On the Average State and Transition Complexity of Finite Languages ⋆

Hermann Gruber [a,1] Markus Holzer [b]

[a] *Institut für Informatik, Ludwig-Maximilians-Universität München, Oettingenstraße 67, D-80538 München, Germany*
email: gruberh@tcs.ifi.lmu.de

[b] *Institut für Informatik, Technische Universität München, Boltzmannstraße 3, D-85748 Garching bei München, Germany*
email: holzer@in.tum.de

Abstract

We investigate the average-case state and transition complexity of deterministic and nondeterministic finite automata, when choosing a finite language of a certain "size" n uniformly at random from all finite languages of that particular size. Here size means that all words of the language are either of length n, or of length at most n. It is shown that almost all deterministic finite automata accepting finite languages over a binary input alphabet have state complexity $\Theta(\frac{2^n}{n})$, while nondeterministic finite automata are shown to perform better, namely the nondeterministic state complexity is in $\Theta(\sqrt{2^n})$. Interestingly, in both cases the aforementioned bounds are asymptotically like in the worst-case. However, the nondeterministic transition complexity is shown to be again $\Theta(\frac{2^n}{n})$. The case of unary finite languages is also considered. Moreover, we develop a framework that allows us to investigate the average-case complexity of operations like, e.g., union, intersection, complementation, and reversal, on finite languages in this setup.

⋆ This paper is a completely revised and expanded version of a paper presented at the 8th Workshop on Descriptional Complexity of Formal Systems (DCFS) held in Las Cruces, New Mexico, USA, June 21–23, 2006

[1] Part of the work was done while the author was as student at Institut für Informatik, Technische Universität München, Boltzmannstraße 3, D-85748 Garching bei München, Germany.

1 Introduction

The study of descriptional complexity issues for finite automata dates back to the mid 1950's. One of the earliest results is that deterministic and nondeterministic finite automata are computationally equivalent, and that nondeterminism can offer exponential state savings compared to determinism, see [19]—by the powerset construction one increases the number of states from n to 2^n, which is known to be a tight bound. Motivated by several applications and implementations of finite automata in software engineering, programming languages and other practical areas in computer science, the descriptional complexity of finite automata problems has gained new interest during the last decade. Tight upper bounds for the deterministic and nondeterministic state complexity of many operations on regular languages are known [14,19,20].

In many applications the regular languages are actually finite as, e.g., in natural language processing or constraint satisfaction problems in artificial intelligence. This prompted quite some research activity on finite languages—see [19] for an overview. Obviously, the length of the longest word in a finite language is a lower bound on the number of states of a finite automaton accepting a finite language. In fact it can be even exponential in the length of the longest word in the finite language as shown in [3,6]. To be more precise, there is a finite language L over a binary alphabet whose longest word is of length n such that the minimal deterministic finite automaton accepting L needs $\Theta(\frac{2^n}{n})$ states. For the state savings for changing from a deterministic finite automaton to a nondeterministic finite automaton the bound for automata accepting finite languages is slightly weaker than in the general case. In [17] it was shown that one can transform every nondeterministic finite automaton accepting a finite language over a binary alphabet into an equivalent deterministic finite automaton, thereby increasing the number of states from n to $\Theta(\sqrt{2^n})$, and this bound was shown to be sharp. More results on the state complexity of operations on finite languages can be found in [4,14].

However, most of the work on descriptional complexity of regular languages yields worst-case results. To our knowledge, very few attempts have been made in order to understand certain aspects of the average behavior of regular languages [2,5,7,16]. Average-case complexity turns out to be much harder to determine than worst-case complexity, as it is currently unknown how many non-isomorphic n-state automata there are over a two letter alphabet. For a recent survey on the problem of enumerating finite automata we refer to [9]. However, for finite automata with a singleton letter input alphabet the enumeration problem was solved in [16], where also the average-case state complexity of operations on unary languages was studied. In this paper we concentrate on the average-case descriptional complexity of deterministic and non-

deterministic finite automata accepting finite languages. By choosing a finite language L of a certain size (length of the longest word) uniformly at random, one can treat the size of the minimal deterministic or nondeterministic finite automaton accepting L as a random variable. Observe that our setup is different to that used in [16]. There deterministic finite automata are chosen at random among all n-state deterministic finite *automata*, whereas our setup is centered at *languages*. Due to this difference in the model, the results cannot be directly compared to each other.

At first glance we show that almost all finite languages over a k-letter alphabet with word length at most n have state complexity $\Theta(\frac{k^n}{n})$, which is asymptotically like the worst-case. Then we introduce a stochastic process to generate finite languages, which is shown to be equivalent to the above mentioned setup choosing a finite language uniformly at random. This stochastic language generation process allows us to investigate operations on finite languages from the average-case point of view. It turns out that, for binary alphabets, the expected value of the state complexity of a deterministic finite automaton accepting the union, intersection, or complement of a finite language is larger than $c \cdot \frac{2^n}{n}$, as n tends to infinity, where c depends only on the operation and the probability of the stochastic processes generating the operands mentioned above. Moreover, also the average-case complexity of unary languages is investigated. Finally, nondeterministic finite automata are considered. There average-case bounds on deterministic and nondeterministic state complexity, as well as nondeterministic transition complexity on finite languages are obtained. It turns out that the nondeterministic state complexity is in $\Theta(\sqrt{2^n})$ on the average, which is slightly better compared to the deterministic case. However, interestingly we show that the number of transitions needed is again $\Theta(\frac{2^n}{n})$ in most cases. Hence, the overall size, i.e., the length of a description of a finite automaton, is from the average-case complexity point of view the same for both deterministic and nondeterministic finite automata.

2 Preliminaries

First we recall some definitions from formal language and automata theory; see, e.g., [19]. In particular, let Σ be an alphabet and Σ^* the set of all words, including the empty word λ, over the alphabet Σ. The length of a word w is denoted by $|w|$, where $|\lambda| = 0$. The reversal of a word w is denoted by w^R and the reversal of a language $L \subseteq \Sigma^*$ by L^R, which equals the set $\{ w^R \mid w \in L \}$. Furthermore let $\Sigma^{\leq n} = \{ w \in \Sigma^* \mid |w| \leq n \}$ and $\Sigma^n = \{ w \in \Sigma^* \mid |w| = n \}$. For any set S, we use the notation $\mathfrak{P}(S)$ to denote the powerset of S. In this paper we are interested in certain families of finite languages over a given input alphabet Σ, namely the powersets $\mathfrak{P}(\Sigma^n)$ and $\mathfrak{P}(\Sigma^{\leq n})$. In particular, in the case of a binary input alphabet, we write (1) $F_n = \mathfrak{P}(\{0,1\}^{\leq n})$ of size

$|F_n| = 2^{2^{n+1}-1}$, and (2) $B_n = \mathfrak{P}(\{0,1\}^n)$ of size $|B_n| = 2^{2^n}$.

