# Technical Report No. 2006–515
# Lower Bounds for the Transition Complexity of NFAs

Michael Domaratzki
Jodrey School of Computer Science,
Acadia University,
Wolfville, Nova Scotia B4P 2R6, Canada,
email: mike.domaratzki@acadiau.ca

Kai Salomaa
School of Computing,
Queen's University,
Kingston, Ontario K7L 3N6, Canada,
email: ksalomaa@cs.queensu.ca

June 7, 2006

**Abstract**

We construct regular languages $L_n$, $n \geq 1$, such that any NFA recognizing $L_n$ needs $\Omega(\text{nsc}(L_n) \cdot \sqrt{\text{nsc}(L_n)})$ transitions. Here $\text{nsc}(L_n)$ is the nondeterministic state complexity of $L_n$. Also, we study trade-offs between the number of states and the number of transitions of an NFA. We show that adding one additional state can result in significant reductions in the number of transitions and that there exist regular languages $L_n$, $n \geq 2$, where the transition minimal NFA for $L_n$ has more than $c \cdot \text{nsc}(L_n)$ states, for some constant $c > 1$.

## 1 Introduction

Recently there has been much work on descriptional complexity, or state complexity, of deterministic finite automata (DFAs, see Section 2 for definitions). The results are surveyed and more references can be found, for example, in [5, 13, 22, 23, 24]. The state complexity of nondeterministic finite automaton (NFA) operations has been investigated by Holzer and Kutrib [10, 11]. The minimal DFA equivalent to an arbitrary DFA can be constructed efficiently, whereas Jiang and Ravikumar [16] have shown that minimization of NFAs is PSPACE-complete, even if the input is given as a DFA, and Malcher [21] has shown that minimization remains NP-complete for classes of NFAs that are nearly deterministic. A technique for proving lower bounds for the state complexity of NFAs has been given using fooling sets, see Hromkovič [12] or Glaister and Shallit [4]. Gruber and Holzer [7] show that it is computationally hard to decide whether lower bounds given by the extended fooling set technique can be reached.

The number of transitions gives, in some sense, a more realistic measure for the size of an NFA than the number of states. In a worst-case comparison of nondeterministic state complexity and transition complexity it has been recently established by Gruber and Holzer [8] and by Kari [18] that there exist finite languages $L_n$, $n \geq 1$, such that any NFA for $L_n$ needs $\Omega(\frac{\text{nsc}(L_n)^2}{\log(\text{nsc}(L_n))})$ transitions. Here $\text{nsc}(L)$ is the nondeterministic state complexity of $L$. The above shows that the nondeterministic transition complexity of regular languages, as a function of the nondeterministic state complexity, can approach the quadratic upper bound. The lower bounds of [8, 18] are proved using probabilistic methods and the results do not yield efficiently constructible languages having a corresponding lower bound for their transition complexity. A non-trivial transition complexity lower bound, albeit a weaker lower bound, for a concrete family of regular languages can be obtained from the work of Hromkovič and Schnitger [14].

Here we give an explicit construction of a family of regular languages $L_n$, $n \geq 1$, such that the nondeterministic state complexity of $L_n$ is $O(n)$ and any NFA for the language $L_n$ needs $\Omega(n \cdot \sqrt{n})$ transitions. The lower bound relies on nontrivial combinatorial properties, namely on the existence of projective planes of arbitrarily large degree [2, 17]. When considering the number of transitions as a function of the nondeterministic state complexity, our lower bound is better than the one obtained in [14] for concrete regular languages over a fixed alphabet. It should be noted that the family of languages in [14] was designed for the purpose of maximizing the increase of transition complexity when converting an $\varepsilon$-NFA to an NFA without $\varepsilon$-transitions, as opposed to maximizing transition complexity as a function of nondeterministic state complexity.

Our lower bound applies to a specific family of regular languages. A topic for further research would be to extend the result as a more general purpose technique for establishing lower bounds for transition complexity, in the spirit of the methods studied in [4, 12] for proving lower bounds for nondeterministic state complexity. The proof of our lower bound result introduces techniques that may turn out to be useful for work in this direction.

We also study the trade-offs between nondeterministic state complexity and the number of transitions needed by an NFA. By the *strict transition complexity* of a language $L$ we mean the number of transitions needed by any NFA for $L$ with $\mathrm{nsc}(L)$ states. It turns out that already allowing one additional state can cause a drastic reduction in the number of transitions, that is, there are languages $L_n$ such that the strict transition complexity of $L_n$ is $\Omega(n^2)$ but if an NFA for $L_n$ can use $\mathrm{nsc}(L_n) + 1$ states it needs only $O(n)$ transitions. We show that for each $k \geq 1$ there are languages that exhibit an analogous gap in transition complexity when comparing NFAs that allow, respectively, $k - 1$ and $k$ additional states compared to the size of the state-minimal NFA.

To conclude the introduction, we mention some earlier work on transition complexity. Upper and lower bounds for the number of transitions of an NFA equivalent to a given regular expression are obtained in [9, 15, 20]. There the lower bounds use variable size alphabets. Inapproximability results for minimizing the number of NFA transitions for a given regular language are established by Gramlich and Schnitger [6]. The nondeterministic transition complexity of operations on regular languages is studied by the current authors in [3]. We note that the transition complexity of NFAs seems not to be related to the ambiguity of NFAs. Leung [19] constructs for each $n \geq 1$ an exponentially ambiguous NFA $A_n$ with $n$ states such that any equivalent polynomially ambiguous NFA needs $2^n - 1$ states. However, $A_n$ has only $2n$ transitions.

## 2 Preliminaries

For $n \geq 1$ denote $[n] = \{1, \ldots, n\}$. The set of prefixes (respectively, suffixes) of a word $w \in \Sigma^*$ is denoted $\mathrm{pref}(w)$ (respectively, $\mathrm{suf}(w)$). The length of $w \in \Sigma^*$ is $|w|$.

An NFA is denoted as $A = (\Sigma, Q, q_0, Q_F, \delta)$ where $\Sigma$ is the input alphabet, $Q$ is the set of states, $q_0 \in Q$ is the start state, $Q_F \subseteq Q$ is the set of accepting states and $\delta \subseteq Q \times \Sigma \times Q$ gives the set of transitions. A state $q \in Q$ is useful if $q$ is reachable from the start state and some accepting state is reachable from $q$. It is well known that the set of useful states can be efficiently found [22] and, without loss of generality, we assume that the NFAs under consideration have only useful states. Unless otherwise mentioned, by an NFA we mean an automaton without $\varepsilon$-transitions. A DFA is an NFA with $|\{(q, a, q') \in \delta \ : \ q' \in Q\}| \leq 1$ for all $(q, a) \in Q \times \Sigma$.

The nondeterministic state complexity of a regular language $L$ is the smallest number of states of any NFA recognizing $L$. It is denoted as $\text{nsc}(L)$. Generally, we consider the transition complexity of a language $L$ as a function of $\text{nsc}(L)$.

