thus, in order to show that (4) is the true upper bound, we must show that it is larger than (3). That is, we must show that the inequality

$$\begin{aligned} & n4^n - f2^{2n-1} - ((2n - f)(2^{f-1}) - (n - f))(2^{2n-f-1}) - (n - f)(2^n - 1) \\ \leqslant\ & \frac{6n-3}{8}4^n - (n-1)2^n - n \end{aligned}$$

holds for all $n \geqslant 3$ and $2 \leqslant f \leqslant n-1$.

Rewriting the left-hand side of the inequality, we get

$$\begin{aligned} & n4^n - f2^{2n-1} - \big((2n - f)(2^{f-1}) - (n - f)\big)2^{2n-f-1} - (n - f)(2^n - 1) \\ =\ & 4^n\left(n - \frac{f}{2} - \frac{(2n - f)2^{f-1} - (n - f)}{2^{f+1}} - \frac{n-f}{2^n}\right) + (n - f) \\ =\ & 4^n\left(n - \frac{f}{2} - \frac{2n - f}{4} + \frac{n-f}{2^{f+1}} - \frac{n-f}{2^n}\right) + (n - f) \\ =\ & 4^n\left(\frac{n}{2} - \frac{f}{4} + \frac{n-f}{2^{f+1}} - \frac{n-f}{2^n}\right) + (n - f). \end{aligned}$$

We can now approximate this quantity as follows, using the inequalities $n \geqslant 3$ and $2 \leqslant f \leqslant n-1$:

$$\begin{aligned} & 4^n\left(\frac{n}{2} - \frac{f}{4} + \frac{n-f}{2^{f+1}} - \frac{n-f}{2^n}\right) + (n-f) \\ \leqslant\ & 4^n\left(\frac{n}{2} - \frac{1}{2} + \frac{n-2}{8}\right) + (n-2) = 4^n\left(\frac{5n-6}{8}\right) + (n-2). \end{aligned}$$

Now, note that subtracting this quantity from the right-hand side of the original inequality gives

$$\left(\frac{n+3}{8}\right) 4^n - (n-1)2^n - 2n + 2.$$

It is now easy to verify that this quantity is at least zero for all $n \geqslant 3$. □

4.2 Lower bound

We now turn to showing that the upper bound in Lemma 6 is attainable over a four-letter alphabet. Consider a sequence of DFAs $\{A_n\}_{n\geqslant 3}$ defined over the alphabet $\Sigma = \{a, b, c, d\}$, where each automaton A_n has the set of states $Q = \{0, \ldots, n-1\}$, of which 0 is the initial state and $n-1$ is the only accepting state, while the transition function is defined as follows:

$$\delta(i,a) = \begin{cases} i+1 & \text{if } 0 \leqslant i \leqslant n-3, \\ 1 & \text{if } i = n-2, \\ n-1 & \text{if } i = n-1, \end{cases} \qquad \delta(i,b) = \begin{cases} 0 & \text{if } i = 0, \\ i+1 & \text{if } 1 \leqslant i \leqslant n-2, \\ 1 & \text{if } i = n-1, \end{cases}$$

$$\delta(i,c) = \begin{cases} n-1 & \text{if } i = 0, \\ i & \text{if } 1 \leqslant i \leqslant n-1, \\ 0 & \text{if } i = n-1, \end{cases} \qquad \delta(i,d) = \begin{cases} i & \text{if } 0 \leqslant i \leqslant n-2, \\ 0 & \text{if } i = n-1. \end{cases}$$

The form of these automata is illustrated in Figure 6. Note that the transition tables for a, b and c are permutations of the set of states, and therefore, for every $\sigma \in \{a, b, c\}$, one can consider its inverse $\sigma^{-1} : Q \to Q$. Denote by $\sigma^{-1}(j)$ for $j \in Q$, the unique state k such that $\delta(k, \sigma) = j$. One can consider sequences of negative symbols; for any $\ell \geqslant 0$ denote by $\sigma^{-\ell}(j)$ the unique state k with $\delta(k, \sigma^\ell) = j$.

This notation is naturally extended to sets of states: for any set $P \subseteq \{0, \ldots, n-1\}$, for any letter $\sigma \in \{a, b, c\}$ and for any $\ell \geqslant 0$, we use the notation $\sigma^{-\ell}(P)$ to denote the uniquely defined set $P' \subseteq \{0, \ldots, n-1\}$ such that $\delta(P', \sigma^\ell) = P$.

We use the construction for $(A_n)^3$ given in Section 3.1. We also again use diagrams as in the case of L^k in Section 3 to represent states.

We now establish three lemmas to show reachability of states in $(A_n)^3$: first those states whose third component is empty, then those of the form (i, P, R) where $i \notin P$, and finally those with $i \in P$.

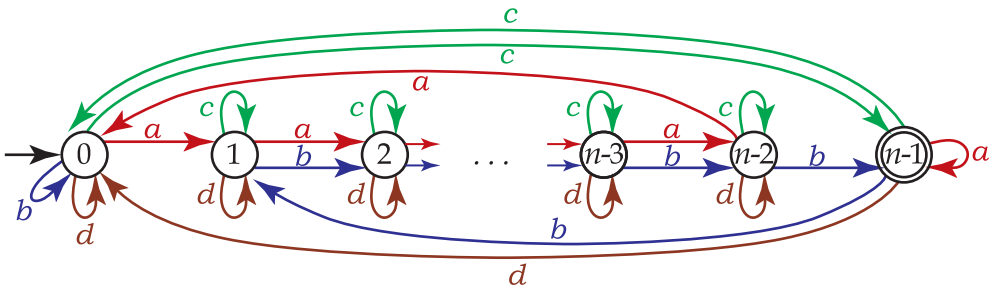

Figure 6: Witness automata A_n for cube.

Lemma 7. *Every state of the form $(i, P, \varnothing)$, where $n-1 \notin P$, $i \notin P$, and if $i = n-1$ then $0 \in P$, is reachable by a string from $\{a, b\}^*$.*

Proof. Induction on $|P|$.

Basis: $P = \varnothing$. A state $(i, \varnothing, \varnothing)$ with $0 \leqslant i < n-1$ is reachable via a^i from the start state $(0, \varnothing, \varnothing)$.

