# Descriptional Complexity of Determinization and Complementation for Finite Automata

**Aniruddh Gandhi** **Nan Rosemary Ke** **Bakhadyr Khoussainov**

Department of Computer Science, University of Auckland
Private Bag 92019, Auckland, New Zealand
{agan014, nke001}@aucklanduni.ac.nz
bmk@cs.auckland.ac.nz

## Abstract

In this paper we study the subset construction that transforms nondeterministic finite automata (NFA) to deterministic finite automata (DFA). It is well known that given a $n$-state NFA, the subset construction algorithm produces a $2^n$-state DFA in the worst case. It has been shown that given $n, m$ ($n < m \leq 2^n$), there is a $n$-state NFA $\mathcal{N}$ such that the minimal DFA recognizing $L(\mathcal{N})$ has $m$ states. However this construction requires $O(n^2)$ number of transitions in the worst case. We give an alternative solution to this problem that requires asymptotically fewer transitions. We also investigate the question of the complementation of NFA. In this case also, it known that given $n, m$ ($n < m \leq 2^n$), there exists a $n$-state NFA $\mathcal{N}$ such that the minimal NFA recognizing the complement of $L(\mathcal{N})$ needs $m$ states. We provide regular languages such that given $n, k$ ($k > 1$ and $n > k$), the NFA recognizing these languages need $n$ states and the NFA recognizing their complement needs $(k+1)n - (k+1)^2 + 2$ states. Finally we show that for given $n, k > 1$, there exists a $O(n)$-state NFA $\mathcal{A}$ such that the minimal NFA recognizing the complement of $L(\mathcal{A})$ has between $O(n^{k-1})$ and $O(n^{2k})$ states. Importantly however, the constructed NFA's have a small number of transitions, typically in the order of $O(n)$ or $O(n^2/log_2(n))$. These are better than the comparable results in the literature.

*Keywords:* Finite automata, state complexity, the subset construction, determinization, exponential blow-up, complementation.

## 1 Introduction

The subset construction is one of the fundamental constructions in automata theory that converts nondeterministic finite automata into equivalent deterministic automata. Under the subset construction, the states of the constructed DFA are subsets of the underlying NFA. Therefore, if the underlying NFA has $n$ states the then resulting equivalent DFA has at most $2^n$ states. Hence, the cost of determinization is an exponential blow-up in the number of states (Rabin & Scott 1959). In (Moore 1971) it was shown that this blow-up in the number of states is sharp. This sharpness result implies hardness for the complementation problem as well. Namely, for any $n$ there exists an $n$ state NFA recognizing a language $L$ such that $2^n$ number of states are needed for a DFA to recognize the complement of the language $L$.

In general, the study of the descriptional complexity of regular languages is one of the active areas of current research in the theory of finite automata (Yu 2005). An important measure for the descriptional complexity of regular languages is *state complexity.* Recall that given a regular language $L$, its *NFA-state complexity (DFA-state complexity)* is the number of states in a minimal NFA (minimal DFA) that recognizes $L$. There have been a series of results that study the state complexity of the Boolean operations, the concatenation and the Kleene-star operation on regular languages. For instance, in (Holzer & Kutrib 2003*b*) it is shown that $n+m+1$ states are necessary and sufficient to recognize the union of regular languages $L_1$ and $L_2$, recognized by $n$ state and $m$ state NFA, respectively. Similarly, in (Holzer & Kutrib 2003*b*) it is shown that $n \cdot m$ states are necessary and sufficient to recognize the intersection of regular languages $L_1$ and $L_2$, recognized by $n$ state and $m$ state NFA, respectively. The papers (Holzer & Kutrib 2003*b*) and (Jirásek et al. 2005) study the state complexity of other operations. Also, there has recently been some work on the study of average state complexity of regular languages and operations thereon.

Another measure for the descriptional complexity of regular languages is the *transition complexity.* For a given regular language $L$, its transition complexity is the number of transitions in the minimal NFA recognizing $L$. Transition complexity of a regular language seems to be a better measure of the descriptional complexity of a regular language since the transitions of the minimal NFA are needed to completely specify a regular language. Moreover, the transition complexity of a regular language $L$ may be exponentially greater than the NFA-state complexity of $L$. The papers (Gramlich & Schnitger 2007, Schnitger 2006) investigate the transition complexity of regular languages. Our paper fits in the realm of these investigations.