A *nondeterministic finite automaton* is a 5-tuple $A = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is a finite set of input symbols, $\delta : Q \times \Sigma \to 2^Q$ is a transition function, $q_0 \in Q$ is an initial state, and $F \subseteq S$ is a set of accepting states. The transition function δ is extended to a function $\delta : Q \times \Sigma^* \to 2^Q$ in the natural way, i.e., $\delta(q, \lambda) = \{q\}$ and $\delta(q, aw) = \bigcup_{q' \in \delta(q,a)} \delta(q', w)$, for $q \in Q$, $a \in \Sigma$, and $w \in \Sigma^*$. A nondeterministic finite automaton $A = (Q, \Sigma, \delta, q_0, F)$ is *deterministic*, if $|\delta(q, a)| = 1$ for every $q \in Q$ and $a \in \Sigma$. In this case we simply write $\delta(q, a) = p$ instead of $\delta(q, a) = \{p\}$. The *language accepted* by a finite automaton A is $L(A) = \{\, w \in \Sigma^* \mid \delta(q_0, w) \cap F \neq \emptyset \,\}$. Two automata are equivalent if they accept the same language.

For a regular language L, the deterministic (nondeterministic, respectively) state complexity of L, denoted by $\mathrm{sc}(L)$ ($\mathrm{nsc}(L)$, respectively) is the minimal number of states needed by a deterministic (nondeterministic, respectively) finite automaton accepting L. The transition complexity is analogously defined as the state complexity and we abbreviate the deterministic (nondeterministic, respectively) transition complexity of a regular language L by $\mathrm{tc}(L)$ ($\mathrm{ntc}(L)$, respectively). To be more precise, for a nondeterministic finite automaton $A = (Q, \Sigma, \delta, q_0, F)$ the number of transitions equals $|\{\, (q, a, p) \mid p \in \delta(q, a) \,\}|$. This naturally extends to deterministic finite automata. Obviously, a deterministic finite automaton with n states and input alphabet Σ has exactly $|\Sigma| \cdot n$ transitions, because every state has exactly $|\Sigma|$ transitions leaving it. Moreover, it is easy to see that for deterministic finite automata the state minimal finite automaton is also transition minimal. Hence, in the forthcoming we will only consider the nondeterministic transition complexity of regular languages.

Moreover, we assume the reader to be familiar with the basic notations in probability theory as contained in textbooks such as [18]. In particular, we make use of Markov's inequality and Chernoff's bound.

Theorem 1 *(1) Let X be a random variable taking on nonnegative values. Then for every $t \in \mathbb{R}^+$ holds*

$$\mathbb{P}[X \geq t] \leq \frac{\mathbb{E}[X]}{t}.$$

(2) Assume X is a binomially distributed random variable. Then for $0 < d < 1$ holds

$$\mathbb{P}\left[\left|\frac{\mathbb{E}[X] - X}{\mathbb{E}[X]}\right| > d\right] < 2\exp\left(\frac{-d^2\,\mathbb{E}[X]}{3}\right).$$

3 Average Complexity of Deterministic Finite Automata

3.1 The Basic Model: Choosing a Language Uniformly at Random

A natural language family to study the descriptional complexity of finite languages is the family of languages over a fixed alphabet whose longest word has a certain length. This leads us to the language families F_n and B_n, when restricting to two-letter alphabet. These language families have recently attracted some research interest, see, e.g., [1,3,6,13]. What concerns the worst-case deterministic state complexities of the aforementioned language families the following is known: In [6] the maximum deterministic state complexity among all languages in B_n was investigated. Later, in [3] their results were in parts generalized to the language family F_n, and moreover to larger alphabet sizes. The relevant result on the finite language family under consideration reads as follows:

Theorem 2 *Let Σ be an alphabet of size k, and let $M(\Sigma^{\leq n})$ denote the maximum deterministic state complexity among all languages in $\mathfrak{P}(\Sigma^{\leq n})$. Then $M(\Sigma^{\leq n}) \leq (1 + \mathrm{o}(1))\frac{k^{n+1}}{d_k n}$, as n tends to infinity, with $d_k = \frac{k}{(k-1)^2 \log_2 k}$.*

The respective authors also gave an asymptotic lower bound for B_n, and more complex but precise formulae for $M(\Sigma^n)$ and $M(\Sigma^{\leq n})$. For our purposes these asymptotic upper bounds are sufficient. The state complexity in the best case is easily determined to be 1, which is uniquely attained by the empty language. For the worst-case, it was noted in [3] that

> "[...] several automata can reach the maximal upper bound for the state complexity. These automata are very similar, but it is very difficult to determine the languages or the number of these languages."

We show that indeed *almost every* language in $\mathfrak{P}(\Sigma^n)$ or $\mathfrak{P}(\Sigma^{\leq n})$ has deterministic state complexity in $\Theta(\frac{k^n}{n})$, and that the worst-case upper bound is also tight up to a factor of $(1 + \mathrm{o}(1))\frac{k^2}{(k-1)}$ on the average.

