Results on the Average State and Transition Complexity of Finite Automata Accepting Finite Languages (Extended Abstract)

Hermann Gruber and Markus Holzer
Institut für Informatik, Technische Universität München,
Boltzmannstraße 3, D-85748 Garching bei München, Germany
{gruberh,holzer}@informatik.tu-muenchen.de

Abstract

We investigate the average-case state and transition complexity of deterministic and nondeterministic finite automata, when choosing a finite language of given maximum word length n uniformly at random. The case where all words are of equal length is also taken into account. It is shown that almost all deterministic finite automata accepting finite languages over a binary input alphabet have state complexity $\Theta(\frac{2^n}{n})$. Moreover, we develop a framework that allows us to investigate the average-case complexity of operations like union, intersection, complementation, and reversal on finite languages. Nondeterministic finite automata are shown to perform better than deterministic ones, namely their state complexity is in $\Theta(\sqrt{2^n})$ on the average. 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.

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 nondeterministic finite automata can offer exponential state savings compared to deterministic ones, see [18]—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 [12, 18, 19].

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 [18] 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, 7]. 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 bounds for automata accepting finite languages is slightly weaker than in the general case. In [16] it was shown that one can transform every nondeterministic finite automaton accepting a finite language into an equivalent deterministic finite automaton 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, 12].

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 behaviour of regular languages [2, 5, 6, 8, 14]. Average-case complexity turns out to be much harder to determine than worst-case complexity, as it is currently unknown how many non-isomorphic automata of n states there are over a binary 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 [14], 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 nondeterministic finite automata accepting finite languages. By choosing a finite language L with given maximal word length 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 [14]. 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 on finite languages cannot be directly compared to each other.

At first glance we show that almost all deterministic finite automata accepting finite languages over a binary input alphabet with word length at most n have state complexity $\Theta(\frac{2^n}{n})$, 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 the expected value of the state complexity of a deterministic finite automaton accepting the union or intersection of two finite languages is larger than $\frac{4}{5} \cdot \frac{2^n}{n}$ on the average, as n tends to infinity. Similar bounds can be derived for the iteration of Boolean operations. Moreover, the average-case complexity of operations on finite unary languages is determined exactly. Finally, nondeterministic finite automata are considered. It turns out that the nondeterministic state complexity is in $\Theta(\sqrt{2^n})$ on the average, which is superior to the deterministic case with respect to the number of *states*. 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 state machine, is from the average-case complexity point of view the same for both deterministic and nondeterministic finite automata. Finally, we note that results similar to those for the binary case can be derived for larger alphabet sizes along the way outlined here.

2 Preliminaries

First we recall some definitions from formal language and automata theory; see, e.g., [18]. In particular, let Σ be an alphabet and Σ^* the set of all words over the alphabet Σ containing the empty word λ. 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 $\Sigma^n = \{ w \in \Sigma^* \mid |w| = n \}$. In this paper we are interested in the following two language families over a binary input alphabet Σ: (1) $F_n = 2^{\Sigma^{\leq n}}$ of size $|F_n| = 2^{2^{n+1}-1}$, and (2) $B_n = 2^{\Sigma^n}$ 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 the transition function, $q_0 \in Q$ is the initial state, and $F \subseteq S$ is the set of accepting states. The transition function δ is extended to a function from $\delta : Q \times \Sigma^* \to 2^Q$ 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. Observe that only non-blocking transitions are counted; a transition is *blocking*, if $\delta(q, a) = \emptyset$, for some $q \in Q$ and $a \in \Sigma$. Obviously, a deterministic finite automaton with n states and input alphabet Σ has exactly $|\Sigma| \cdot n$ transitions, because every state has $|\Sigma|$ transitions leaving it. And it is easy to see that in the deterministic model 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 of probability theory as contained in textbooks such as [17]. In particular, we make use of Markov's inequality and Chernoff's bound.

Theorem 1 *1. Let X be a random variable taking on non-negative 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 variable. Then for every $0 < d < 1$ holds $\mathbb{P}\left[\left|\frac{\mathbb{E}[X]-X}{\mathbb{E}[X]}\right| > d\right] < 2\exp\left(\frac{-d^2\,\mathbb{E}[X]}{3}\right)$.