Induction step. First consider the case of $i = n-1$, that is, of a state $S = (n-1, P, \varnothing)$ with $0 \in P$ and $n-1 \notin P$. If $1 \notin P$, then, as shown in Figure 7(i), S is reachable from $(b^{-1}(n-1), b^{-1}(P \setminus \{0\}), \varnothing)$ by b, while the latter state is reachable according to the induction hypothesis, as $|b^{-1}(P \setminus \{0\})| < |P|$. If $1 \in P$, then S is reachable by a from $(n-1, b^{-1}(P \setminus \{0\}), \varnothing)$, which is in turn reachable by the induction hypothesis; see Figure 7(ii).

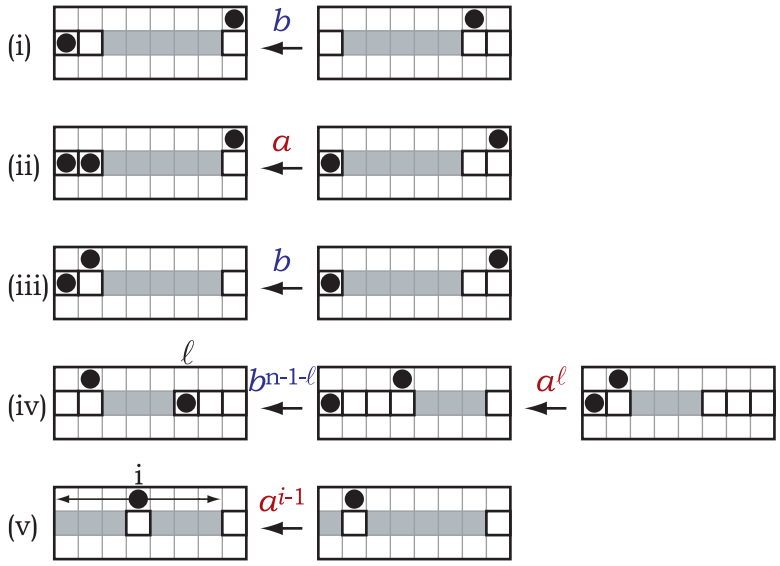

Figure 7: Reachability of states $(i, P, \varnothing)$ in Lemma 7.

The next case is $i = 1$: consider any state $S = (1, P, \varnothing)$ with $1, n-1 \notin P$. If $0 \in P$ (Figure 7(iii)), then S is reachable from $(n-1, \{0\} \cup b^{-1}(P \setminus \{0\}), \varnothing)$ by b, where the latter state was shown to be reachable in the previous case. If $0 \notin P$, consider the greatest number ℓ with $\ell \in P$. The state $(1, \{0\} \cup (P \setminus$

$\{\ell\}), \varnothing)$ is reachable as shown in the previous subcase, and it goes to S by $b^{n-1-\ell}a^\ell$, see Figure 7(iv).

Finally, any state $S = (i, P, \varnothing)$ with $0 \leqslant i \leqslant n-2$ and $n-1 \notin P$ is reachable from the state $(1, a^{-(i-1)}(P), \varnothing)$ by a^{i-1}, which is shown in Figure 7(v). States of the latter form have been shown reachable in the previous case. □

Lemma 8. *Every state of the form (i, P, R), where $|P| \geqslant 1$; $i \notin P$; if $i = n-1$, then $0 \in P$; if $n-1 \in P$, then $0 \in R$, is reachable.*

Proof. Induction on $|R|$. The basis, $R = \varnothing$, is given by Lemma 7.

Induction step: We have three major cases, each of which is broken into several subcases.

Case 1: $n-1 \in P$**.**

Case 1(a): $1 \notin P$, $i \neq 1$. Then the state $(b^{-1}(i), b^{-1}(P), b^{-1}(R \setminus \{0\}))$ is reachable by the induction hypothesis, and from it the state (i, P, R) is reachable by b.

Case 1(b): $1 \notin P$, $i = 1$, $0 \notin P$. Then (i, P, R) is reachable from $(c^{-1}(i), c^{-1}(P), c^{-1}(R \setminus \{n-1\}))$ by c, where the latter state is reachable by the induction hypothesis.

Case 1(c): $1 \notin P$, $i = 1$, $0 \in P$. Then (i, P, R) is reachable from $(n-1, b^{-1}(P), b^{-1}(R \setminus \{0\}))$ by b.

Case 1(d): $1 \in P$. Let j be the greatest number, such that $1, \ldots, j \in P$. Then either $i > j$ or $i = 0$, and in each case (i, P, R) is reachable from $(b^{-j}(i), b^{-j}(P), b^{-j}(Q))$ by b^j. The latter state has $n-1 \in b^{-j}(P)$ and $1 \notin b^{-j}(P)$, and hence it has been proved reachable in Cases 1(a)–1(c).

Case 2: $n-1 \notin P$**,** $n-1 \in R$**.**

Case 2(a): $0 \in P$. This state is reachable by c from $(c^{-1}(i), c^{-1}(P), c^{-1}(R))$, which has $n-1 \in c^{-1}(P)$ and is therefore reachable as in Case 1.

Case 2(b): $0 \notin P$. Let j be the least number in P. Then this state is reachable by a^j from $(a^{-j}(i), a^{-j}(P), a^{-j}(R))$, which is reachable as in Case 2(a).

Case 3: $n-1 \notin P$**,** $n-1 \notin R$**.**

Case 3(a): $0 \in P$, $0 \in R$. This case is further split into three subcases depending on the cardinality of P and R:

3(a^1). First assume $|P| \geqslant 2$ and let j be the least element of $P \setminus \{0\}$. Then (i, P, R) is reached by b^j from $(b^{-j}(i), b^{-j}(P), b^{-j}(R))$, which is in turn reachable as in Case 1, since $n-1 \in b^{-j}(P)$.

3(a^2). Similarly, if $|R| \geqslant 2$, then setting j as the least element of $R \setminus \{0\}$ one can reach (i, P, R) by b^j from $(b^{-j}(i), b^{-j}(P), b^{-j}(R))$, which is reachable as in Case 1 or Case 2.