Theorem 3 *Let Σ be an alphabet size k, $0 < \delta < 1$, and $c_k = (k-1)\log k$. Then the number of languages acceptable by deterministic finite automata with at most $(1-\delta)\frac{k^n}{c_k n}$ states is in $\mathrm{o}(|\mathfrak{P}(\Sigma^n)|)$, and hence $\mathrm{o}(|\mathfrak{P}(\Sigma^{\leq n})|)$.*

PROOF. Let $g_k(m)$ be the function counting the number of languages over Σ acceptable by deterministic finite automata with at most m states. In [8,

Theorem 9] it was shown that $g_k(m) \leq m2^m \frac{m^{km}}{m!}$. A simple estimate yields

$$\log m! > \int_1^m \log x \, dx = m \log m - \frac{1}{\ln 2}(m-1),$$

and using $\frac{1}{\ln 2} < \frac{3}{2}$, we obtain $\log(g_k(m)) < (k-1)m \log m + \frac{5}{2}m + \log m$. Thus for every constant δ with $0 < \delta < 1$,

$$\log g_k \left((1-\delta)\frac{k^n}{c_k n} \right) < (1-\delta) \left(1 + \frac{5}{2c_k n} \right) k^n + n \log k = (1-\delta)k^n + \mathrm{o}(k^n),$$

and for n large enough, this is much smaller than $k^n = \log |\mathfrak{P}(\Sigma^n)|$, that is

$$\log g_k \left((1-\delta)\frac{k^n}{c_k n} \right) - \log |\mathfrak{P}(\Sigma^n)|$$

tends to $-\infty$. We can deduce that $\lim_{n\to\infty} g_k((1-\delta)\frac{k^n}{c_k n})/|\mathfrak{P}(\Sigma^n)| = 0$ for every such δ. □

As a corollary, we get:

Corollary 4 *Let Σ be an alphabet size k and $c_k = (k-1)\log k$. If L is a language chosen from $\mathfrak{P}(\Sigma^n)$ ($\mathfrak{P}(\Sigma^{\leq n})$, respectively) uniformly at random, then for large enough n holds*

$$\mathbb{E}[\mathrm{sc}(L)] \geq (1 - \mathrm{o}(1))\frac{k^n}{c_k n}.$$

PROOF. By Theorem 3 holds $\lim_{n\to\infty} \mathbb{P}\left[\mathrm{sc}(L) > \frac{k^n}{c_k n}\right] = 1$. The result follows by applying Markov's Inequality. □

3.2 A Different Probabilistic Model for Finite Languages

The considerations in the previous section can be seen as a model of random finite languages which are subsets of Σ^n or $\Sigma^{\leq n}$, where all languages in the respective set are equiprobable. A different model is based on a stochastic process: Given a finite set of words S, we generate a random language L by deciding for each word $w \in S$ at random whether $w \in L$ or not. This leads us to the following definition:

Definition 5 *Let Σ be a finite alphabet and S be a finite set of words over Σ. Assume $0 < p < 1$. For every $w \in S$, we define a Bernoulli experiment with two possible events $w \in L$ and $w \notin L$, such that $\mathbb{P}[w \in L] = p$ and*

$\mathbb{P}[w \notin L] = 1 - p$. *Let L denote the random event (language) obtained by carrying out this experiment independently for each word in S. Then we say that L is (S, p)-distributed.*

In fact, it is not hard to see that the equiprobable model from the previous subsection conicides with the above described Bernoulli experiment with parameter $p = \frac{1}{2}$.

Lemma 6 *Let Σ be a finite alphabet, S a finite set of words over Σ. The random language L is $(S, \frac{1}{2})$-distributed if and only if all subsets of S are equally probable.*

PROOF. Assume we pick a subset $L \subseteq S$ at random such that all subsets of S are equally probable. Note that exactly half of the subsets of S contain the word w, since there is a bijection between the subsets containing w and the subsets not containing w. Thus for every word w in S holds $\mathbb{P}[w \in L] = \frac{1}{2}$. For the other direction, assume L is $(S, \frac{1}{2})$-distributed. Then for every $L \subseteq S$ holds $\mathbb{P}[L] = (\frac{1}{2})^{|L|}(1 - \frac{1}{2})^{|S|-|L|} = \frac{1}{2^{|S|}}$. $\square$

The latter model has some conceptual advantages for the average case study of the descriptional complexity of operations on finite languages. If we randomly and independently pick two languages L_1 and L_2 in S, then for each word w in S holds: $\mathbb{P}[w \in L_1 \cap L_2] = \frac{1}{4}$. More generally spoken, we find the following result:

Lemma 7 *Let Σ be a finite alphabet, S be a finite set of words over Σ, and $0 < p_1, p_2 < 1$. If L_1 and L_2 are independent (S, p_1)-distributed and (S, p_2)-distributed languages, then $L_1 \cap L_2$ is (S, p_1p_2)-distributed, $L_1 \cup L_2$ has distribution $(S, p_1 + p_2 - p_1p_2)$, the distribution of L_1^R is (S, p_1), and that of $S \setminus L_1$ is $(S, 1 - p_1)$.* $\square$

We proceed with an easy, yet useful observation about the cardinality of L, namely that $|L|$ is a binomially distributed random variable with parameters $(2^{|S|}, p)$. The deterministic state complexity $\mathrm{sc}(L)$ is also a random variable. For (S, p) distributions, of course our main interest is devoted to the cases $S = \Sigma^n$ and $S = \Sigma^{\leq n}$. For ease of exposition, we will discuss only the case of a binary alphabet in the rest of this work, though some of the results can readily be generalized to the case of larger alphabets. So, unless stated otherwise, Σ is a binary alphabet in what follows. Next, we give an exact formula for the expected value in the case $S = \Sigma^n$.

Theorem 8 *Let L be a (Σ^n, p)-distributed language and $0 \leq p \leq 1$. Then*[2]

$$\mathbb{E}[\mathrm{sc}(L)] = 1 + \sum_{i=0}^{n} \sum_{j=1}^{2^{n-i}} \binom{2^{n-i}}{j} \left(1 - \left(1 - p^j (1-p)^{2^{n-i}-j}\right)^{2^i}\right).$$

PROOF. For $0 \leq i \leq n$, every word w of length i has a right (or residual) language $L_w = \{\, x \in \Sigma^{n-i} \mid wx \in L \,\}$ w.r.t. L. Observe that L_w is (Σ^{n-i}, p)-distributed in our model. Leave w fixed for a moment, with $|w| = i$. If we fix an arbitrary language $X \subseteq \Sigma^{n-i}$, and set $j = |X|$ then

$$\mathbb{P}[L_w = X] = p^j (1-p)^{2^{n-i}-j} \tag{1}$$

Resorting to the Myhill-Nerode theorem, we say that two words w and w' are nonequivalent, if $L_w \neq L_{w'}$. Then the number of pairwise nonequivalent words equals $\mathrm{sc}(L)$. Any two words of different length are clearly nonequivalent in our setup, (unless their right language is empty, a case of which we have to take extra care) so we discuss the expected value of the random variable Y_i denoting the number of pairwise nonequivalent prefixes in Σ^i, for $0 \leq i \leq n$, analyze the effect of possibly empty right languages, and then sum up over all i.