In this paper we revisit the subset construction that transforms NFA to DFA and investigate the problems around the following questions: (1) Given $n$ and $m$ such that $n \leq m \leq 2^n$, does there exist a regular language whose NFA-state complexity is $n$ and its DFA-state complexity is $m$? In (Jirásek et al. 2007) and (Jirásková 2008), it has been shown that it is possible to fill in this exponential gap. Here we seek to provide a solution to this problem which has asymptotically fewer transitions than the constructions of (Jirásek et al. 2007) and (Jirásková 2008). (2) Given $n$ and $m$ with $n \leq m \leq 2^n$, does there exist a regular language whose NFA-state complexity is $n$ such that the NFA-state complexity of the complement of the language is $m$? This question is answered in the affirmative by (Jirásek et al. 2005) and (Jirásková 2008). Again we would like to provide an alternative solution which has has asymptotically fewer transitions. Below we outline some of the known results related to these questions.

In (Piotr Berman 1977) it is shown that for every $n$ there exists a language $L$ whose NFA-state complexity is $n$ but DFA state complexity is $2^{n-1}$. Interestingly the complement of this language is recognized by an NFA with $O(n)$ states. In other words, the complementation problem for the language $L$ is easy in the class of nondeterministic finite automata. In (Holzer & Kutrib 2003*b*) a language $M$ is constructed whose NFA-state complexity is $n$ but whose NFA-state complexity for the complement of $M$ is $2^{n-2}$. In other words, the complementation problem for the language $M$ is hard in the class of nondeterministic finite automata. A natural question arises whether one can fill in the exponential gap.

In (Jirásek et al. 2005) for every $n$ and $m$ such that $n \leq m \leq 2^n$ a regular language $L$ is constructed such that its NFA state complexity is $n$ and the NFA-state complexity of the complement is $m$. These results also show that for every $n$ and $m$ such that $n \leq m \leq 2^n$ there exists a regular language whose NFA-state complexity is $n$ and whose DFA-state complexity is $m$. However, these precise bounds are obtained in the expense of increasing the alphabet size exponentially on $n$. The authors of (Jirásek et al. 2005) pose the problem if the sizes of the alphabets can be controlled. For instance, can the sizes of alphabets be dependent on $n$ linearly or be of a fixed size. In (Jirásek et al. 2007), the authors prove that for for every $m, n$ such that $n \leq m \leq 2^n$, there exists a $n$-state NFA whose DFA state complexity is $m$ for a fixed four letter alphabet. Furthermore in (Jirásková 2008), the authors prove that for every $m, n$ such that $n \leq m \leq 2^n$, there exists a $n$-state NFA $\mathcal{A}$ such that the NFA state complexity of the complement of $L(\mathcal{A})$ is $m$ for a fixed five letter alphabet. The question of whether similar results can be achieved using a binary alphabet is still open.

In this paper we present asymptotic solutions to the problems posed. The languages we construct are over either binary alphabets or alphabets that depend on $n$ linearly. These languages exhibit the same behavior as the languages in (Jirásek et al. 2007) but the bounds on the number of states are not sharp and the size of the alphabet varies linearly with $n$. However the $n$-state NFA constructed in (Jirásek et al. 2007) have $O(n^2)$ transitions in the worst case. The $n$-state NFA constructed by us have asymptotically fewer transitions than the NFA constructed by the authors of (Jirásek et al. 2007) in the worst case.

More precisely, we construct the following languages:

1. For every $k > 1$ there exists a regular language $L_n$ over a $k$-letter alphabet, where $n > k$, such that a minimal NFA recognizing $L_n$ needs exactly $n$ states and the minimal DFA recognizing $L_n$ needs exactly $(k+1) \cdot n - c$ states and $O(n)$ transitions, where $c = (k+1)^2 - 2$ (Theorem 1).

2. For every $n = k + m$ there exists a regular language $L_n$ over the binary alphabet such that the minimal NFA recognizing $L_n$ needs exactly $n$ states and the minimal DFA needs exactly $p \cdot n$ states and $O(n)$ transitions (Theorem 2). Unlike in (Jirásek et al. 2007), we use a binary alphabet instead of a four letter alphabet.

3. For every $k > 1$ there exists a regular language $L_n$ over a $k$-letter alphabet such that the minimal NFA $\mathcal{A}$ recognizing $L_n$ needs $n$ states, where $n > k$, and the minimal DFA recognizing $L_n$ has asymptotically $n^k$ states. The NFA $\mathcal{A}$ has $O(\frac{n^2}{log_2 n})$ transitions which is asymptotically fewer than the $O(n^2)$ transitions required by the NFA described in (Jirásek et al. 2007) in the worst case (Theorem 3).