3(a^3). The remaining possibility is $|P| = |R| = 1$, that is, $P = \{0\}$ and $R = \{0\}$. Such a state $(i, \{0\}, \{0\})$, with $1 \leqslant i \leqslant n-1$, is reachable by

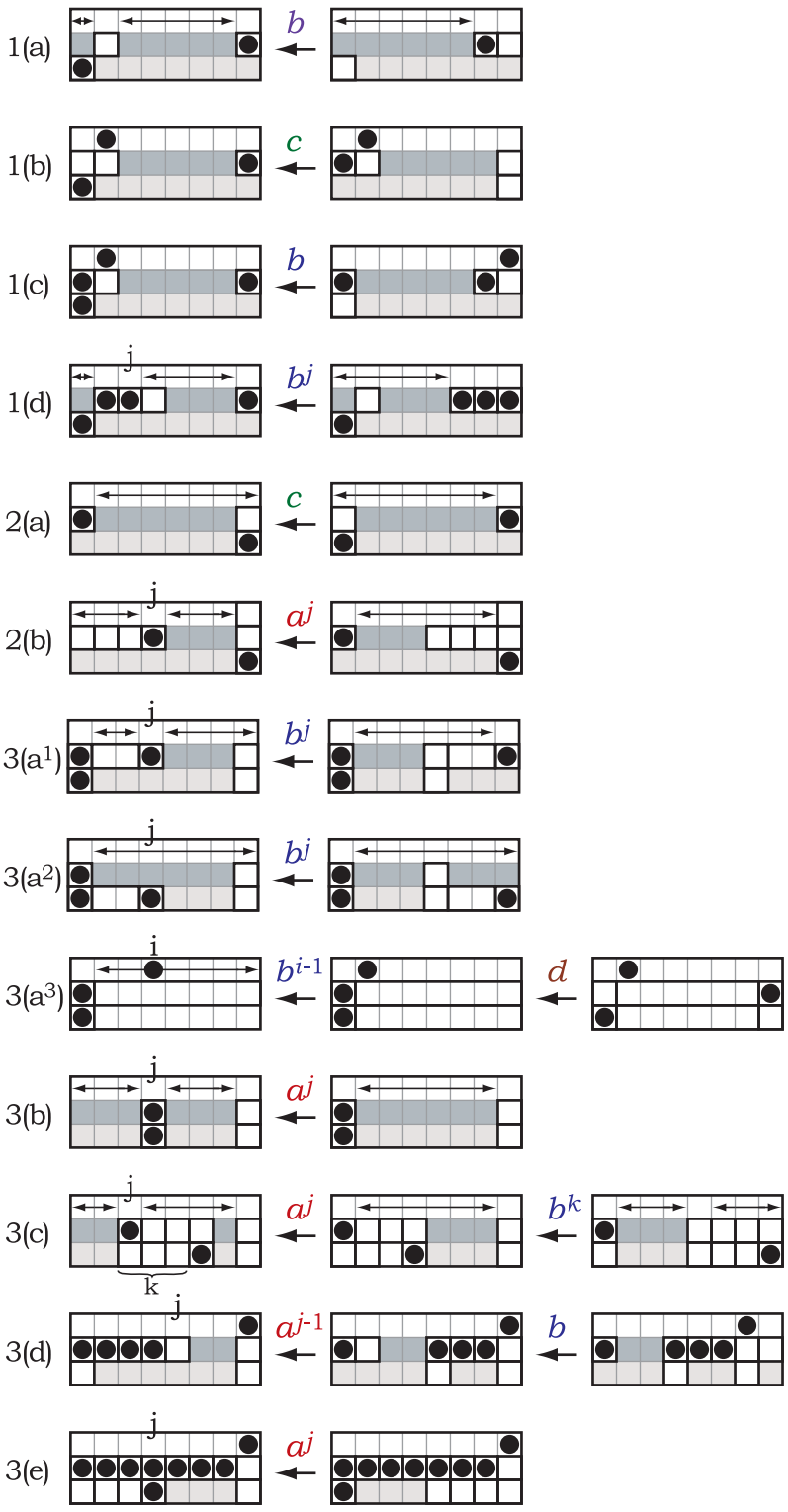

Figure 8: Reachability of (i, P, R): cases in the proof of Lemma 8.

db^{i-1} from $(1, \{n-1\}, \{0\})$, while the latter state was shown reachable in Case 1(b).

Case 3(b): $0 \leqslant i \leqslant n-2$, $P \cap R \neq \varnothing$. Let $j \in P \cap Q$ be the least such number. Then this state is reachable by a^j from $(a^{-j}(i), a^{-j}(P), a^{-j}(R))$, which is reachable as in Case 3(a).

Case 3(c): $0 \leqslant i \leqslant n-2$, $P \cap R = \varnothing$. Since $P, R \neq \varnothing$, there exists at least one pair (j, k) with $j \in P$ and $j + k \pmod{n-1} \in R$. Consider any such pair. Then this state is reachable by $b^k a^j$ from $(b^{-k}(a^{-j}(i)), b^{-k}(a^{-j}(P)), b^{-k}(a^{-j}(R)))$, which is itself reachable as in Case 2(a).

Case 3(d): $i = n-1$, $P \neq \{0, 1, \ldots, n-2\}$. Assume $0 \notin R$ (if $0 \in R$, then this state falls under Case 3(a)). Let j be the least number not in P. Then this state is reachable by a^{j-1} from the state $(n-1, a^{-(j-1)}(P), a^{-(j-1)}(R))$, which has $1 \notin a^{-(j-1)}(P)$ and is therefore reachable by b from $(n-2, b^{-1}(a^{-(j-1)}(P)), b^{-1}(a^{-(j-1)}(R)))$. The latter state has $n-1 \notin b^{-1}(a^{-(j-1)}(P))$; if $n-1 \notin b^{-1}(a^{-(j-1)}(R))$, it is reachable as in Case 3(b) or in Case 3(c), and if $n-1 \in b^{-1}(a^{-(j-1)}(R))$, it is reachable as in Case 2.