**Definition 2.1** *Let $L$ be a regular language. The (nondeterministic)* transition complexity *of $L$, $\text{tc}(L)$, is the smallest number of transitions of any NFA that recognizes $L$.*

*For $k \geq 0$, the $k$-*strict transition complexity *of $L$, $\text{stc}_k(L)$, is the smallest number of transitions of any NFA $A$ for $L$ such that $A$ has at most $\text{nsc}(L) + k$ states.*

The 0-strict transition complexity of $L$ is simply called the *strict transition complexity* of $L$. For a regular language $L$, the strict transition complexity of $L$ is the smallest number of transitions needed by any NFA for $L$ that has $\text{nsc}(L)$ states. For any regular language $L$ and $k \geq 0$,

$$\text{nsc}(L) + 1 \leq \text{tc}(L) \leq \text{stc}_{k+1}(L) \leq \text{stc}_k(L). \tag{1}$$

# 3 Non-linear lower bound for NFA transition complexity

We construct a family of languages $L_n$, $n \geq 1$, such that any NFA for $L_n$ requires provably $\Omega(\text{nsc}(L_n) \cdot \sqrt{\text{nsc}(L_n)})$ transitions. The intuitive idea of the construction is that we define languages where the corresponding NFAs consist of two "parts" that require a large number of "connections", and the language is designed in a way that prevents "funneling" large numbers of connections through a single state.

## 3.1 Definition of a language associated with a binary relation

We begin by defining a class of finite languages with particular properties. Each language is associated with a binary relation on $[n]$, $n \geq 1$.

For $w = a_1 \cdots a_s$, $s \geq 1$, $a_i \in \Sigma$, $i = 1, \ldots, s$, and $1 \leq i, j \leq s$ the following notations are used for the prefix of $w$ of length $i$, and the suffix of $w$ beginning from the $j$th position of $w$:

$$w[\rightarrow i] = a_1 a_2 \cdots a_i, \quad w[j \leftarrow] = a_j a_{j+1} \cdots a_s.$$

Also for $1 \leq i, j \leq s$ denote

$$w[i, j] = \begin{cases} a_i a_{i+1} \cdots a_{i+j} \text{ if } i + j \leq s, \\ a_i a_{i+1} \cdots a_s, \text{otherwise.} \end{cases}$$

Above $w[i, j]$ is the subword of length $j + 1$ starting at the $i$th symbol of $w$, if this subword is "inside" of $w$ and otherwise $w[i, j]$ is the suffix of $w$ starting at the $i$th position.

Let $w_0$, $w_1$, $w_2$, ... be the enumeration of all non-empty words over alphabet $\{0, 1\}$ in length-lexicographic order. We define the following infinite word over the alphabet $\{0, 1, \#\}$:

$$\alpha = w_0 \# w_1 \# w_2 \# \ldots \;=\; 0\#1\#00\#01\#10\#11\#000\#\ldots$$

Let $\alpha_r$, $r \geq 1$, be the prefix of $\alpha$ of length $r$.

Consider a fixed $n \geq 1$ such that our binary relations will be defined as subsets of $[n] \times [n]$. We denote

$$m = n + \lceil \log n \rceil.^1$$

[1] By using endmarkers, it would be possible to use words of length exactly $n$. The additive term $\lceil \log n \rceil$ will not change bounds that ignore constant factors, and it will make parts of the notation more uniform. All our logarithms are to the base 2.

The value of $m$ depends on $n$. It is easy to verify that the following property holds.

$$\text{If } 1 \leq i \leq n, \text{ the subword } \alpha_m[i, \lceil \log m \rceil - 1] \text{ occurs in only one position in } \alpha_m. \tag{2}$$

For $1 \leq i \leq n$, we define

$$f(i) = \alpha_m[i, \lceil \log m \rceil - 1]. \tag{3}$$

Here $f(i)$ is the subword of $\alpha_m$ of length $\lceil \log m \rceil$ beginning at position $i$. By (2), $f(i)$ occurs only once in the word $\alpha_m$.

In the following of this section let $\Sigma = \{0, 1, \#, \$\}$. Let $\omega_n \subseteq [n] \times [n]$ be an arbitrary binary relation. Now we define the language $L(\omega_n) \subseteq \Sigma^*$ as follows:

$$L(\omega_n) = \{\ \alpha_m[\rightarrow i + \lceil \log m \rceil - 1]\ \$\ \alpha_m[j \leftarrow]\ \mid\ (i, j) \in \omega_n,\ 1 \leq i, j \leq n\ \}.$$

Intuitively, the language $L(\omega_n)$ consists of, for each $(i, j) \in \omega_n$, all words beginning with the prefix of $\alpha_m$ that ends with the unique occurrence of the subword $f(i)$, followed by the middle marker \$ and the suffix of $\alpha_m$ beginning with the unique occurrence of the subword $f(j)$.

Since $m \leq 2n$, the following lemma is immediate.

**Lemma 3.1** *Let $\omega_n$ be any binary relation on $[n]$. Then* $\mathrm{nsc}(L(\omega_n)) \in O(n)$. $\square$

Next we define some terminology associated with an arbitrary NFA recognizing some of the languages defined above. We use graph theoretic arguments and below we refer to transitions and states of the NFA interchangeably as edges and nodes, respectively, of the corresponding state graph.

Let $A = (\Sigma, Q, q_0, Q_F, \delta)$ be an NFA recognizing $L(\omega_n)$ and $k \geq 1$. We say that a state $q_2$ is *$k$-reachable* from a state $q_1$ if, for some word $w \in \Sigma^k$, $q_2 \in \delta(q_1, w)$. If the above holds we say also that $q_2$ is $k$-reachable from $q_1$ via the word $w$. Note that $k$-reachability refers to reachability using a word of length exactly $k$.

By a *funnel edge* (or transition) we mean any edge $e$ of the underlying graph of $A$ labeled by \$. A *funnel* of $A$ consists of a funnel edge $e$ together with

- all states $q \in Q$ such that the in-node of $e$ is $\lceil \log m \rceil$-reachable from $q$, these are called the *in-states* of the funnel associated with $e$; and,
- all states $q \in Q$ such that $q$ is $\lceil \log m \rceil$-reachable from the out-node of $e$, these are called the *out-states* of the funnel associated with $e$.

Below we establish some properties of $A$. Without loss of generality we can assume that $A$ has a single accepting state, $Q_F = \{q_f\}$. This follows from the observation that any word of $L(\omega_n)$ ends with a unique suffix of length $\lceil \log m \rceil$.

For $1 \leq i \leq n$ we define the following subsets of $Q$:

$$\begin{aligned} B(i) &= \delta(q_0, \alpha_m[\rightarrow (i-1)]), \\ C(i) &= \{q \in Q \mid q_f \in \delta(q, \alpha_m[(i + \lceil \log m \rceil) \leftarrow])\}. \end{aligned}$$

The set $B(i)$ consists of states that are reachable from the start state using a prefix of length $i - 1$ of some word in $L(\omega_n)$ that does not contain the middle marker \$ (the words of $L(\omega_n)$ have only one possible prefix without the marker \$ having length $i - 1$, but it is not the prefix of all words in $L(\omega_n)$). The set $C(i)$ consists of states that can accept a suffix of $\alpha_m$ starting from the $(i + \lceil \log m \rceil)$th position of $\alpha_m$.