To each prefix w with $|w| = i$, we randomly assign a language L_w, where the probability for each choice is given by Equation 1. This can be seen in analogy to throwing 2^i balls (the prefixes) randomly into $2^{2^{n-i}}$ bins (the subsets of Σ^{n-i} as candidates for being a right language), whose probability distribution is given above. We then ask for the expected number of nonempty bins, which equals the number of distinct right languages. Clearly, the expected number of nonempty bins is the total number of bins (that is, $2^{2^{n-i}}$) minus the expected number of empty bins. The empty bins can be further partitioned according to their "size," which is the cardinality of the corresponding right language L_w. So we turn to the empty bins: The probability that a candidate $X \subseteq \Sigma^{n-i}$ with $|X| = j$ is *not* equal to L_w for *any* w of length i is

$$\mathbb{P}\left[\bigwedge_{w \in \Sigma^i} L_w \neq X\right] = \prod_{w \in \Sigma^i} \mathbb{P}[L_w \neq X] = \left(1 - p^j (1-p)^{2^{n-i}-j}\right)^{2^i},$$

as the 2^i languages L_w, for $w \in \Sigma^i$, are identically distributed and chosen independently. As there are $\binom{2^{n-i}}{j}$ subsets of Σ^{n-i}, the number of empty bins of size j can be modeled as a Bernoulli chain. Since each bin is empty with the above probability, its expectation equals $\binom{2^{n-i}}{j}\left(1 - p^j (1-p)^{2^{n-i}-j}\right)^{2^i}$. By the summation formula for the expected value, we get

[2] Here we adopt the usual convention $0^0 := 1$ (see, e.g., [12, p.162]).

$$\begin{aligned}\mathbb{E}[Y_i] &= 2^{2^{n-i}} - \sum_{j=0}^{2^{n-i}} \binom{2^{n-i}}{j} \left(1 - p^j (1-p)^{2^{n-i}-j}\right)^{2^i} \\ &= \sum_{j=0}^{2^{n-i}} \binom{2^{n-i}}{j} \left(1 - \left(1 - p^j (1-p)^{2^{n-i}-j}\right)^{2^i}\right).\end{aligned}$$

Beware that the theorem is *not* obtained by simply summing over all $\mathbb{E}[Y_i]$. Before we undertake the final summation, we have to analyze the instances of empty right languages. So we take a look of the term $j = 0$ in the above sum, in order not to double-count the dead state in the minimal deterministic finite automaton. In a first try, we simply discard this term from the sum, and do not count the dead state for each slice i. This is the expected number of non-dead states in the minimal deterministic finite automaton for L. So we under-estimated the expected value of $\mathrm{sc}(L)$. By how much? For every finite language, the minimal deterministic finite automaton *definitely* has a dead state, so we simply have to add 1. $\square$

Note that the above result generalizes to k-symbol alphabets by replacing each occurrence of 2^i with k^i and each occurrence of 2^{n-i} with k^{n-i}, respectively. In the case p is constant while n grows, we can also derive an asymptotic lower bound on the expected value of the state complexity. We write $H(p) = -p\log(p) - (1-p)\log(1-p)$ to denote the entropy of the outcome of flipping a p-biased coin.

Theorem 9 *Assume $0 < p < 1$, and $S = \Sigma^n$ or $S = \Sigma^{\leq n}$. Let L be a (S, p)-distributed language. Then*

$$\mathbb{E}[\mathrm{sc}(L)] \geq (H(p) - \mathrm{o}(1)) \frac{2^n}{n}.$$

PROOF. We will prove first that

$$\lim_{n\to\infty} \mathbb{P}\left[\mathrm{sc}(L) > c\frac{2^n}{n}\right] = 1 \tag{2}$$

for some constant c depending on p only. We explain at the end of the proof why every choice for c is valid as long as $c < H(p)$. To establish Equation 2, we begin with a basic fact about conditional probabilities:

$$\mathbb{P}\left[\mathrm{sc}(L) > c\frac{2^n}{n}\right] \geq \sum_m \left(1 - \mathbb{P}\left[\mathrm{sc}(L) \leq c\frac{2^n}{n} \,\middle|\, |L| = m\right]\right) \mathbb{P}\left[|L| = m\right], \tag{3}$$

where m runs over any subset of $\{1, 2, \ldots, |S|\}$. To estimate the probability

$$\mathbb{P}\left[\mathrm{sc}(L) \leq c\frac{2^n}{n} \,\middle|\, |L| = m\right],$$

we note first that, independent of p, all $\binom{|S|}{m}$ languages containing m words are equally probable because L is generated by a Bernoulli process. Since there are $g_2(c\frac{2^n}{n})$ languages over a binary alphabet acceptable by deterministic finite automata with at most $g_2\left(c\frac{2^n}{n}\right)$ states,

$$\mathbb{P}\left[\mathrm{sc}(L) \leq c\frac{2^n}{n} \,\middle|\, |L| = m\right] \leq \frac{g_2\left(c\frac{2^n}{n}\right)}{\binom{|S|}{m}}. \tag{4}$$

We now investigate the region where m is close to $\mathbb{E}[|L|] = p|S|$, namely $(1-d)\mathbb{E}[|L|] \leq m \leq (1+d)\mathbb{E}[|L|]$ for some d. To this end, we choose a small constant $d = d_c$ depending only on c (to be fixed later). For now, we require only $0 < d < 1$ and $(1+d)p < 1$. Next, we derive a lower bound for the binomial coefficients $\binom{|S|}{m}$ occurring in Inequality 4 in the case $(1-d)p|S| \leq m \leq (1+d)p|S|$. Set $\alpha = (1-d)p$ and $\beta = (1+d)p$. We assume that $p \geq \frac{1}{2}$, that is, $\alpha|S|$ is at least as close to $\frac{|S|}{2}$ as $\beta|S|$. For the other case we replace α with β in all of the following computations. Then $\binom{|S|}{m} \geq \binom{|S|}{\alpha|S|}$ for every m under consideration. Asymptotic estimates for this binomial coefficient are known, e.g., from Stirling's formula one obtains:

$$\lim_{n\to\infty} \log \binom{|S|}{\alpha|S|} - \left[H(\alpha)|S| - \frac{1}{2}\log(2\pi\alpha(1-\alpha)|S|)\right] = 0. \tag{5}$$