Case 3(e): $i = n-1$, $P = \{0, 1, \ldots, n-2\}$. Again, it can be assumed that $0 \notin R$. This time define j as the least number in R, which exists since $R \neq \varnothing$. In this case the state in question is reachable by a^j from $(n-1, P, a^{-j}(R))$, which is reachable as in Case 3(a). □

Lemma 9. *Every state of the form (i, P, R), where*

I. $i \in P$;

II. $|P| \geqslant 1$;

III. *if* $i = n-1$, *then* $0 \in P$;

IV. *if* $n-1 \in P$, *then* $0 \in R$;

V. *if* $P = \{i\}$, *then* $R \neq Q$ *and* $R \neq Q \setminus \{i\}$.

VI. *if* $i = 0$ *and* $P = Q$, *then* $R \neq \{0\}$.

is reachable.

Note that the last two conditions of Lemma 9 exactly match the last three cases of Lemma 5.

Proof. The proof again involves examining several cases, though this time there is no induction. The first case is based upon Lemma 8, the other cases depend on the first case and on each other. All cases except the last Case 4 deal with $i \neq n-1$: Case 1 assumes $n-1 \notin P$ and $n-1 \notin R$, Case 2 uses $n-1 \in P$ and Case 3 handles the last possibility: $n-1 \notin P$ and $n-1 \in R$.

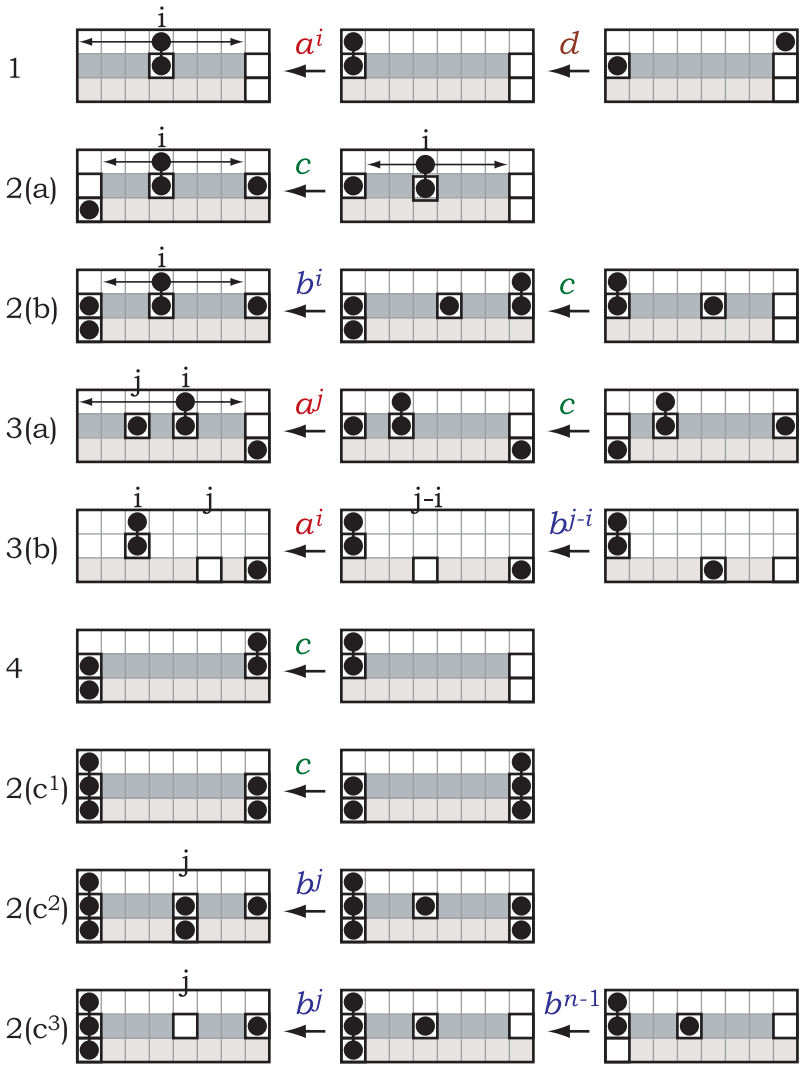

Figure 9: Reachability of (i, P, R): cases in the proof of Lemma 9.

Case 1: $i \neq n-1$, $n-1 \notin P$ and $n-1 \notin R$ (that is, the columns $n-1$ in a diagram are empty). Any such state is reachable by da^i from $(n-1, a^{-i}(P), a^{-i}(R))$, which has $n-1 \notin a^{-i}(P)$ and $n-1 \notin a^{-i}(R)$, and is therefore reachable by Lemma 8.

Case 2: $i \neq n-1$ and $n-1 \in P$ (and therefore $0 \in R$).

Case 2(a): $0 \notin P$, and therefore $i \neq 0$. Reachable from $\big(i, c^{-1}(P), c^{-1}(R \setminus \{0\})\big)$ by c, which is reachable as in Case 1.

Case 2(b): $0 \in P$ and $i \neq 0$. Consider the state $\big(0, b^{-i}(P) \setminus \{n-1\}, c^{-1}(b^{-i}(R)) \setminus \{n-1\}\big)$, which has empty column $n-1$ and is therefore reachable as in the Case 1. From this state, the automaton goes to $(n-1, b^{-i}(P), b^{-i}(R))$ by c, which has $0 \in b^{-i}(P)$ and $0 \in b^{-i}(R)$. Therefore, by b^i the automaton further proceeds to (i, P, R).

Case 2(c): $0 \in P$ and $i = 0$. This case will be proved in the end of the

proof.

Case 3: $i \neq n-1$, $n-1 \notin P$ and $n-1 \in R$.

Case 3(a): $|P| \geqslant 2$. Let $j \in P \setminus \{i\}$ and consider the state $(a^{-j}(i), a^{-j}(P), a^{-j}(R) \setminus \{n-1\})$, which is reachable as in Case 2(a). From this state, the automaton goes to $(a^{-j}(i), a^{-j}(P), a^{-j}(R))$ by cc and then to (i, P, R) by a^j.

Case 3(b): $|P| = 1$. Then this is a state is of the form $(i, \{i\}, R)$. By Condition V in the statement of the lemma, $R \neq Q$ and $R \neq Q \setminus \{i\}$. Therefore, there exists $j \notin R$ with $j \neq i$. Then $(i, \{i\}, R)$ is reachable by $b^{j-i+n-1}a^i$ from $(0, 0, b^{j-i+n-1}(a^{-i}(R)))$. The latter state has $n-1 \notin b^{j-i+n-1}(a^{-i}(R))$ and so it is reachable as in Case 1.