The sets $B(i)$ and $B(j)$ (respectively, $C(i)$ and $C(j)$), $i \neq j$, may in general contain common elements. However, the crucial property that will be used in our lower bound estimate is that if a state $q$ belongs to some funnel then $q$ can belong to at most one of the sets $B(i)$, $1 \leq i \leq n$, (respectively, at most one of the sets $C(i)$, $1 \leq i \leq n$). This property will be stated in (4) and (5) below.

The below properties (F1)–(F3) follow from (2) and from the definition of the words $f(i)$, $1 \leq i \leq n$.

(F1) If a state $q \in B(i)$, $1 \leq i \leq n$, is in the funnel corresponding to edge $e$, then the in-node of $e$ is reachable from $q$ via the word $f(i)$ (as in (3)), and $f(i)$ is the only word of length $\lceil \log m \rceil$ with this property.

(F2) If a state $q \in C(i)$, $1 \leq i \leq n$, is in the funnel corresponding to edge $e$, then $q$ is reachable from the out-node of $e$ via word $f(i)$ (as in (3)), and $q$ is not reachable from the out-node of $e$ using any other word of length $\lceil \log m \rceil$.

(F3) There exist states $p \in B(i)$ and $q \in C(j)$ belonging to the same funnel if and only if $(i, j) \in \omega_n$, $1 \leq i, j \leq n$.

Since $f(i) \neq f(j)$ when $i \neq j$, from (F1) we get the following:

$$\text{If } q \in Q \text{ belongs to some funnel, then } q \text{ cannot be in distinct sets } B(i) \text{ and } B(j),\ i \neq j. \tag{4}$$

Recall that states belonging to a funnel associated with edge $e$ can reach $e$ using a path of length $\lceil \log m \rceil$ (or can be reached from $e$ using a path of length $\lceil \log m \rceil$). Analogously (F2) implies the following:

$$\text{If } q \in Q \text{ belongs to some funnel, then } q \text{ cannot be in distinct sets } C(i) \text{ and } C(j),\ i \neq j. \tag{5}$$

By the *count* of a set of funnels $\mathcal{F}$ we mean the cardinality of the set

$$\{(i, j) \mid (\exists p \in B(i))(\exists q \in C(j))(\exists F \in \mathcal{F}) : \ p \text{ is an in-node of } F \text{ and } q \text{ is an out-node of } F\ \}.$$

Now conditions (4) and (5) together with (F3) give us the following property concerning the set of all funnels of $A$.

**Claim 3.1** *If $\mathcal{F}$ consists of all funnels of the NFA $A$, then the count of $\mathcal{F}$ equals to the cardinality of $\omega_n$ (as a subset of $[n] \times [n]$).*

Now the crucial question is how does the count of a set of funnels relate to the number of transitions that the NFA needs to "realize" the funnels. Note that relations like

$$\begin{aligned} \text{less\_than}_n &= \{(i, j) \mid 1 \leq i < j \leq n\}, \\ \text{in\_equal}_n &= \{(i, j) \mid i \neq j, 1 \leq i, j \leq n\} \end{aligned}$$

require $\Omega(n^2)$ connections through the funnels and, in light of the seemingly strong properties (F1), (F2), (F3), such languages might look potentially promising for establishing non-linear lower bounds.

It is easy to see that the languages $L(\text{less\_than}_n)$ and $L(\text{in\_equal}_n)$ can be recognized by NFAs having $O(n \cdot \log n)$ transitions. However, it turns out to be difficult to prove lower bounds, and depending on the distribution of letters in the words $f(i)$, the languages could conceivably be recognized using much fewer transitions.

In the next subsection we consider binary relations that actually give a much better lower bound than $\Omega(n \cdot \log n)$.

### 3.2 One-overlapping families

Here we define a family of binary relations $\omega_n$, $n \geq 1$, such that $\mathrm{tc}(L(\omega_n))$, $n \geq 1$, is provably not linear in $\mathrm{nsc}(L(\omega_n))$. First we need some more definitions.

For $n > k \geq 1$, a *one-overlap* $(n,k)$*-family* is a collection of $n$ subsets of $[n]$,

$$\Delta(n,k) = \{D_1^{(n)}, \ldots, D_n^{(n)}\},\ \ D_i^{(n)} \subseteq [n],\ i = 1, \ldots, n, \tag{6}$$

such that $|D_i^{(n)}| = k$, $i = 1, \ldots, n$, and $|D_i^{(n)} \cap D_j^{(n)}| \leq 1$ for all $i \neq j$, $1 \leq i, j \leq n$.

We define $\phi : \mathbb{N} \longrightarrow \mathbb{N}$ by setting $\phi(n) = k_n$, where $k_n$ is the greatest integer such that a one-overlap $(n, k_n)$-family exists. It is easy to verify an $\Omega(\log n)$ lower bound for the function $\phi(n)$, however, we can get a better lower bound by relying on results from combinatorics.

We recall that a finite projective plane of order $q$ consists of a set of $q^2+q+1$ elements called "points" and a set of $q^2+q+1$ "lines" each consisting of $q+1$ points such that any two lines meet at a unique point, see, e.g., Cameron [2] or Jukna [17]. Thus, a projective plane of order $q$ gives a one-overlap $(q^2+q+1, q+1)$-family.

It is known that when $q$ is a power of a prime number, a projective plane of order $q$ exists [2, 17]. Furthermore, when $q$ is a prime power there is an elegant construction for a projective plane of order $q$. The construction depends on the existence of a finite field $\mathrm{GF}(q)$ containing precisely $q$ elements. The construction is outlined below, for details see, e.g., Anderson [1]. Let $T$ be the set of triples $\mathbf{x} = (x_0, x_1, x_2)$ of elements of $\mathrm{GF}(q)$ where at least one of $x_0$, $x_1$, $x_2$ is nonzero. Elements $\mathbf{x}$ and $\mathbf{y}$ of $T$ are said to be equivalent if $\mathbf{x} = z \cdot \mathbf{y}$ for some non-zero element $z$ of $\mathrm{GF}(q)$. The equivalence classes $[\mathbf{x}]$ of elements $\mathbf{x} \in T$ are taken as the points of the projective plane and it is easy to see that there are $q^2+q+1$ equivalence classes. An element $\mathbf{y} = (y_0, y_1, y_2) \in T$ defines a line $\ell(\mathbf{y})$ that consists of all points $[\mathbf{x}]$, $\mathbf{x} = (x_0, x_1, x_2)$, such that $y_0x_0 + y_1x_1 + y_2x_2 = 0$. It can be verified that a line $\ell(\mathbf{y})$ does not depend on the choice of the representing vector from $[\mathbf{y}]$, there are $q^2+q+1$ lines and each line consists of $q+1$ points.