Finally, we introduce the following notion in order to compare functions: Let $f, g : \mathbb{N} \to \mathbb{R}^+$ be two monotone functions. Then $f(n) \sim g(n)$, if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$, and $f(n) \ll g(n)$, if $\lim_{n\to\infty} \frac{f(n)}{g(n)} < 1$.

Lemma 2 *Assume $f, g : \mathbb{N} \to \mathbb{R}^+$ be two monotone functions. Then the inequality $\lim_{n\to\infty}(\log f(n) - \log g(n)) < \infty$ implies $f(n) = O(g(n))$, where* log *denotes the logarithm of base* 2.

In particular, this lemma shows that $\log(f(n)) \ll \log(g(n))$ implies $f(n) = o(g(n))$.

3 Finite Languages with Bounded Word Length

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 interests, see, e.g., [1, 3, 7, 11]. Concerning the worst-case deterministic state complexities of the aforementioned language families the following is known: In [7] the maximum deterministic state complexity among all languages in B_n was investigated. Later, in [3] this result was generalized to the language family F_n, and moreover to larger alphabet sizes. The results on the language families under consideration read as follows:

Theorem 3 ([3, 7]) *1. Let $M(B_n)$ denote the maximum deterministic state complexity among all languages in B_n. Then $M(B_n) \leq (1 + o(1))2^{n+1}/n$.*

2. Let $M(F_n)$ denote the maximum deterministic state complexity among all languages in F_n. Then $M(F_n) \leq (1 + o(1))2^{n+2}/n$.

The respective authors also gave more complex but precise formulas for $M(B_n)$ and $M(F_n)$, but for our purposes knowledge about the asymptotic behavior is sufficient; also note that $M(B_n)$ is a lower bound for $M(F_n)$. So in both cases the asymptotics are in $\Theta(\frac{2^n}{n})$. We show that indeed *almost every* language in F_n or B_n has deterministic state complexity in $\Theta(\frac{2^n}{n})$. The theorem reads as follows:

Theorem 4 *Let $c < 1$ be a constant. If L is a language chosen from B_n (F_n, respectively) uniformly at random, then for large enough n holds $\mathbb{E}[\mathrm{sc}(L)] > c\frac{2^n}{n}$.*

Proof We show that the number of languages in B_n acceptable by deterministic finite automata with at most $c\frac{2^n}{n}$ states is in $o(|B_n|)$, and hence $o(|F_n|)$. Stated another way, $\lim_{n\to\infty} \mathbb{P}\left[\mathrm{sc}(L) > c\frac{2^n}{n}\right] = 1$. The result then follows immediately if we apply Markov's inequality. Let $g(n)$ be the function counting the number of languages acceptable by deterministic finite automata with at most n states. In [9, Theorem 9] it was shown that $g(n) \leq n2^n \frac{n^{2n}}{n!}$. Thus, $\log(g(n)) < n\log n + \frac{5}{2}n + \frac{1}{2}\log n + \frac{3}{2}$, since $n! > \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$ [15] and $\log e < \frac{3}{2}$. Hence, $\log\left(g\left(c\frac{2^n}{n}\right)\right) < 2^n\left(\frac{cn}{n} + \frac{5}{4n}\right) + \frac{1}{2}n + \frac{3}{2} \leq c \cdot 2^n + o(2^n) \ll \log(|B_n|)$, and using Lemma 2, we conclude that the number of languages acceptable by deterministic finite automata with at most $c\frac{2^n}{n}$ states is in $o(|B_n|)$. □

3.1 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 $\Sigma^{\leq n}$, with $|\Sigma| = 2$, where all languages in the respective

set are equiprobable. A different model is based on a stochastic process: Given a 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 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 form the previous subsection and the above described Bernoulli experiment are equivalent, if $p = \frac{1}{2}$.

Lemma 6 *Let Σ be a finite alphabet, S a finite set of words over Σ, and $0 < p < 1$. Let L be a random subset of S. The event 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: For every subset S' of the former type, take $S'' = S \setminus S'$. 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 $\mathbb{P}[L] = (\frac{1}{2})^{|L|}(1-\frac{1}{2})^{|S|-|L|} = \frac{1}{2^{|S|}}$. □

The latter model has some conceptual advantages for the average case study of the descriptional complexity of operations on finite languages. For instance, the following result is easily obtained:

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 $(S, 1-p_1)$.*

We proceed with a 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. These are primary ingredients for proving most of the results to come.