Case 4: $i = n-1$ (and therefore $0 \in P$ and $0 \in R$). This state is reachable by c from $(0, c^{-1}(P) \setminus \{n-1\}, c^{-1}(R) \setminus \{n-1\})$, which is in turn reachable as in Case 1.

This completes the case study. Now it remains to prove the last case 2(c), in which $i = 0$, $0 \in P$ and $n-1 \in P$ (and therefore $0 \in R$). It follows from Condition VI in the statement of the lemma that there exists $j > 0$ with $j \notin P$ or $j \in R$: indeed, if there were no such j, then $P = Q$ and $R = \{0\}$, which would contradict Condition VI. The proof splits into three subcases depending on j and its membership in P and in R:

2(c^1). $j = n-1 \in P$ (and therefore $n-1 \in R$ by the definition of j). This state is reachable by cb^{-j} from $(n-1, b^{-j}(P), b^{-j}(R))$, which is in turn reachable as in Case 4.

2(c^2). $j \in P$ (and hence $j \in R$) and $j \neq n-1$. Then it is reachable by b^{-j} from $(0, b^{-j}(P), b^{-j}(R))$, which has been proved reachable in the previous Case 2(c^1).

2(c^3). $j \notin P$. Then this state is reachable from $(0, b^{-j}(P), b^{-j}(R) \setminus \{0\})$ by b^{n-1}. The latter state is reachable as in Case 1 or 3(a).

This remaining case concludes the proof. □

Thus, by the previous three lemmas, all the states which are not proven to be unreachable by Lemma 5 are, in fact, reachable. We now prove that distinct states are inequivalent.

Lemma 10. *All states in Q_3 are pairwise inequivalent.*

Proof. Let $(i, P, R) \neq (i', P', R')$. To show the inequivalence of these states, it is sufficient to construct a string that is accepted from one of these states but not from the other.

If $R \neq R'$, then there is a state $j \in R \,\Delta\, R'$. Assume, without loss of generality, that $j \in R \setminus R'$. If $j \geqslant 1$, then the string b^{n-1-j} is accepted from (i, P, R) but not from (i', P', R'). If $j = 0$, then ab^{n-2} is accepted from (i, P, R) but not from (i', P', R').

If $P \neq P'$, then there is a state $j \in P \,\Delta\, P'$. Assume $j \in P \setminus P'$. If $j \leqslant n-2$, then $a^{n-2-j}dacab^{n-2}$ is accepted from (i, P, R) but not from (i', P', R'). If $j = n-1$, then $b^{n-2}dacab^{n-2}$ is accepted from (i, P, R) but not from (i', P', R').

Suppose $i \neq i'$. If $i \leqslant n-2$, then $a^{n-2-i}daca^{n-2}dacab^{n-2}$ is accepted from (i, P, R) but not from (i', P', R'). If $i = n-1$, then $b^{n-2}daca^{n-2}dacab^{n-2}$ is accepted from (i, P, R) but not from (i', P', R'). □

Theorem 2. *The state complexity of L^3 is at most $\frac{6n-3}{8}4^n - (n-1)2^n - n$ for all $n \geqslant 3$. This upper bound is reached on every alphabet of at least 4 letters.*

4.3 From cube to square

We now give an interesting result which states that any witness for the worst case state complexity of L^3 is also a witness for L^2 as well.

Proposition 1. *Let L be a regular language with $sc(L) = n \geqslant 3$ and $sc(L^3) = \frac{6n-3}{8}4^n - (n-1)2^n - n$. Then $sc(L^2) = n2^n - 2^{n-1}$.*

Proof. As $\mathrm{sc}(L) \geqslant 3$, we note that $L \neq \varnothing$.

Let $A = (Q, \Sigma, \delta, 0, F)$ be a DFA for L and assume without loss of generality that $Q = \{0, \ldots, n-1\}$. Then $A_2 = (Q_2, \Sigma, \delta_2, (0, \varnothing), F_2)$ where $Q_2 = Q \times 2^Q \setminus \{(i, P) \ : \ i \in F, 0 \notin P\}$, $F_2 = \{(i, P) : P \cap F \neq \varnothing\}$ and δ_2 is defined as: $\delta_2((i, P), a) = (i', P')$ where:

1. $i' = \delta(i, a)$.
2. if $i' \in F$, then $P' = \{0\} \cup \delta(P, a)$. Otherwise, $P' = \delta(P, a)$.

Assume that $\mathrm{sc}(L^2) < n2^n - 2^{n-1}$. Then when we use the construction of Yu *et al.* [16], we obtain either a state which is unreachable, or a pair of equivalent states.

Consider reachability first. Let $(i, P) \in Q_2$ be arbitrary. Consider the state $S \in Q_3$ defined by $S = (i, P, \varnothing)$ if $P \cap F = \varnothing$ and $S = (i, P, \{0, i'\})$ for some arbitrary state $i' \in Q - \{0\}$ otherwise (note that since $n \geqslant 3$, we can assume that $i' \neq 0$). The construction for L^2 of Yu *et al.* excludes those states such that $i \in F$ and $0 \notin P$, so we note that condition (a) of Lemma 5 does not hold for S. Further, by the definition of S, conditions (b)-(e) trivially hold. Condition (f) also holds since the third component of S has size zero or two by definition. Thus, S does not satisfy the conditions of Lemma 5, so must be reachable. But then (i, P) must also be reachable in A_2 by the same input.

We now turn to equivalence. In what follows, for any $(i_1, P_1), (i_2, P_2) \in Q_2$, we denote by $(i_1, P_1) \sim_2 (i_2, P_2)$ the fact that for all $x \in \Sigma^*$, if $\delta_2((i_1, P_1), x) = (i_1', P_1')$ and $\delta_2((i_2, P_2), x) = (i_2', P_2')$, then $P_1' \cap F \neq \varnothing$ if and only if $P_2' \cap F \neq \varnothing$. That is, $\sim_2$ is the equivalence of states for A_2.