Thus, when $q$ is a prime power,[2]

$$\phi(q^2+q+1) \geq q+1. \tag{7}$$

**Corollary 3.1** $\phi(n) \in \Omega(\sqrt{n})$.

In the following consider the case where $n$ is of the form

$$n = q^2+q+1, \text{ with } q \text{ a prime power.} \tag{8}$$

Let $\Delta(n, \phi(n)) = \{D_1^{(n)}, \ldots, D_n^{(n)}\}$ be a fixed one-overlap $(n, \phi(n))$-family as in (6). Define

$$\omega_n = \{(i,j) \mid j \in D_i^{(n)}\}. \tag{9}$$

Let $A = (\Sigma, Q, q_0, Q_F, \delta)$ be an arbitrary NFA recognizing the language $L(\omega_n)$, where $\omega_n$ is as in (9). Using the set of funnels of $A$, as defined in the previous subsection, we provide a lower bound for the number of transitions of $A$.

Since $\omega_n$ is defined using the one-overlapping family $\Delta(n, \phi(n))$, the condition (F3) implies that for any funnel $F$ of $A$ one of the following conditions holds:

[2] When $q$ is a prime power, the relation (7) has, in fact, equality by the De Bruijn–Erdös theorem [2].

- there exists $i \in [n]$ such that all in-states of $F$ are in $B(i)$, or
- there exists $j \in [n]$ such that all out-states of $F$ are in $C(j)$.

By the *graph representation of a set of funnels* $\mathcal{F}$ we mean the smallest subgraph $G$ of the state graph of the NFA $A$ such that, for all $F \in \mathcal{F}$, $G$ contains all paths from any in-node of $F$ to any out-node of $F$.

**Claim 3.2** *For any set of funnels $\mathcal{F}$, the number of edges in the graph representation of $\mathcal{F}$ is greater or equal to the count of $\mathcal{F}$.*

**Proof.** We prove the claim using induction on the number of funnels in $\mathcal{F}$. By properties (4) and (5), the claim clearly holds for any single funnel (actually, an individual funnel needs many more edges, see Remark 3.1 below).

Inductively, assume that the count of a collection of funnels $\mathcal{F} = \{F_1, \ldots, F_k\}$ is not greater than the number of edges in the graph representation of $\mathcal{F}$, and consider a collection of funnels $\mathcal{F}' = \{F_1, \ldots, F_k, F_{k+1}\}$.

Without loss of generality we assume that $F_{k+1}$ has in-states in only one set $B(x)$, $x \in \{1, \ldots, n\}$, and out-states in sets

$$C(j_1), \ldots, C(j_r), \tag{10}$$

where $j_1, \ldots, j_r$, are pairwise distinct, $r \geq 1$. The other possibility is completely symmetric.

We consider the two situations where $F_{k+1}$ can share some out-states with a funnel $F_i \in \mathcal{F}$, $1 \leq i \leq k$.

(i) Consider the case where $F_i$, $1 \leq i \leq k$, has an in-state in $B(x)$. Now, if $F_i$ has out-states in some set $C(j_s)$ as in (10), $1 \leq s \leq r$, the out-states of $F_{k+1}$ in the set $C(j_s)$ do not increase the count of the set of funnels $\mathcal{F}'$ since the pair $(x, j_s)$ is already included in the count of $\mathcal{F}$. Thus in this case, it does not matter if corresponding to $j_s$ we do not need to include an additional edge for the graph representation of $\mathcal{F}'$.

(ii) Next we consider the case where $F_{k+1}$ shares out-states with some funnel $F_i$, $1 \leq i \leq k$, that is not as above in (i), i.e.,

$$F_i \text{ does not have in-states in } B(x). \tag{11}$$

In this case $F_{k+1}$ can share with $F_i$ out-states in only one of the sets $C(j_s)$ as in (10), $1 \leq s \leq r$. This means that, in order to connect the state(s) of $C(j_s)$ with the state(s) of $B(x)$, the graph representation of $\mathcal{F}'$ needs for each shared set $C(j_s)$ at least one edge that does not appear in the graph representation of $\mathcal{F}$.

Above, note that if the graph representation of $\mathcal{F}'$ would try to "peel off", using a single edge, from the graph representation of $\mathcal{F}$ some state(s) belonging to $C(j_{s_1})$ and some state(s) belonging to $C(j_{s_2})$, $j_{s_1} \neq j_{s_2}$, then these states need to be connected in the graph representation of $\mathcal{F}$. Thus the states would need to be out-states of one funnel $F_i \in \mathcal{F}$, where $F_i$ is as in (11) (since the other possibility was considered in (i) before). Let $y \in [n]$, $y \neq x$, be such that the in-states of $F_i$ are in $B(y)$. Then $j_{s_1}, j_{s_2} \in D_x^{(n)} \cap D_y^{(n)}$, which is a contradiction since $\{D_1^{(n)}, \ldots, D_n^{(n)}\}$ is a one-overlap $(n, \phi(n))$-family. The situation in this case is summarized in Figure 1.

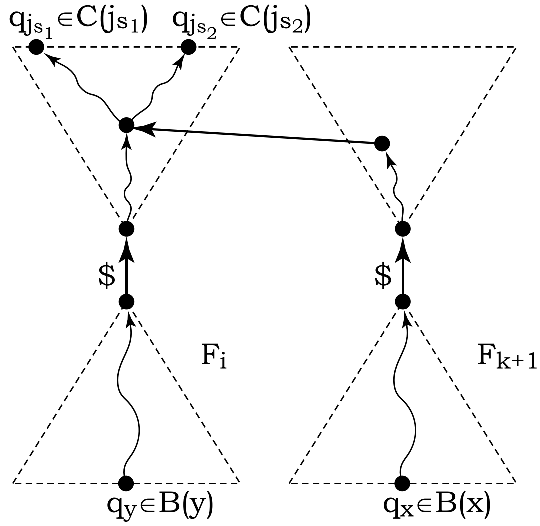


Figure 1: Case (ii) of the proof of Claim 3.2.

Above in (i) and (ii) we have seen that for each pair $(x, j_s)$ that is not already included in the count of $\mathcal{F}$, the graph representation of $\mathcal{F}'$ needs at least one edge that does not occur in the graph representation of $\mathcal{F}$. It follows that the count of $\mathcal{F}'$ is not greater than the number of edges in the graph representation of $\mathcal{F}'$. ■

**Remark 3.1** In the proof of Claim 3.2, note that an individual funnel $F$ naturally needs essentially more edges than the count of $\{F\}$. In particular, if $F$ has a single in-state (or a single out-state) this state has to be connected to the funnel edge by a path of length $\lceil \log m \rceil$. However, when considering a set of funnels we cannot guarantee that the different paths of length $\lceil \log m \rceil$ would not overlap, and therefore they are not included in the estimate.

Typically, the number of edges in the graph representation of a set of funnels $\mathcal{F}$ can be much larger than the count of $\mathcal{F}$. This happens, for example, if the NFA $A$ is ambiguous and some sets $B(i)$ or $C(j)$ are not singletons.

**Theorem 3.1** *Let $\omega_n$ be as in (9), where $n$ is as in (8). Then*

$$\mathrm{tc}(L(\omega_n)) \in \Omega(\mathrm{nsc}(L(\omega_n)) \cdot \sqrt{\mathrm{nsc}(L(\omega_n))}). \tag{12}$$

**Proof.** Let $A_n$ be any NFA for $L(\omega_n)$ and let $\mathcal{F}$ be the set of all funnels of $A_n$. By Claim 3.2, $\mathrm{tc}(L(\omega_n))$ is at least the count of $\mathcal{F}$.