Recall $\log g_2(c2^n/n) < c(1+\frac{5}{2n})2^n + n$ from the proof of Theorem 3; and thus

$$\begin{aligned}\lim_{n\to\infty} \log g_2(c2^n/n) - \log \binom{|S|}{\alpha|S|} &< \lim_{n\to\infty} c(1+\frac{5}{2n})2^n + n \\ &\quad - \left[H(\alpha)2^n - \frac{1}{2}(n+1) - \frac{1}{2}\log(2\pi\alpha(1-\alpha))\right] \\ &= \lim_{n\to\infty} \left(c - H(\alpha)\right)2^n.\end{aligned}$$

The last line is obtained by pulling out the factor 2^n of all terms and then removing the o(1) inner terms. This limit tends to $-\infty$ as long as $c < H(\alpha)$. We conclude that the probability in Inequality 4 tends to zero as n grows. As $\binom{|S|}{m} \geq \binom{|S|}{\alpha|S|}$ for $\alpha|S| \leq m \leq \beta|S|$, a similar fact holds for all m under consideration. Thus for any constant $\delta > 0$ holds

$$\mathbb{P}\left[\mathrm{sc}(L) \leq c\frac{2^n}{n} \,\middle|\, |L| = m\right] < \delta,$$

provided n is large enough. We plug this into Inequality 3 to obtain for every constant $\delta > 0$:

$$\lim_{n\to\infty} \mathbb{P}\left[\mathrm{sc}(L) > c\frac{2^n}{n}\right] > \lim_{n\to\infty} \sum_m (1-\delta)\mathbb{P}\left[|L| = m\right], \tag{6}$$

where the index m ranges from $(1-d)\mathbb{E}[|L|]$ to $(1+d)\mathbb{E}[|L|]$. We show next that the sum $\sum_m \mathbb{P}[|L| = m]$ converges to 1 in the limit. The random variable $|L|$ is binomially distributed; so using Chernoff's bound, we have

$$\sum_m \mathbb{P}[|L| = m] = \mathbb{P}\left[\left|\frac{\mathbb{E}[|L|] - |L|}{\mathbb{E}[|L|]}\right| \leq d\right] \geq 1 - 2\exp\left(\frac{pd^2}{3}\right)^{-|S|}.$$

Since $\frac{pd^2}{3}$ is a positive constant, $\exp(\frac{pd^2}{3}) > 1$, this probability tends to 1 with $n \to \infty$. We may now plug this into Inequality 6 to find that for every $\delta > 0$ holds $\lim_{n\to\infty} \mathbb{P}\left[\mathrm{sc}(L) > c\frac{2^n}{n}\right] > 1 - \delta$, and thus the probability in Equation 2 indeed converges to 1.

Finally, we have to argue that c can be chosen freely as long as $0 < c < H(p)$. Assume still $p \geq \frac{1}{2}$ for the moment. The function $H(x)$ is a strictly increasing function for $x \in (0; \frac{1}{2}]$, with $\lim_{x\to 0^+} H(x) = 0$ and $H(\frac{1}{2}) = 1$. Thus for every $y \in (0; 1]$, there is a unique preimage $x = H^{-1}(y)$ with $x \in (0; \frac{1}{2}]$, and under this restriction, we may speak of H^{-1} as a function $H^{-1} : (0; 1] \mapsto (0; \frac{1}{2}]$. Recall that we have to choose the constant $d = d_c$ such that $0 < d < 1$, $(1 + d)p < 1$, and $c < H(\alpha) = H((1 + d)p)$, in other words $0 < d < 1 - p^{-1}H^{-1}(c)$. Such a d can be found as long as $0 < c < H(p)$. For the case $p < \frac{1}{2}$, note that $H(\beta) = H(1 - \beta)$. We choose the constant d_c such that $c < H(1 - \beta)$, that is $0 < d < p^{-1}(1 - H^{-1}(c)) - 1$. If $c < H(p)$, then $H^{-1}(c) < p < \frac{1}{2}$, and the numerator in the above fraction is greater than the denominator. So we can find a suitable d also in this case. The theorem now follows by applying Markov's Inequality on Equation 2: For all $c < H(p)$ holds

$$\lim_{n\to\infty} \frac{\mathbb{E}(\mathrm{sc}(L))}{c2^n/n} \geq 1,$$

and so $\mathbb{E}(\mathrm{sc}(L)) \geq (H(p) - \mathrm{o}(1))\frac{2^n}{n}$. □

The cases of particular interest are the cases $p = \frac{1}{4}$ and $p = \frac{3}{4}$, since these occur for the state complexities of the results for the union and intersection operations on random finite languages in our setup, see Lemma 7. For $H(\frac{1}{2}) = 1$ and $H(\frac{1}{4}) = H(\frac{3}{4}) > \frac{4}{5}$, the lower bound for the expected value almost matches the a priori upper bound given in Theorem 2.

It is worth mentioning that a corresponding result for larger alphabets can be proved along the lines of the above proof, namely that for $|\Sigma| = k$ holds $\mathbb{E}(\mathrm{sc}(L)) \geq \left(\frac{H(p)}{(k-1)\log k} - \mathrm{o}(1)\right)\frac{k^n}{n}$: Most of this proof works as detailed above; the main difference is that we have to use an inequality similar to Inequality 4, but this time with the term $g_k\left(c\frac{k^n}{(k-1)\log kn}\right)$ on the right-hand side. Then one uses the upper bound on this term derived in the proof of Theorem 3, together with the estimates $n \log k \leq |S| \leq (n + 1)\log k - \log(k - 1)$, to prove that the probability in the mentioned inequality tends to zero.

We turn to the case where $\Sigma = \{0\}$ is a unary alphabet. The case where all words are of equal length is arguably not very interesting, so we consider the subsets of $\{0\}^{\leq n}$ next.