We require the following claim:

Claim 4. *Let $i_1, i_2 \in Q$, $P_1, P_2 \subseteq Q$ with $(i_1, P_1) \sim_2 (i_2, P_2)$. Let $Y \subseteq Q$ be arbitrary. For all $x \in \Sigma^*$, there exists $R \subseteq Q$ such that*

$$\begin{aligned}\delta_3((i_1, P_1, Y), x) &= (i_1', P_1', R),\\ \delta_3((i_2, P_2, Y), x) &= (i_2', P_2', R).\end{aligned}$$

Proof. The proof is by induction on $|x|$. For $|x| = 0$, then $x = \varepsilon$ and we have that

$$\begin{aligned}\delta_3((i_1, P_1, Y), \varepsilon) &= (i_1, P_1, Y),\\ \delta_3((i_2, P_2, Y), \varepsilon) &= (i_2, P_2, Y).\end{aligned}$$

Assume that the result holds for all $x \in \Sigma^*$ with $|x| < k$. Let $x \in \Sigma^*$ be an arbitrary string of length k, and write $x = x'a$ where $|x'| = k - 1$ and $a \in \Sigma$. Thus, note that

$$\begin{aligned}\delta_3((i_1, P_1, Y), x') &= (i_1', P_1', R),\\ \delta_3((i_2, P_2, Y), x') &= (i_2', P_2', R)\end{aligned}$$

for some $R \subseteq Q$. Let

$$\begin{aligned}\delta_3((i_1', P_1', R), a) &= (i_1'', P_1'', R_1),\\ \delta_3((i_2', P_2', R), a) &= (i_2'', P_2'', R_2)\end{aligned}$$

for some $i_1'', i_2'' \in Q$ and $P_1'', P_2'', R_1, R_2 \subseteq Q$.

We have two cases:

(i) $P_1'' \cap F = \varnothing$. By equivalence in A_2, the same is true of P_2''. Thus, by definition of δ_3, we have that $R_1 = \delta(R, a)$ and $R_2 = \delta(R, a)$ as well. Thus, $R_1 = R_2$.

(ii) $P_1'' \cap F \neq \varnothing$. In this case, $R_1 = R_2 = \delta(R, a) \cup \{0\}$.

Thus, the claim holds. □

We now show that all pairs of reachable states in Q_2 are inequivalent. Assume not. Then there exists $(i_1, P_1), (i_2, P_2) \in Q_2$ such that $(i_1, P_1) \sim_2 (i_2, P_2)$. There are three cases:

(i) $P_1 \cap F = \varnothing$ (note that $P_2 \cap F = \varnothing$ as well by equivalence of states, in particular, with $x = \varepsilon$). In this case, as we assume that $\mathrm{sc}(L^3)$ achieves the bound in Lemma 6, and as the states $(i_1, P_1, \varnothing)$ and $(i_2, P_2, \varnothing)$ are not unreachable by Lemma 5, we must have that both $(i_1, P_1, \varnothing)$ and $(i_2, P_2, \varnothing)$ are reachable. In particular, note that conditions (d) and (e) are not satisfied since the final component is empty and $n \geqslant 3$.

Further, $(i_1, P_1, \varnothing)$ and $(i_2, P_2, \varnothing)$ are equivalent in A_3 by Claim 4: every state reachable from them on x has the same third component.

(ii) $P_1 \cap F \neq \varnothing$, but $(i_1, P_1) \neq (0, Q)$ and $(i_2, P_2) \neq (0, Q)$. In this case, the states $(i_1, P_1, \{0\})$ and $(i_2, P_2, \{0\})$ are reachable. Further, as in Case (i), they are equivalent.

(iii) $(i_1, P_1) = (0, Q)$ (a similar case handles $(i_2, P_2) = (0, Q)$). In this case, $(i_1, P_1, \{0, i\})$ and $(i_2, P_2, \{0, i\})$ are reachable states in A_3 for any choice of $0 < i \notin F$. They are equivalent by the same argument used in Case (i).

Thus, in all cases, we have constructed a pair of states in Q_3 which are reachable and equivalent. This is a contradiction, since each pair of states in Q_3 are inequivalent, by assumption. □

We note that the reverse implication in Proposition 1 does not hold: for example, the witness languages given by Rampersad for the worst case complexity of L^2 are over a two-letter alphabet. But by the calculations in Section 6, we will see that no language over a two-letter alphabet may give the worst case complexity for L^3 for small values of n.

5 Nondeterministic State Complexity

We now turn to nondeterministic state complexity. Nondeterministic state complexity for basic operations has been examined by Holzer and Kutrib [5] and Ellul [3]. We give tight bounds on the nondeterministic state complexity for L^k for any $k \geqslant 2$.

We adopt the fooling set method for proving the lower bounds on nondeterministic state complexity in the form of Birget [1, p. 188]. A *fooling set* for an NFA $M = (Q, \Sigma, \delta, q_0, F)$ is a set $S \subseteq \Sigma^* \times \Sigma^*$ such that

(a) $xy \in L(M)$ for all $(x, y) \in S$ and

(b) for all $(x_1, y_1), (x_2, y_2) \in S$ with $(x_1, y_1) \neq (x_2, y_2)$, either $x_1 y_2 \notin L(M)$ or $x_2 y_1 \notin L(M)$.

If S is a fooling set for M, then $\mathrm{nsc}(L) \geqslant |S|$.

Theorem 3. *For all regular languages L with $nsc(L) = n$ and all $k \geqslant 2$, $nsc(L^k) \leqslant kn$. Furthermore, for all $n \geqslant 2$ and $k \geqslant 2$, the bound is reached by a language over a binary alphabet.*

Proof. The upper bound is given by the construction of Holzer and Kutrib [5] or Ellul [3] for concatenation, which states that if $\mathrm{nsc}(L_1) = n$ and $\mathrm{nsc}(L_2) = m$ then $\mathrm{nsc}(L_1 L_2) \leqslant n + m$.