Claim 3.1 gives that the count of $\mathcal{F}$ equals the cardinality of $\omega_n$ which is $n \cdot \phi(n)$. By Corollary 3.1, $\mathrm{tc}(L(\omega_n)) \in \Omega(n \cdot \sqrt{n})$. The relation (12) holds since by Lemma 3.1 we know that $\mathrm{nsc}(L(\omega_n)) \in O(n)$. ■

Except when $q$ is a prime power, very little is known about the existence of projective planes of order $q$. They do not exist for values 6 or 10. It is probably difficult to find good estimates for $\phi(n)$ except by relying on the existence of projective planes.

By modifying the definition of $\omega_n$ when $n$ is not of a form $q^2 + q + 1$ for a prime power $q$, in Theorem 3.1 we can easily define $\omega_n$ for all values of $n$ such that $\mathrm{tc}(L(\omega_n))$ will be monotonic with respect to $n$. Instead of (9) we can define $\omega_n$ using any collection of $n$ subsets of $[n]$ where the intersection of any two sets has cardinality at most one. Note that the proof of Theorem 3.1 does not use the property that the subsets have uniform size. In this way we can construct relations $\omega_n$ with monotonically increasing cardinality that is guaranteed to reach $n \cdot \sqrt{n}$ infinitely often.

The languages $L(\omega_n)$ are defined over a four-letter alphabet. It is easy to see that encoding the symbols over a binary alphabet does not cause any problems with the above argument that establishes the lower bound. It is sufficient to take in place of $\alpha$ an infinite binary sequence in which any prefix of length $n$ has only one occurrence of any subword of length $c \cdot \lceil \log n \rceil$, for some constant $c$, and encode the separation markers \$ by a binary sequence that does not occur anywhere else in the words of the language. For example, we can define $\alpha$ so that every symbol in an even position is a 1 and then encode \$ as 00.

**Corollary 3.2** *We can construct languages $L_n$, $n \geq 1$, over a binary alphabet such that* $\mathrm{nsc}(L_n) \in O(n)$ *and* $\mathrm{tc}(L_n) \in \Omega(n \cdot \sqrt{n})$.

To conclude this section, we compare the lower bound with the earlier result due to Hromkovič and Schnitger [14]. By a $\varepsilon$-NFA we mean an NFA that can have $\varepsilon$-transitions. The following constructive lower bound is established in [14].

**Proposition 3.1** [14] *There exist regular languages $L_n$, $n \geq 1$ (with an explicit definition provided), over a binary alphabet recognized by $\varepsilon$-NFAs having $O(n \cdot \log n)$ transitions such that any NFA without $\varepsilon$-transitions needs at least $n \cdot 2^{c \cdot \sqrt{\log n}}$ transitions for every $c < 1/2$.*

When viewing transition complexity as a function of nondeterministic state complexity, the lower bound from Corollary 3.2 is better. The result of Proposition 3.1 was developed for a different purpose, namely to measure the overhead involved in eliminating $\varepsilon$-transitions from NFAs.

# 4 Strict transition complexity

Here we study trade-offs between the number of states and the number of transitions, i.e., the situations where $\mathrm{stc_k}(L)$, $k \geq 0$, may be much larger than $\mathrm{tc}(L)$. As one expects, it is much easier to establish lower bounds for the $k$-strict transition complexity than general lower bounds for the number of transitions when the number of states is not fixed.

We show that, for any fixed $k \geq 1$, there exist languages $L_n$, $n \geq 2$, having $(k-1)$-strict transition complexity in $\Omega(n^2)$ but by allowing the use of one more state, i.e., a total of $k$ "additional" states compared to the size of the state minimal NFA, the number of transitions can be reduced to $O(n)$.

Let $n \geq 2$. Denote

$$L_{1,n} = (a^{n-1}b)^*\mathrm{pref}(a^{n-1}b) - (a^{n-1}b)^*, \quad L_{2,n} = \mathrm{suf}(c^{n-1}d)(c^{n-1}d)^*\mathrm{pref}(c^{n-1}d) \cap \{c,d\}^+.$$

Let $h : \{c,d\}^* \to \{a,b\}^*$ be the morphism defined by $h(c) = a$, $h(d) = b$. Note that $h(L_{2,n}) \neq L_{1,n}$ since in $h(L_{2,n})$ the first occurrence of $b$ may be preceded by fewer than $n-1$ symbols $a$.

Now we define for $k \geq 1$,

$$L_n[k] = \begin{cases} L_{1,n}(L_{2,n} \cdot h(L_{2,n}))^{(k-1)/2} L_{2,n} & \text{when } k \text{ is odd,} \\ L_{1,n}(L_{2,n} \cdot h(L_{2,n}))^{k/2} & \text{when } k \text{ is even.} \end{cases} \tag{13}$$

Intuitively, $L_n[k]$ is obtained by catenating to the language $L_{1,n} \cdot L_{2,n}$, $k-1$ marked copies of $L_{2,n}$ where consecutive copies are alternately over alphabets $\{a, b\}$ and $\{c, d\}$.

**Lemma 4.1** *Let $k \geq 1$ be fixed. For languages $L_n[k]$, $n \geq 2$, as in (13) we have*

(i) $\mathrm{nsc}(L_n[k]) = (k+1)n$,

(ii) $\mathrm{stc}_k(L_n[k]) \in O(n)$,

(iii) $\mathrm{stc}_{k-1}(L_n[k]) \in \Omega(n^2)$.

**Proof.** (i) It is easy to see that $L_n[k]$ can be recognized by an NFA with $(k+1)n$ states that form $k+1$ cycles of length $n$, $C_1, \ldots, C_{k+1}$, and each state of $C_i$ has a transition to each state of $C_{i+1}$, $1 \leq i \leq k$, with the exception that there are no transitions from the initial state of $C_1$ to states of $C_2$. For odd $i$ (respectively, even $i$), each transition from $C_i$ to $C_{i+1}$ is labeled with the appropriate symbol $c$ or $d$ (respectively, $a$ or $b$).

In order to establish that any NFA for $L_n[k]$ needs $(k+1)n$ states we introduce some terminology that will be used also in proving (iii) below. We observe that for each $n \geq 1$,

$$L_n[k] \subseteq \begin{cases} (\{a,b\}^+\{b,c\}^+)^{k/2}\{a,b\}^+ & \text{if } k \text{ is even,} \\ (\{a,b\}^+\{b,c\}^+)^{(k+1)/2} & \text{if } k \text{ is odd.} \end{cases} \tag{14}$$

Thus every word $w \in L_n[k]$ has a unique decomposition

$$w = w_1 w_2 \cdots w_{k+1}, \text{ where } w_i \in \{a,b\}^+ \text{ when } i \text{ is odd, and, } w_j \in \{c,d\}^+ \text{ when } j \text{ is even,} \tag{15}$$

$1 \leq i, j \leq k+1$. Above the subword $w_i$ is called the *$i$th component,* $1 \leq i \leq k+1$, of $w \in L_n[k]$.