Lemma 10 *Let L be a $(\{0\}^{\leq n}, p)$-distributed language with $0 < p < 1$. Then*

$$\mathbb{E}(\mathrm{sc}(L)) = n + 2 - p^{-1}(1-p) + p^{-1}(1-p)^{n+2}.$$

PROOF. The state complexity is governed by the longest word in the language. We have $\mathrm{sc}(L) = 1$ if and only if $L = \emptyset$, and the probability of this event equals $(1-p)^{n+1}$; otherwise $\mathrm{sc}(L) = k$ if and only if $k-2$ is the length of the longest word in L. The probability of the event "length of the longest word in L equals $k-2$" conditional on the event "$L \neq \emptyset$" equals $p \cdot (1-p)^{n-k+2}$. An easy observation is

$$\mathbb{P}\left[\text{longest word in } L \text{ has length } k-2 \mid L \neq \emptyset\right] = p \cdot (1-p)^{n-k+2}.$$

And for $k > 1$, we have $\mathbb{P}[\mathrm{sc}(L) = k] = \mathbb{P}[\mathrm{sc}(L) = k \mid L \neq \emptyset]$. Using the geometric series formulae $\sum_{k=0}^{n} q^k = p^{-1}(1-q^{n+1})$ and the identity $\sum_{k=0}^{n} kq^k = -(n+1)p^{-1}q^{n+1} - p^{-2}q^{n+2} + p^{-2}q$, and setting $q = 1-p$, the expected value computes as

$$\begin{aligned}\mathbb{E}(\mathrm{sc}(L)) &= q^{n+1} + \sum_{k=2}^{n+2} kpq^{n-k+2} \\ &= q^{n+1} + p(n+2)\sum_{k=0}^{n} q^k - p\sum_{k=0}^{n} kq^k = n + 2 - p^{-1}q + p^{-1}q^{n+2}.\end{aligned}$$

This proves the stated claim. □

Using Lemma 7, we obtain for the union of two $(\{0\}^{\leq n}, \frac{1}{2})$-distributed languages over an unary alphabet an expected value very close to $n + \frac{5}{3}$, if n is large; for the intersection it is close to $n-1$, and for reversal and bounded complement, that is, complement with respect to the set $\{0\}^{\leq n}$, it is the same as the operand, i.e., close to $n+1$.

4 Average Complexity of Nondeterministic Finite Automata

Now let us turn our attention to the nondeterministic state and transition complexity of finite languages. For the unary case, observe that for all nonempty finite languages, the nondeterministic state complexity is almost equal to the deterministic one, except that we can remove the dead state, and for the empty language it equals 1. Elementary computations with conditional expectations then give, in the terminology of Lemma 10,

$$\mathbb{E}(\mathrm{nsc}(L)) = \mathbb{E}(\mathrm{sc}(L)) - 1 + (1-p)^{n+1}.$$

For the binary case, a result in the same spirit as Theorem 3 but now concerning the size of nondeterministic finite automata was obtained in [13].

Lemma 11 *(1) The number of languages over Σ acceptable by nondeterministic finite automata with at most $\frac{1}{2}\sqrt{2^n}$ states is bounded above by $\sqrt{2^{n+2^n}} = \mathrm{o}(|B_n|) = \mathrm{o}(|F_n|)$.*

(2) The number of languages over Σ acceptable by nondeterministic finite automata with at most $\frac{2^n}{20n}$ transitions is bounded above by $\sqrt{2^{2^n}} = \mathrm{o}(|B_n|) = \mathrm{o}(|F_n|)$.

The descriptional complexity in the nondeterministic model cannot exceed the corresponding one in the deterministic model. And in the latter model, transition complexity is linear in state complexity. Thus, we have a preliminary worst-case estimate of $\mathrm{O}(\frac{2^n}{n})$ for both nondeterministic state and transition complexity. By Lemma 11, this is essentially optimal for the number of transitions, but it can be improved for the number of states:

Lemma 12 *Assume $L \subseteq \Sigma^{\leq n}$. Then* $\mathrm{nsc}(L) < \frac{3}{\sqrt{2}}\sqrt{2^n}$.

PROOF. Let $\ell = \lfloor (n-1)/2 \rfloor$ and $m = \lceil (n-1)/2 \rceil$. We construct a nondeterministic finite automaton $A = (Q, \{0,1\}, \delta, p_\lambda, F)$, where $Q = P_1 \cup P_2$ (the union is disjoint) with $P_1 = \{\, p_w \mid w \in \{0,1\}^* \text{ and } |w| \leq \ell \,\}$ and $P_2 = \{\, q_w \mid w \in \{0,1\}^* \text{ and } |w| \leq m \,\}$, the set $F = \{q_\lambda\} \cup \{\, p_\lambda \mid \lambda \in L \,\}$, and the transition function is specified as follows:

(1) For all $p_w \in P_1$ and $a \in \{0,1\}$, the set $\delta(p_w, a)$ contains the element p_{wa}.
(2) For all $w \in L \setminus \{\lambda\}$, if $w = xay$ is the unique decomposition, where $|x| = \lfloor (|w|-1)/2 \rfloor$, a is a single letter, and $|y| = \lceil (|w|-1)/2 \rceil$, then let $\delta(p_x, a)$ contain the element q_y.
(3) For all $q_w \in P_2 \setminus \{q_\lambda\}$ and $a \in \{0,1\}$, the set $\delta(p_{aw}, a)$ contains the element q_w.

This completes the construction of the nondeterministic finite automaton. It is easy to see that for the number of states in A, we have

$$|P_1| + |P_2| = 2^{\ell+1} - 1 + 2^{m+1} - 1 < \frac{3}{\sqrt{2}}\sqrt{2^n}.$$

It remains to show that $L(A) = L$. Note that every state p_w in P_1 is only reachable by the word w from the initial state p_λ, and that for every state q_w in P_2 there is only one path leading to the final state q_λ. So every transition leading from P_1 to P_2 leads to the acceptance of exactly one word in L. This proves the stated claim. □

Lemma 11 tells us that the above construction for finding a compact nondeterministic finite automaton works pretty well on the average, if we wish to keep the number of states as small as possible. Also, this construction is almost optimal in the worst case, as witnessed by the language family A_k in [11, Example 3]. The drawback in this construction is that the number of transitions is at least equal to the cardinality of the accepted language. Now we have determined the growth order of the average descriptional complexity in the families F_n and B_n for three descriptional measures: deterministic state complexity, and nondeterministic state and transition complexity.