For the lower bound, consider the language $L_n = a^{n-1}(ba^{n-1})^*$, which is recognized by an n-state NFA given in Figure 10(a). The language $(L_n)^k = (a^{n-1}(ba^{n-1})^*)^k$ is recognized by the NFA in Figure 10(b). The following facts will be useful:

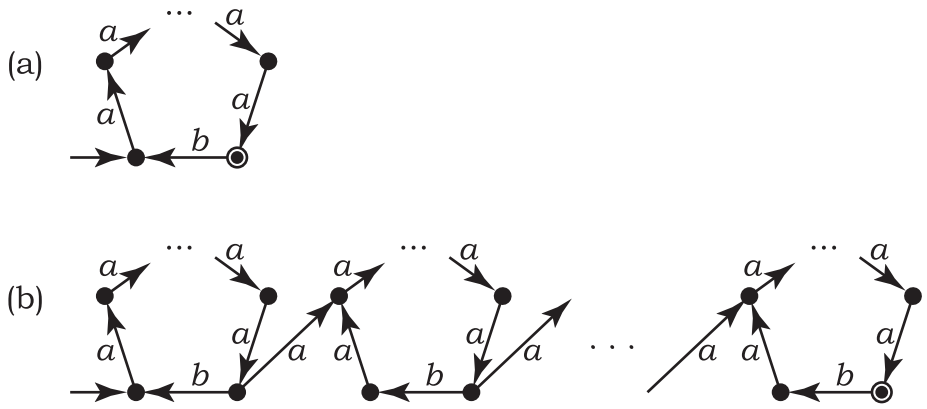

Figure 10: NFAs for L_n and for $(L_n)^k$.

Claim 5. *The only string in* $(L_n)^k \cap a^*$ *is* $a^{k(n-1)}$.

Claim 6. *The following equality holds:* $(L_n)^k \cap a^*ba^* = \{a^{j(n-1)}ba^{(k-j+1)(n-1)} : 1 \leqslant j \leqslant k\}$. *In particular, each string in the intersection has length* $(k+1)(n-1)+1$.

Our fooling set is $S_{n,k} = \{(\varepsilon, a^{n-1}ba^{k(n-1)})\} \cup S_{n,k,1} \cup S_{n,k,2}$, where

$$\begin{aligned} S_{n,k,1} &= \{(a^{(n-1)j+i}, a^{n-i-1}ba^{(n-1)(k-j)}) : 1 \leqslant i \leqslant n-1, 0 \leqslant j \leqslant k-1\} \\ S_{n,k,2} &= \{(a^{(n-1)j}b, a^{(k-j+1)(n-1)} : 2 \leqslant j \leqslant k\}. \end{aligned}$$

The total size of the fooling set is nk, as $S_{n,k,1}$ has size $k(n-1)$ and $S_{n,k,2}$ has size $k-1$. Further, by Claim 6, all of the elements $(x,y) \in S_{n,k}$ satisfy $xy \in (L_n)^k$. It remains to show that for all $(x_1,y_1),(x_2,y_2) \in S_{n,k}$ with $(x_1,y_1) \neq (x_2,y_2)$, either $x_1y_2 \notin (L_n)^k$ or $x_2y_1 \notin (L_n)^k$. We say such pairs are inequivalent in what follows.

First note that none of $a^{n-i-1}ba^{(n-1)(k-j)}$ with $1 \leqslant i \leqslant n-1$ and $0 \leqslant j \leqslant k-1$ or $a^{(k-j+1)(n-1)}$ are in $(L_n)^k$. Thus, the element $(\varepsilon, a^{n-1}ba^{k(n-1)})$ is inequivalent with all elements of $S_{n,k,1} \cup S_{n,k,2}$.

Next, we consider two pairs from $S_{n,k,1}$. Take the pairs $(a^{(n-1)j+i}, a^{n-i-1}ba^{(n-1)(k-j)})$ and $(a^{(n-1)j'+i'}, a^{n-i'-1}ba^{(n-1)(k-j')})$ for some i,i',j,j' with $1 \leqslant i,i' \leqslant n-1$ and $0 \leqslant j,j' \leqslant k-1$. Assume $(i,j) \neq (i',j')$. Consider the string $a^{(n-1)j+i}a^{n-i'-1}ba^{(n-1)(k-j')}$. Its length is $(n-1)j+i+n-i'+(n-1)(k-j') = (j-j')(n-1)+(i-i')+(n-1)(k+1)+1$. Suppose $j \neq j'$; then $|(j-j')(n-1)| \geqslant n-1$, and since $|i-i'| < n-1$, we have $(j-j')(n-1)+(i-i') \neq 0$, that is, the length of the string is different from $(n-1)(k+1)+1$. If $j = j'$ and $i \neq i'$, then $(j-j')(n-1)+(i-i') = i-i' \neq 0$, and again the string is not of length $(n-1)(k+1)+1$. In each case the string is not in $(L_n)^k$ by Claim 6.

Now consider two pairs from $S_{n,k,2}$. If we take $(a^{(n-1)j}b, a^{(k-j+1)(n-1)})$ and $(a^{(n-1)j'}b, a^{(k-j'+1)(n-1)})$, for some $2 \leqslant j < j' \leqslant k$, then we can consider the

string $w = a^{(n-1)j}ba^{(k-j'+1)(n-1)}$. Note that this string has length $(n-1)(k-(j'-j)+1)+1 < (n-1)(k+1)+1$. Therefore, w is not in $(L_n)^k$ by Claim 6.

Finally, it remains to consider pairs from $S_{n,k,1} \times S_{n,k,2}$. Consider $p_1 = (a^{(n-1)j+i}, a^{n-i-1}ba^{(n-1)(k-j)})$ and $p_2 = (a^{(n-1)j'}b, a^{(k-j'+1)(n-1)})$ for some $1 \leqslant i \leqslant n-1$, $0 \leqslant j \leqslant k-1$ and $2 \leqslant j' \leqslant k$. There are two cases:

(a) if $i \neq n-1$, then consider $a^{(n-1)j+i}a^{(k-j'+1)(n-1)}$, obtained from concatenating the first component of p_1 and the second component of p_2. As $i \neq n-1$, the length of the above string is not divisible by $n-1$ and thus is certainly not in $(L_n)^k \cap a^*$ by Claim 5.

(b) if $i = n-1$, then consider $a^{(n-1)j'}ba^{n-i-1}ba^{(n-1)(k-j)}$, which is first component of p_2 concatenated with the second component of p_1. Simplifying, we note that this string has an occurrence of bb, which is impossible as $n \geqslant 2$.