For (S,p) distributions, our main interest is devoted to the cases $S = \Sigma^n$ and $S = \Sigma^{\leq n}$, where Σ is an alphabet of size 2, and $p = \frac{1}{2}$. For these sets, we have seen that $\mathrm{sc}(L) = \Theta(\frac{2^n}{n})$ in the worst-case. So the presented estimate is far from sharp, unless p is very small compared to n. We determine the exact average value in the case $S = \Sigma^n$. The proof, obtained with a blend of combinatorial and probabilistic arguments, is omitted due to lack of space.

Theorem 8 *Let L be a (Σ^n, p)-distributed language with $|\Sigma| = 2$ and $0 \leq p \leq 1$. Then* $\mathbb{E}[\mathrm{sc}(L)] = 1 + \sum_{i=0}^{n} \sum_{k=1}^{2^{n-i}} \binom{2^{n-i}}{k} \left(1 - \left(1 - p^k(1-p)^{2^{n-i}-k}\right)^{2^i}\right)$.

In the case p is constant while n grows, we can also derive a simpler asymptotic lower bound on the expected value of the state complexity.

Theorem 9 *Assume $0 < p < 1$, and $S = \Sigma^n$ or $S = \Sigma^{\leq n}$ with $|\Sigma| = 2$. Let $H(p) = -p\log(p) - (1-p)\log(1-p)$ denote the entropy of p, and L be a (S,p)-distributed language. Then $\mathbb{E}[\mathrm{sc}(L)] > c\frac{2^n}{n}$, for every $c < H(p)$, provided n is large enough.*

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 bounds given in Theorem 3.

3.2 Unary Finite Languages

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

Lemma 10 *Let Σ be an unary alphabet, and L be a $(\Sigma^{\leq n}, p)$-distributed language with $0 < p < 1$. Then $\mathbb{E}(\mathrm{sc}(L)) = n + 2 - \frac{1-p}{p} + \frac{(1-p)^{n+2}}{p}$.*

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 we have $\mathrm{sc}(L) = k$ if and only if $k-2$ is the length of the longest word in L. An easy observation is $\mathbb{P}\left[\text{longest word in } L \text{ has length } k-2 \mid L \neq \emptyset\right] = p \cdot (1-p)^{n-k+2}$. Thus, setting $q = 1-p$, the expected value computes as $\mathbb{E}(\mathrm{sc}(L)) = q^{n+1} + \sum_{k=2}^{n+2} kpq^{n-k+2} = q^{n+1} + p(n+2)\sum_{k=0}^{n} q^k - p\sum_{k=0}^{n} kq^k$ and the claimed result is obtained using elementary calculus. □

Using Lemma 7, we obtain for the union of two $(\Sigma^{\leq n}, \frac{1}{2})$-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 it is the same as the operand, i.e., close to $n+1$.

4 Descriptional Complexity of Nondeterministic Finite Automata

Now let us turn our attention to the nondeterministic state and transition complexity of finite languages. A result in the same spirit as Theorem 4 but now concerning the size of nondeterministic finite automata was obtained in [11].

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

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

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 $O(\frac{2^n}{n})$ for both nondeterministic state and transition complexity. This cannot be improved substantially for the number of transitions, but for the number of states in a minimal nondeterministic finite automaton.

Lemma 12 *Assume L is a finite language with $L \subseteq \Sigma^{\leq n}$. 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 of final states equals $F = \{q_\lambda\} \cup \{\, p_\lambda \mid \lambda \in L \,\}$, 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 this heuristics for finding compact nondeterministic finite automata works pretty well on the average if we wish to keep the number of states as small as possible. Also, this construction is optimal in the worst case, as witnessed by the language family A_k in [10, 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 state complexity in the families F_n and B_n for three descriptional measures: Deterministic state complexity, nondeterministic state, and transition complexity.