We say that words not included in the right side of (14) are *illegal.*

Let $A = (\Sigma, Q, q_0, Q_F, \delta)$ be any NFA recognizing $L_n[k]$. It follows that for every transition $e \in \delta$ there exists $1 \leq i(e) \leq k+1$ such that in any computation of $A$ on $w \in L_n[k]$ that uses transition $e$, the symbol consumed by transition $e$ is from the $i(e)$th component of $w$. The value $i(e)$ is called the *level* of the transition $e$. Note that if, for some transition $e$, the level $i(e)$ would not be uniquely defined, $A$ would necessarily accept illegal words (because all states are useful).

Let $q$ be some state of $A$ and let $e_1$ and $e_2$ be any transitions (incoming or outgoing) connected to $q$. It is easy to see that $|i(e_1) - i(e_2)|$ is either 0 or 1, because otherwise $A$ would accept illegal words. Also,

$$\text{if } e_1 \text{ is an incoming transition and } e_2 \text{ an outgoing transition of } q, \text{ then } i(e_1) \leq i(e_2). \tag{16}$$

We note that $L_n[k]$ has words as in (15) where the $i$th component when $i$ is odd (respectively, even) is $(a^{n-1}b)^x$ (respectively, $(c^{n-1}d)^x$) for arbitrarily large integers $x$. Since we know that each transition has a unique level, it follows that $A$ must have $k+1$ cycles of length $n$ that do not share any transitions. Note that if $w_i$ is the $i$th component, every occurrence of $b$ (respectively, $d$) in $w_i$ except the first one, is preceded by exactly $n-1$ symbols $a$ (respectively, $c$) and this implies that the cycles could not be shorter than $n$. Now the condition (16) implies that any two cycles also cannot share a state. Thus $A$ needs $(k+1)n$ states.

(ii) It is easy to construct an NFA $A_{n,k}$ with $(k+1)n$ states where the number of transitions is linear in $n$. As above in (i), we can have disjoint cycles of length $n$, $C_1, \ldots, C_{k+1}$ and we can introduce "funnel states" $f_i$ such that there is a transition from each state of $C_i$ to $f_i$ and a transition from $f_i$ to each state of the cycle $C_{i+1}$, $1 \le i \le k$, with the exception that there is no transition from the $(n-1)$th state of $C_1$ to $f_1$. Additionally, consecutive funnel states are connected by transitions labeled with $c$ and $d$, or with $a$ and $b$, to simulate computations where we read only one symbol from the cycle in between.

We leave the details of the construction of the NFA $A_{n,k}$ to the reader. The NFA $A_{4,2}$ is depicted in Figure 2. In general, $A_{n,k}$ has $(k+1)n + k$ states and $(3k+1)n + 2k - 3$ transitions.

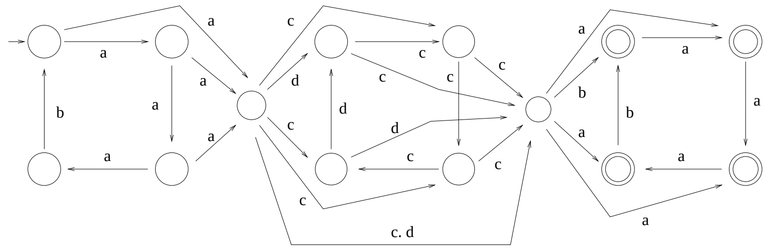


Figure 2: The NFA $A_{4,2}$ recognizing $L_4[2]$.

(iii) In the following assume that $A$ is an arbitrary NFA with at most $\mathrm{nsc}(L_n[k]) + k - 1 = (k+1)n + k - 1$ states. Since $k$ is fixed, without loss of generality we can assume that $n$ is much larger than $k$.

As above in (i) we see that $A$ must have $k+1$ disjoint cycles $C_1, \ldots, C_{k+1}$ of length $n$ where for odd $i$, $C_i$ consists of $n-1$ $a$-transitions and one $b$-transition having level $i$, and for even $j$, $C_j$ consists of $n-1$ $c$-transitions and one $d$-transition having level $j$. Since $n$ is chosen to be larger than $k$, it is not possible that the length of some cycle would be a multiple of $n$.

Note that the cycles $C_1, \ldots, C_{k+1}$ need not, in general, be unique and the computations of $A$ on a "long" component of a word in $L_n[k]$ may even go through several partially overlapping cycles. The cycles $C_1, \ldots, C_{k+1}$ are simply some (arbitrarily) chosen cycles of length $n$ with the property that all transitions in $C_i$ have level $i$, $1 \le i \le k+1$.

By a *supplementary state* of $A$ we mean a state $q$ that does not belong to any of the above chosen cycles $C_1, \ldots, C_{k+1}$. The set of supplementary states is denoted by $Q_{\mathrm{sup}}$. A supplementary state may be part of some other cycle, e.g., it can be connected by in-transitions and out-transitions to a cycle $C_i$ in a way that the resulting loop again has length $n$. However, an essential observation that the following argument relies on is that, since $k < n$,

$$\text{there can be no cycle consisting only of supplementary states.} \tag{17}$$

Note that from the form of words of $L_n[k]$ it follows immediately that $A$ cannot have a cycle of length less than $n$.

We say that a state $p$ is *outside-reachable* from a state $q$ if there exists a path from $q$ to $p$ where all the intermediate states are in $Q_{\mathrm{sup}}$. Note that the definition requires only that the intermediate states are in $Q_{\mathrm{sup}}$, the states $p$ and $q$ may or may not be supplementary, depending on the context. In particular, if there is a transition from $p$ to $q$, then $q$ is outside-reachable from $p$.

The following claim says, roughly, that if some state is connected to and from states in a cycle $C_i$ it can be connected only to few states of $C_i$, and the number of such states depends only on $k$.

**Claim 4.1** Let $q$ be a supplementary state and $1 \le i \le k+1$. Let $R_i$ be the set of states of $C_i$ that are outside-reachable from $q$, and $S_i$ be the set of states $p$ of $C_i$ such that $q$ is outside-reachable from $p$. We claim that

(i) If $S_i \neq \emptyset$, then $|R_i| < k$.

(ii) If $R_i \neq \emptyset$, then $|S_i| < k$.

*Proof of the claim.* We consider part (i) of the claim and the case where $i$ is odd. The case where $i$ is even is analogous. Assume that $S_i \neq \emptyset$. If $R_i = \emptyset$ there is nothing to prove. When $R_i \neq \emptyset$, $q$ is part of a cycle connected to $C_i$, and since the level of edges in $C_i$ is $i$, all edges in this cycle must be labeled by symbols $a$ and $b$.

For any supplementary state $s$, if $s$ is outside-reachable from the cycle $C_i$, then there can be a direct transition from $s$ to at most one state of $C_i$. Note that if there would be transitions from $s$ to two distinct states of $C_i$ this would produce a cycle of length less than $n$. Since the number of supplementary states is $k-1$, the above means that there can be paths originating from the state $q$, where the intermediate states are all in $Q_{\text{sup}}$, to at most $k-1$ states of $C_i$.

Exactly the same argument works for (ii), we just consider the cycles in reversed direction.

This concludes the proof of Claim 4.1 and we continue with the proof of Lemma 4.1.