This completes the proof. □

6 Calculations

We present some numerical calculations of the worst case state complexity of L^k for k from 2 to 8 and for small values of n. In each case, this state complexity can be computed by considering automata over an n^n-letter alphabet, in which the transitions by different letters represent all possible functions from $Q \to Q$. For the accepting states, we follow the computational technique described by Domaratzki *et al.* [2], which requires only considering $O(n)$ different assignments of final states. The computed results are given in Table 1. For instance, the worst case complexity of L^5 for all DFAs of size 5 (1713946) is taken with respect to an alphabet of size 3125.

In particular, the column for L^2 starting from $n = 3$ is known from Rampersad [11], who obtained a closed-form expression $n2^n - 2^{n-1}$; note that for $n = 2$ the upper bound is five states, which is slightly less than the general bound.

n	L^2	L^3	L^4	L^5	L^6	L^7	L^8
2	5	7	9	11	13	15	17
3	20	101	410	1331	3729	8833	18176
4	56	620	6738	65854	564566		
5	144	3323	76736	1713946			
6	352	16570	782092				
7	832	79097					

Table 1: Worst case complexity of L^k.

The case of L^3 is presented in more detail in Table 2, which demonstrates the worst case state complexity of L^3 over alphabets of size 2,3,4 and of size

n^n (where n is the number of states) for automata of size n between 1 and 5. The final column gives the upper bound from Theorem 2. Note that the table demonstrates that this upper bound cannot be reached for small values of n on alphabets of size three or fewer.

	2	3	4	n^n	Upper Bound
1	1	1	1	1	
2	7	7	7	7	
3	64	96	101	101	101
4	410	608	620	620	620
5	2277			3323	3323
6				16570	16570
7				79097	79097

Table 2: Worst case state complexity of L^3.

Let us mention how these calculations helped us in obtaining the theoretical results in this paper. One of our computations considered all minimal 4-state DFAs over a 4-letter alphabet, pairwise nonisomorphic with respect to permutations of states and letters. There are 364644290 such automata; for each of them, the minimal DFA for its cube was computed, which took in total less than 6 days of machine time. In total 52 DFAs giving the top result (620 states) were found, and one of them was exactly the DFA A_4 defined in Section 4.2. We obtained the general form of the automata A_n that witness the state complexity of the cube by generalizing this single example.

7 Conclusions and Open Problems

We have continued the investigation of the state complexity of power, previously investigated by Rampersad [11]. We have given an upper bound for the state complexity of L^3 over alphabets of size two or more, and shown that it is optimal for alphabets of size four by giving a matching lower bound. By calculation, the bound is not attainable for alphabets of size two or three for all DFA sizes.

For the case of general L^k, we have established an asymptotically tight bound. In particular, we have shown that if L is a regular language with state complexity n and $k \geqslant 2$, then the state complexity of L^k is $\Theta(n2^{(k-1)n})$. The upper and lower bound on the state complexity of L^k differ by a factor of $2^{k(k-1)}\frac{n}{n-k}$; we leave it as a topic for future research to improve the bounds for $k \geqslant 4$.

We have also considered the nondeterministic state complexity of L^k for alphabets of size two or more, and have shown a tight bound of kn.

We leave open the problem of the nondeterministic state complexity of

L^k over a unary alphabet, as the nondeterministic state complexity of concatenation over a unary alphabet is not currently known exactly [5].

Acknowledgments

Research of the first author is supported in part by the Natural Sciences and Engineering Research Council of Canada. The research was conducted at the Department of Mathematics, University of Turku, during a research visit supported by the Academy of Finland grant 118540.

Research of the second author supported by the Academy of Finland under grant 118540.

References

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

[2] M. Domaratzki, D. Kisman, J. Shallit, "On the number of distinct languages accepted by finite automata with n states", *Journal of Automata, Languages and Combinatorics*, 7 (2002), 469–486.

[3] K. Ellul, *Descriptional Complexity Measures of Regular Languages*, Master's thesis, University of Waterloo (Canada), 2002.

[4] Y. Gao, K. Salomaa, S. Yu, "State complexity of star of catenation and reversal", *DCFS 2006* (Las Cruces, USA), 153–164.

[5] M. Holzer, M. Kutrib, "Nondeterministic descriptional complexity of regular languages", *International Journal of Foundations of Computer Science*, 14 (2003), 1087–1102.

[6] J. Jirásek, G. Jirásková, A. Szabari, "State complexity of concatenation and complementation", *International Journal of Foundations of Computer Science*, 16:3 (2005), 511–529.

[7] G. Jirásková, A. Okhotin, "On the state complexity of star of union and star of intersection", *Turku Centre for Computer Science Technical Report 825*, Turku, Finland, August 2007.

[8] G. Liu, C. Martín-Vide, A. Salomaa, S. Yu, "State complexity of basic operations combined with reversal", *LATA 2007* (Tarragona, Spain).

[9] A. N. Maslov, "Estimates of the number of states of finite automata", *Soviet Mathematics Doklady*, 11 (1970), 1373–1375.

[10] G. Pighizzini, J. Shallit, "Unary language operations, state complexity and Jacobsthal's function" *International Journal of Foundations of Computer Science*, 13:1 (2002) 145–159.

[11] N. Rampersad, "The state complexity of L^2 and L^k", *Information Processing Letters*, 98 (2006), 231–234.

[12] G. Rozenberg, A. Salomaa (eds.) *Handbook of Formal Languages*, Springer, 1997.

[13] A. Salomaa, K. Salomaa, S. Yu, "State complexity of combined operations", *Theoretical Computer Science*, 383:2–3 (2007), 140–152.

[14] K. Salomaa, S. Yu, "On the state complexity of combined operations and their estimation", *International Journal of Foundations of Computer Science*, 18 (2007), 683–698.

[15] S. Yu, "State complexity: recent results and open problems", *Fundamenta Informaticae*, 64 (2005), 471–480.

[16] S. Yu, Q. Zhuang, K. Salomaa, "The state complexity of some basic operations on regular languages", *Theoretical Computer Science*, 125 (1994), 315–328.

Turku
Centre *for*
Computer
Science

Lemminkäisenkatu 14 A, 20520 Turku, Finland | www.tucs.fi

University of Turku
- Department of Information Technology
- Department of Mathematical Sciences

Åbo Akademi University
- Department of Computer Science
- Institute for Advanced Management Systems Research

Turku School of Economics and Business Administration
- Institute of Information Systems Sciences

ISBN 978–952–12–1964–1
ISSN 1239-1891