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

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

$$\frac{1}{20} \cdot \frac{2^n}{n} < \mathrm{ntc}(L) < \frac{2^{n+4}}{n}, \qquad \text{and} \qquad \frac{2^{n-1}}{n} < \mathrm{sc}(L) < \frac{2^{n+3}}{n}.$$

Proof First, we note that the upper bounds are obtained by the worst-case estimates established in Lemma 12 and Theorem 3. The latter theorem states that $\mathrm{sc}(L) \leq (1 + o(1))\frac{2^{n+2}}{n}$, so the claimed upper bound holds for sufficiently large n. And the upper bound on nondeterministic transition complexity is explained by noting that this measure cannot exceed the deterministic state complexity by a factor of more than two. The lower bounds follow from Lemma 11 and the proof of Theorem 4, which can be summarized as

follows: The total number of languages acceptable by nondeterministic automata with at most $\frac{1}{2} \cdot \sqrt{2^n}$ states *or* by nondeterministic automata with at most $\frac{1}{20} \cdot \frac{2^n}{n}$ transitions *or* deterministic automata with at most $\frac{2^{n-1}}{n}$ states is in $o(2^{|S|})$. □

Our setup also allows us to give 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, but occasionally blowed up the number of transitions.[1] 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 acceptable by nondeterministic finite automata with a given number of states exhibit this behavior.

Theorem 14 *For every $k > 4$ and $|\Sigma| = 2$, there is a set T of finite languages over Σ such that for every $L \in T$ holds* $\mathrm{nsc}(L) < k$ *but* $\mathrm{ntc}(L) > \frac{k^2}{c \cdot \log k}$, *with some constant $c \leq 360$. Moreover the size of this set is greater than $2^{k^2/9-1}$.*

We note that a similar but by much weaker bound can be obtained from recent work on nondeterministic transition complexity [13]. A concrete example of a language family K_n in B_{2n} is presented there for which $\mathrm{ntc}(K_n) = \Omega(2^{n+c\sqrt{n}})$ holds, provided $c < \frac{1}{2}$ [13, Theorem 1(iv)]. In contrast, Lemma 12 implies that for this language holds $\mathrm{nsc}(K_n) < \frac{3}{\sqrt{2}} 2^n$. To compare this with the above result, set $k = \frac{3}{\sqrt{2}} 2^n$. Then we have still $\mathrm{ntc}(K_n) = \omega(k \cdot \mathrm{poly}(\log k))$, but also note that for every $\delta > 0$ holds $2^{n+c\sqrt{n}} = o(k^{1+\delta})$.

Acknowledgments Thanks to Felix Fischer for some useful discussion on the subject.

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, *Workshop on Implementing Automata*, number 2214 of LNCS, pages 1–12, 2001. Springer.

[2] F. Bassino and C. Nicaud. Enumeration and random generation of accessible automata. Preprint, Institut d'électronique et d'informatique Gaspard-Monge, Université de Marne-la-vallée, France, 2005.

[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, 2004.

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

[1] "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 [...]"

[5] J.-M. Champarnaud, G. Hansel, T. Paranthoën and D. Ziadi. Random Generation Models for NFAs. *Journal of Automata, Languages and Combinatorics*, 9(2–3):203–216, 2004.

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

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

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

[9] 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.

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

[11] G. Gramlich and G. Schnitger. Minimizing NFA's and regular expressions. In V. Diekert and B. Durand, editors, *Annual Symposium on Theoretical Aspects of Computer Science*, number 3404 of LNCS, pages 399–411, 2005. Springer.

[12] M. Holzer and M. Kutrib. State complexity of basic operations on nondeterministic finite automata. In J.-M. Champarnaud and D. Maurel, editors, *Conference on Implementation and Application of Automata*, number 2608 of LNCS, pages 148–157, 2003. Springer.

[13] Juraj Hromkovic and Georg Schnitger. NFAs with and without ϵ-transitions. In *International Colloquium on Automata, Languages and Programming*, number 3580 of LNCS, pages 385–396, 2005. Springer.

[14] C. Nicaud. Average state complexity of operations on unary automata. In M. Kutylowski, L. Pacholski, and T. Wierzbicki, editors, *Symposium on Mathematical Foundations of Computer Science*, number 1672 of LNCS, pages 231–240, 1999. Springer.

[15] H. Robbins. A remark on Stirling's formula. *American Mathematical Monthly*, 62:26–29, 1955.

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

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

[18] S. Yu. Chapter 2: Regular languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume I, pages 41–110, 1997. Springer.

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