We say that a supplementary state $q$ is an $(i, i+1)$*-funnel,* $1 \leq i \leq k$, if $q$ is outside-reachable from at least $k$ states of $C_i$, and, at least $k$ states of $C_{i+1}$ are outside-reachable from $q$.

First we observe that is $q \in Q_{\text{sup}}$ is an $(i, i+1)$-funnel and a $(j, j+1)$-funnel where $i+1 < j$, then $q$ is reachable from states of $C_j$ and states of $C_{i+1}$ are reachable from $q$ which produces a cycle that necessarily violates (16).

Secondly, if $q \in Q_{\text{sup}}$ is an $(i, i+1)$-funnel and an $(i+1, i+2)$-funnel, then $q$ would outside-reachable from at least $k$ states of $C_{i+1}$, and at least $k$ states of $C_{i+1}$ would be outside-reachable from $q$. This is impossible by Claim 4.1.

Thus, we have shown that any supplementary state can be an $(i, i+1)$-funnel for at most one value $1 \leq i \leq k$. By the pigeon-hole principle there exists $i_1 \in \{1, \ldots, k\}$ such that there is no $(i_1, i_1+1)$-funnel. To simplify notations we assume that $i_1$ is odd. The case for even $i_1$ is similar.

We note that $A$ has to accept words $w$ as in (15), where the $i_1$th component is of the form

$$(a^{n-1}b)^x a^r, \;\; 0 \leq r \leq n-1, \;\; x \geq 1, \tag{18}$$

and the $(i_1+1)$th component is of the form

$$c^s(dc^{n-1})^y d, \;\; 0 \leq s \leq n-1, \;\; y \geq 1. \tag{19}$$

When $i_1 = 1$, the condition in (18) needs to exclude the case $r = 0$. The modification does not influence the below argument, and we do not mention this case separately below.

Since the number of supplementary states $k$ is smaller than $n$, in accepting computations on words with $i_1$th and $(i_1+1)$th components as in (18) and (19), respectively, $A$ must enter the cycles $C_{i_1}$ and $C_{i_1+1}$. The $i_1$th components (18) can end with any suffix $a^r$, $0 \leq r \leq n-1$ corresponding to different states of $C_{i_1}$ and need to be connected to any prefix $c^s$, $0 \leq s \leq n-1$ of the $(i_1+1)$th component (19). That is, the edge of $C_{i_1}$ labeled by $b$ must be connected to the edge of $C_{i_1+1}$ labeled by $d$ by

$$\text{any path of the form } a^r c^s,\ 0 \leq r, s \leq n-1. \tag{20}$$

Consider a fixed supplementary state $q$. If $q$ is outside-reachable from more than $k$ states of $C_{i_1}$, then at most $k-1$ states of $C_{i_1+1}$ are outside-reachable from $q$, since otherwise $q$ would be an $(i_1, i_1+1)$-funnel. To get a very rough upper bound, we can assume that $q$ is outside-reachable from all $n$ states of $C_{i_1}$ by as many cycle-free paths as can go through all the $k-1$ supplementary states. (This is a worst case estimate and, in particular, would mean that the same supplementary states could not be used between cycles $C_j$ and $C_{j+1}$ where $|i_1 - j| \geq 2$.)

By (17), $q$ can be outside-reachable from a fixed state $p$ of $C_{i_1}$ only by a finite number of paths that depends only on $k$, (i.e., the number of paths does not depend on $n$). Thus, at most $n \cdot (k-1) \cdot f(k)$ of the paths (20) can go through $q$ where $f(k)$ is the number of cycle-free paths through at most $k-1$ states that use alphabet $\{a, b, c, d\}$. The other case, where at least $k$ distinct states of $C_{i_1+1}$ are outside reachable from $q$, is completely symmetric.

Thus at most $c_k \cdot n$ of the $n^2$ paths (20) can go through a given supplementary state, where $c_k$ is a constant depending on $k$. Since the number of supplementary states is $k-1$, the NFA $A$ needs $\Omega(n^2)$ direct transitions between states of $C_{i_1}$ and $C_{i_1+1}$. ■

By Lemma 4.1 we have the following result.

**Theorem 4.1** *Let $k \geq 1$ be fixed. There exist regular languages $L_n[k]$, $n \geq 2$, such that*

(i) $\mathrm{stc}_k(L_n[k]) \in O(\mathrm{nsc}(L_n[k]))$,

(ii) $\mathrm{stc}_{k-1}(L_n[k]) \in \Omega((\mathrm{nsc}(L_n[k])^2)$.

The languages $L_n[k]$ in (13) are defined over a four letter alphabet. Using any reasonable encoding, the result of Theorem 4.1 holds also for languages over a binary alphabet.

Note that Theorem 4.1 (i) and (1) imply that also $\mathrm{tc}(L_n[k]) \in O(\mathrm{nsc}(L_n[k]))$. In particular, by Lemma 4.1, for any fixed integer $k$ there exist languages $L_n$, $n \geq 2$, with $\mathrm{tc}(L_n) \in O(n)$, such that any NFAs for $L_n$ with at most $\mathrm{nsc}(L_n) + k$ states need $\Omega(n^2)$ transitions.

Finally, we show that there exist families of regular languages where the number of "additional" states in transition-minimal NFAs is not bounded by any constant. We show that the number of states required by transition minimal NFAs for a family of regular languages may be $c$ times the size of the state minimal NFAs for the same languages, for some $c > 1$.

**Theorem 4.2** *There exist regular languages $L_n$, $n \geq 2$, such that* $\mathrm{nsc}(L_n) = 5n - 3$ *and any transition minimal NFA for $L_n$ has at least $6n - 4$ states.*

**Proof.** Let $\Gamma = \{a, b, c, d, e\}$, $\Sigma = \Gamma \cup \{\$\}$ and define

$$L_n = \{x^i \$ y^j \mid x \in \{a, b, c\},\ y \in \{d, e\},\ i + j = n\} \cup \{x^n \mid x \in \Gamma\}.$$

Consider the set of pairs of words

$$P_n = \{(\varepsilon, a^n), (a^n, \varepsilon)\} \cup \{(x^i, x^j) \mid x \in \Gamma,\ i + j = n,\ 1 \leq i, j \leq n - 1\}. \tag{21}$$

The set $P_n$ satisfies the conditions of the fooling set lower bound technique of, e.g., Hromkovič [12] or Glaister and Shallit [4, Thm. 1]. It follows that any NFA for $L_n$ needs at least $|P_n| = 5n - 3$ states. On the other hand, it is easy to construct an NFA for $L_n$ ($n \geq 2$) with $5n - 3$ states and therefore $\mathrm{nsc}(L_n) = 5n - 3$.

Let $A = (\Sigma, Q, q_0, Q_F, \delta)$ be an arbitrary NFA for $L_n$. We say that a state $q \in Q$ has *depth* $r$, $r \geq 0$, if $q$ is reachable from $q_0$ on input $w$ where $w$ has exactly $r$ symbols of $\Gamma$ (that is, the symbol \$ is ignored when considering depth). By the *depth* of a transition $(q_1, x, q_2) \in \delta$, $q_1, q_2 \in Q$, $x \in \Sigma$, we mean the depth of $q_1$. Since all words of $L_n$ have exactly $n$ symbols of $\Gamma$ (and all states of $Q$ are useful), it follows that any state of $Q$ and any transition of $\delta$ has a unique depth.