Theorem 13 *Let L be a $(S, \frac{1}{2})$-distributed language with $S = \Sigma^{\leq n}$ or $S = \Sigma^n$. Then for every $\delta > 0$, language L has* all *of the following properties with probability at least $1 - \delta$, provided n is large enough:*

$$\frac{1}{2} \cdot \sqrt{2^n} < \mathrm{nsc}(L) < \frac{3}{\sqrt{2}} \cdot \sqrt{2^n},$$

$$\frac{1}{20} \cdot \frac{2^n}{n} < \mathrm{ntc}(L) < \frac{2^{n+4}}{n},$$

and

$$\frac{2^{n-1}}{n} < \mathrm{sc}(L) < \frac{2^{n+3}}{n}.$$

□

As an application of the probabilistic method used here, we present a worst-case comparison of nondeterministic state complexity versus nondeterministic transition complexity. In [1], a heuristics for reducing the number of states of nondeterministic finite automata accepting languages in B_n is proposed. It was observed that, although the heuristics performed well in reducing the number of states in the given automata, it occasionally blew up the number of transitions:

> "It seems that the number of states is always used to measure the size of automata. [Our] experimentations show that it would be better to also take

into account the number of transitions [...]. This is clearly important from a practical point of view, but perhaps also from a theoretical one [...]."

We substantiate this empirical study by proving that there can be a superlinear lower bound on nondeterministic transition complexity when expressed as a function of nondeterministic state complexity. And in fact *many* languages that can be accepted by nondeterministic finite automata with a given number of states exhibit this behavior.

We can extend the model of nondeterministic finite automata by allowing ε-transitions. In the latter model, the nondeterministic transition complexity will be denoted $\mathrm{ntc}_\varepsilon(L)$. By definition, $\mathrm{ntc}_\varepsilon(L) \leq \mathrm{ntc}(L)$, but there is an infinite family of languages K_n such that $\mathrm{ntc}_\varepsilon(K_n) \in \mathrm{O}(n)$, while $\mathrm{ntc}(K_n) = \Omega(n(\log n)^k)$, for all $k > 0$, holds, see [15]. To prepare the next result, we derive a counting argument similar to Lemma 11 first—which gives at the same time an improved lower bound:

Lemma 14 *For $n \geq 8$, the number of languages over Σ that can be accepted by nondeterministic finite automata with ε-transitions having at most $\frac{2^n}{4n}$ transitions is bounded above by $\sqrt{|B_n|} = \mathrm{o}(|B_n|) = \mathrm{o}(|F_n|)$.*

PROOF. For the proof it will be more convenient to bound the number of languages acceptable by nondeterministic finite automata with ε-transitions having at most $\frac{2^n}{4n}$ "edges" instead—by an edge, we mean an edge in the underlying simple directed graph of the automaton. As an edge can be labeled with more than one alphabet symbol, there are always at least as many transitions in the automaton as edges in the underlying graph.

Combining the arguments in [8,13], there are at most $7\binom{s^2}{t}(2s-1)+1$ languages over a binary alphabet that can be accepted by nondeterministic finite automata with ε-transitions with exactly s states and exactly t edges: there are $\binom{s^2}{t}$ ways to place t edges between pairs of states, and every such edge may be labeled with one of the 7 nonempty subsets of $\{\varepsilon, a, b\}$. Either the initial state q_0 is accepting or not, and we can assume that the other accepting states are labeled $q_1, q_2, \ldots, q_k$ with $0 \leq k \leq s-1$. If no final state is selected, only one language can be accepted, namely the empty language.

If we bound only the number of edges from above, observe that the number of states needed can exceed the number of edges needed by at most 1. Overmore, if a language can be accepted by a nondeterministic finite automaton with at most t edges, then it can also be accepted by an automaton with exactly t edges and exactly $t+1$ states: In case exactly t edges are needed in order to accept the language, we can just add as many additional useless states as needed to the automaton without changing the accepted language. Otherwise,

the language can be accepted by an automaton with exactly $t' < t$ edges and $t'+1$ states. We then add as many useless (nonaccepting) states as needed, and for each such state we extend the transition function by adding an edge leading from the start state to the newly added useless state, in order to get a total number of t edges and $t+1$ states without altering the accepted language. Thus we obtain an upper bound of $7(2t+1)\binom{(t+1)^2}{t}+1$ on the number of these languages. Using $\binom{m}{k} < m^k/k!$ and $\log k! > k\log k - \frac{3}{2}k$, we find that

$$\log\binom{(t+1)^2}{t} < 2t\log(t+1) - t\log t + \frac{3}{2}t < 2t\log t,$$

for $t \geq 8$, and the number of languages under consideration is at most $7(2t+1)t^{2t}+1$. Setting $t = 2^{n-2}/n$ with $n \geq 8$, we find that this number is smaller than

$$\frac{7(2^{n-1}/n+1)}{(4n)^{2^{n-1}/n}}2^{2^{n-1}} + 1 < 2^{2^{n-1}}.$$

This proves the stated claim. □

Now we are ready for the last theorem.

Theorem 15 *For every $k \geq 34$, there is a set T of finite languages over Σ such that for every $L \in T$ holds*

$$\mathrm{nsc}(L) < k, \quad \textit{but} \quad \mathrm{ntc}_\varepsilon(L) > \frac{k^2}{c\cdot\log k},$$

for some constant $c \leq 72$. Moreover the size of T is of order $2^{\Omega(k^2)}$.

PROOF. Let n be the unique integer such that $\frac{3}{\sqrt{2}}\sqrt{2^n} < k \leq 3\cdot\sqrt{2^n}$. Then by our choice of n holds $\log k > \frac{1}{2}n + \log\frac{3}{\sqrt{2}} > \frac{1}{2}n$ and $k^2 \leq 9\cdot 2^n$.

By Lemma 14, there are more than $|B_n| - \sqrt{(|B_n|)}$ languages in B_n that cannot be accepted by nondeterministic finite automata with ε-transitions having at most $\frac{2^n}{4n}$ transitions, provided $n \geq 8$. The lemma is applicable for $k \geq 34$, since $\frac{3}{\sqrt{2}}\sqrt{2^8} < 34$. These languages form the set T. Furthermore,

$$|T| > 2^{2^n-1} \geq 2^{k^2/9-1} = 2^{\Omega(k^2)},$$

for $k \geq 34$. On the other hand, for every $L \in T$ holds $\mathrm{nsc}(L) < \frac{3}{\sqrt{2}}\sqrt{2^n} < k$ by Lemma 12. But any nondeterministic finite automaton accepting a language $L \in T$ has more than $\frac{2^n}{4n}$ transitions, even if ε-transitions are allowed, and $\frac{k^2}{\log k} < \frac{9\cdot 2^n}{1/2n} = 72\cdot\frac{2^n}{4n}$, which completes the proof. □

In [10], it is reported that a simlar result for ε-free nondeterministic finite automata was found independently by J. Kari. We also note that a lower bound for the gap between nondeterministic state and transition complexity was obtained in [10] by more constructive means. There an explicitly defined family of languages is given where $\mathrm{nsc}(L_n) = \Theta(n)$, but $\mathrm{ntc}(L_n) = \Theta(n^{\frac{3}{2}})$.