Let $R_{x,i} \subseteq Q$ be the set of states reachable from $q_0$ on input word $x^i$, $x \in \{a,b,c,d,e\}$, $1 \le i \le n$. From (21) we know that $R_{x,i} \cap R_{y,j} = \emptyset$ when $(x,i) \neq (y,j)$, $x, y \in \{a,b,c,d,e\}$, $1 \le i,j \le n-1$. All the sets $R_{x,n}$ may naturally coincide for different $x \in \{a,b,c,d,e\}$.

Hence for each $1 \le i \le n-1$, $A$ has at least five states of depth $i$. If $A$ has only five states of depth $i$, it needs at least 11 transitions of depth $i$ (i.e., 11 transitions originating from states of depth $i$). This is because for $x \in \{a,b,c,d,e\}$, $1 \le i \le n-1$, there must be a transition labeled by $x$ from a state of $R_{x,i}$ to a state of $R_{x,i+1}$, and for each $x \in \{a,b,c\}$, and $1 \le i \le n-1$, there must be a \$-transition from the state of $R_{x,i}$ to the state of $R_{y,i}$, where $y \in \{d,e\}$. The number of transitions having depth $i$ can be reduced by one as follows. We introduce a new state $p_i$ with incoming \$-transitions from $R_{x,i}$, for each $x \in \{a,b,c\}$, such that $p_i$ has outgoing transitions labeled by $d$ and $e$, respectively, to a state of $R_{d,i+1}$ and a state of $R_{e,i+1}$, respectively. Here the new state $p_i$ has depth $i$ since the source and target of a \$-transition will have same depth. Since we have observed that each state has a unique depth, the changes made for depth $i$ states and transitions are localized to depth $i$. In particular, the new state $p_i$ cannot be used as the target of \$-transitions originating from any state of depth $j \neq i$. It follows that for $j \neq i$ such that there are only five states of depth $j$, the NFA $A$ needs at least 11 transitions of depth $j$, and to reduce this number we need to add a new state of depth $j$.

Thus, an NFA that uses a minimal number of transitions has to have at least 6 states of each depth $1 \le i \le n-1$. ■

## 5 Conclusion

We have given an explicit construction of regular languages $L_n$, $n \ge 1$, having nondeterministic state complexity in $O(n)$, such that the transition complexity of $L_n$ is $\Omega(n^{\frac{3}{2}})$. This does still not reach the lower bound $\Omega(\frac{n^2}{\log n})$ obtained using probabilistic combinatorial methods [8, 18]. Naturally the main open question is whether it is possible to prove that for all families of regular languages $L_n$, $n \ge 1$, with $\mathrm{nsc}(L_n) \in O(n)$, the transition complexity satisfies $\mathrm{tc}(L_n) \in o(n^2)$.

We would like to have more general purpose tools for proving transition complexity lower bounds, in the spirit of the techniques considered in [4, 12] for nondeterministic state complexity lower bounds. It remains to be seen whether the notions associated with funnels and one-overlapping families that were introduced for the result of Theorem 3.1 can be extended in this way.

Theorem 4.2 gives a family of regular languages $L_n$ where any transition minimal NFA for $L_n$ has at least $c \cdot \mathrm{nsc}(L_n)$ states where $c \approx 6/5$. By increasing the size of the alphabet, the constant $c$ can be somewhat increased. However, we do not know how to construct a language $L$, where the transition minimal NFA would need, for example, $2 \cdot \mathrm{nsc}(L)$ states.

## References


[1] I. Anderson, *Combinatorial Designs: Construction Methods,* John Wiley & Sons, 1990.

[2] P.J. Cameron, *Combinatorics: Topics, Techniques, Algorithms,* Cambridge University Press, 1996.

[3] M. Domaratzki and K. Salomaa, Transition complexity of language operations. *Proceedings of Descriptional Complexity of Formal Systems, DCFS 2006,* to appear.

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

[5] J. Goldstine, M. Kappes, C.M.R. Kintala, H. Leung, A. Malcher and D. Wotschke, Descriptional complexity of machines with limited resources, *Journal of Universal Computer Science* **8** (2002) 193–234.

[6] G. Gramlich and G. Schnitger, Minimizing NFA's and regular expressions, *Proceedings of STACS 2005,* Lecture Notes in Computer Science **3404**, Springer–Verlag, 2005, pp. 399–411.

[7] H. Gruber and M. Holzer, Finding lower bounds for nondeterministic state complexity is hard. *Proceedings of DLT 2006,* (O. H. Ibarra and Z. Dang, Eds.) Lecture Notes in Computer Science **4036**, Springer, 2006.

[8] H. Gruber and M. Holzer, Results on the average state complexity of finite automata accepting finite languages. *Proceedings of Descriptional Complexity of Formal Systems, DCFS 2006,* to appear.

[9] Ch. Hagenah and A. Muscholl, Computing $\varepsilon$-free NFA from regular expressions in $O(n\log^2(n))$ time. *RAIRO Theoret. Inform. Appl.* **34** (2000) 257–277.

[10] M. Holzer and M. Kutrib, Unary language operations and their nondeterministic state complexity, *Proceedings of DLT 2002,* Lecture Notes in Computer Science **2450**, Springer–Verlag, 2003, pp. 162–172.

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

[12] J. Hromkovič, *Communication Complexity and Parallel Computing*, Springer–Verlag, 1997.

[13] J. Hromkovič, Descriptional complexity of finite automata: Concepts and open problems, *Journal of Automata, Languages and Combinatorics* **7** (2002) 519–531.

[14] J. Hromkovič and G. Schnitger, NFAs with and without $\varepsilon$-transitions. *Proceedings of ICALP 2005*, Lecture Notes in Computer Science **3580**, Springer, 2005, pp. 385–396.

[15] J. Hromkovič, S. Seibert and T. Wilke, Translating regular expressions in small $\varepsilon$-free nondeterministic finite automata. *J. Comput. System Sci.* **62** (2001) 565–588.

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

[17] S. Jukna, *Extremal Combinatorics with Applications in Computer Science*, EATCS Texts in Theoretical Computer Science, Springer–Verlag, 2001.

[18] J. Kari, personal communication, April 2006.

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

[20] Yu. Lifshits, A lower bound on the size of $\varepsilon$-free NFA corresponding to a regular expression. *Inform. Proc. Lett.* **85** (2003) 293–299.

[21] A. Malcher, Minimizing finite automata is computationally hard, *Theoret. Comput. Sci.* **327** (2004) 375–390.

[22] S. Yu, Regular languages, in *Handbook of Formal Languages,* Vol. I, (G. Rozenberg and A. Salomaa, Eds.), Springer–Verlag, 1997, pp. 41–110.

[23] S. Yu, State complexity of regular languages, *Journal of Automata, Languages and Combinatorics* **6** (2001) 221–234.

[24] S. Yu, State complexity of finite and infinite regular languages, *Bulletin of the EATCS,* **76** (2002) 142–152.