5 Discussion

We investigated the average descriptional complexity of finite automata for two natural families of finite languages over a unary, binary and k-letter alphabet: In the first family, all words have the same length, and in the second family, words of length up to a given bound are allowed. These language families were already subject to worst-case analysis of the deterministic model in [3,6], and lower bounds on the average for the nondeterministic model were obtained in [13].

We tried to complete the picture by providing an average-case analysis with asymptotically tight results, which are in all cases close to the worst-case upper bounds. Namely, the average deterministic state complexity in both families is $\Theta(\frac{k^n}{n})$, for a fixed k-letter alphabet, and $\Theta(n)$ for unary alphabet, where n is the maximal allowed word length. We introduced a stochastic process allowing us to investigate the average effect on state complexity of various language operations, too. We found that the average state complexity cannot essentially increase compared to that of the operands, and also that it cannot decrease by more than a constant factor, the size of the constant depending only on the operation. In the case of unary finite languages, we found that the average state complexity of the result of an operation is for some operations indeed smaller than that of the operands. So there is a notable difference to worst-case results: There the outcome of union and intersection can have complexity quadratic in the size of each operand; and the reversal operation can even cause an exponential blow-up in the number of states. Then we turned to the nondeterministic model. The nondeterministic state complexity is in $\Theta(\sqrt{2^n})$ on the average over a binary alphabet, suggesting superiority over the deterministic model; however the number of transitions needed is again $\Theta(\frac{2^n}{n})$ in almost all cases; and this still holds in the case where ε-transitions are allowed. One can deduce that there are many languages for which the gap between nondeterministic state and transition complexities can be almost quadratic.

Acknowledgements Thanks to Felix Fischer for some useful discussion and to the anonymous referees for valuable suggestions and corrections.

References

[1] J. Amilhastre, P. Janssen, and M.-C. Vilarem. FA minimisation heuristics for a class of finite languages. In O. Boldt and H. Jürgensen, editors, *Proceedings of the 4th International Workshop on Implementation of Automata*, number 2214 in LNCS, pages 1–12, Potsdam, Germany, July 1999. Springer.

[2] F. Bassino and C. Nicaud. Enumeration and random generation of accessible automata. Enumeration and random generation of accessible automata. *Theoretical Computer Science*, to appear.

[3] C. Câmpeanu and W. H. Ho. The maximum state complexity for finite languages. *Journal of Automata, Languages and Combinatorics*, 9(2–3):189–202, September 2004.

[4] C. Câmpeanu, K. Čulik II, K. Salomaa, and S. Yu. State complexity of basic operations on finite languages. In O. Boldt and H. Jürgensen, editors, *Proceedings of the 4th International Workshop on Implementing Automata*, number 2214 in LNCS, pages 60–70, Potsdam, Germany, July 1999. Springer.

[5] J.-M. Champarnaud and T. Paranthoën. Random generation of DFAs. *Theoretical Computer Science*, 330(2):221–235, 2005.

[6] J.-M. Champarnaud and J.-E. Pin. A maxmin problem on finite automata. *Discrete Applied Mathematics*, 23:91–96, 1989.

[7] M. Domaratzki. State complexity of proportional removals. *Journal of Automata, Languages and Combinatorics*, 7(4):455–468, 2002.

[8] M. Domaratzki, D. Kisman, and J. Shallit. On the number of distinct languages accepted by finite automata with n states. *Journal of Automata, Languages and Combinatorics*, 7(4):469–486, 2002.

[9] M. Domaratzki. Enumeration of formal languages. *Bulletin of the EATCS*, 89:117-133, 2006.

[10] M. Domaratzki and K. Salomaa. Lower bounds for the transition complexity of NFAs. In R. Královič and P. Urzycyn, editors, *Proceedings of the 31st Conference on Mathematical Foundations of Computer Science*, number 4162 in LNCS, pages 315–326, Stará Lesná, Slovakia, August–September 2006. Springer.

[11] I. Glaister and J. Shallit. A lower bound technique for the size of nondeterministic finite automata. *Information Processing Letters*, 59:75–77, 1996.

[12] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, 1988.

[13] G. Gramlich and G. Schnitger. Minimizing NFA's and regular expressions. In V. Diekert and B. Durand, editors, *Proceedings of the 22nd Annual Symposium*

on Theoretical Aspects of Computer Science, number 3404 in LNCS, pages 399–411, Stuttgart, Germany, February 2005. Springer.

[14] M. Holzer and M. Kutrib. State complexity of basic operations on nondeterministic finite automata. In J.-M. Champarnaud and D. Maurel, editors, *Proceedings of the 7th International Conference Implementation and Application of Automata*, number 2608 in LNCS, pages 148–157, Tours, France, July 2003. Springer.

[15] Juraj Hromkovič and Georg Schnitger. NFAs with and without ε-transitions. In L. Caires, G. G. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Proceedings of the 32nd International Colloquium Automata, Languages and Programming*, number 3580 in LNCS, pages 385–396, Lisbon, Portugal, July 2005. Springer.

[16] C. Nicaud. Average state complexity of operations on unary automata. In M. Kutylowski, L. Pacholski, and T. Wierzbicki, editors, *Proceedings of the 24th Conference on Mathematical Foundations of Computer Science*, number 1672 in LNCS, pages 231–240, Szklarska Poreba, Poland, September 1999. Springer.

[17] K. Salomaa and S. Yu. NFA to DFA transformation for finite language over arbitrary alphabets. *Journal of Automata, Languages and Combinatorics*, 2(3):177–186, 1997.

[18] T. Schickinger and A. Steger. *Diskrete Strukturen II (in German)*. Springer, 2001.

[19] S. Yu. Regular languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 1, pages 41–110. Springer, 1997.

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