4. For every $k > 1$ there exists a regular language $L_n$ over the $k$-letter alphabet, where $n > k$, such that the minimal NFA recognizing $L_n$ needs exactly $n$ states and the minimal NFA recognizing the complement of $L_n$ needs exactly $(k+1)n - c$ states and $O(n)$ transitions, where $c = (k+1)^2 - 2$ (Theorem 4).

5. For every $k > 1$ there exists a regular language $L_n$ over the $k$-letter alphabet, where $n > k$, such that the minimal NFA $\mathcal{A}$ recognizing $L_n$ needs exactly $n$ states and the minimal NFA recognizing the complement of $L_n$ needs between $O(n^{k-1})$ and $O(n^{2k})$ states. Moreover, in the worst case $\mathcal{A}$ has $O(\frac{n^2}{log_2 n})$ transitions in the worst case, which is asymptotically fewer than the $O(n^2)$ transitions of the NFA described in (Jirásková 2008) (Theorem 5).

The outline of this paper is as follows. The next section gives basic definitions and introduces a necessary notation. In our proofs we use Myhill-Nerode theorem that is also stated in the next section. Section 3 is devoted to proving Theorems 1 and 2. Section 4 proves Theorem 3. In Section 5 we provide a proof of Theorems 4 and 5.

## 2 Basic Notations and Definitions

A *deterministic finite automaton* (DFA) $\mathcal{A}$ is a 5-tuple $\langle S, \Sigma, \delta, s_0, F\rangle$ such that:

1. S is a finite set of states.
2. $\Sigma$ is an alphabet.
3. $\delta : S \times \Sigma \rightarrow S$ is the transition function.
4. $s_0 \in S$ is the initial state.
5. F is the set of accepting states.

A *nondeterministic finite automaton* (NFA) $\mathcal{A}$ is a 5 tuple $\langle S, \Sigma, \delta, s_0, F\rangle$ such that:

1. S is a finite set of states.
2. $\Sigma$ is an alphabet.
3. $\delta : \ S \times \Sigma \rightarrow 2^S$ is the transition function.
4. $s_0 \subseteq S$ is the set of initial states.
5. F is the set of accepting states.

For an alphabet $\Sigma$, let $\Sigma^*$ denote the set of all words over the alphabet, let $\lambda$ denote the empty string and $\Sigma^+ = \Sigma^* \backslash \{\lambda\}$. For $\sigma \in \Sigma$, $\sigma^m$ denotes letter $\sigma$ concatenated $m$ times, $\sigma^+ = \sigma^* \backslash \{\lambda\}$ and $\sigma^0 = \lambda$.

We define $\delta^+ : S \times \Sigma^+ \rightarrow 2^S$ recursively by:

1. $\delta^+(s, \sigma) = \delta(s, \sigma)$ and
2. $\delta^+(s, w \cdot \sigma) = \delta(\delta^+(s, w), \sigma)$

where $s \in S$, $\sigma \in \Sigma$ and $w \in \Sigma^+$. Here, for each $X \subset S$, we set $\delta(X, \sigma) = \cup_{s \in X} \delta(s, \sigma)$.

A run of the automaton $\mathcal{A}$ on the word $v = \sigma_1 \sigma_2 ... \sigma_n$ is a sequence of states $s_0, s_1 ... s_{n-1}, s_n$, such that $s_0$ is the initial state and $s_{i+1} \in \delta(s_i, \sigma_i)$. If for this run $s_n \in F$ then we say that the run is *accepting*. The automaton accepts the word $v$ if it has an accepting run on $v$. *The language accepted by an automaton $\mathcal{A}$*, denoted by $L(\mathcal{A})$, is as follows:

$$\{w \mid \textit{the automaton } A \textit{ accepts } w\}.$$

Consider a language $L \subseteq \Sigma^*$. We define an equivalence relation $\equiv_L$ for pair of words $u, w \in \Sigma^*$. We say that $u$ and $v$ are $\equiv_L$-equivalent, written $u \equiv_L w$, if $u \cdot z \in L \iff w \cdot z \in L$ for all $z \in \Sigma^*$. The well known Myhill-Nerode Theorem states (Nerode 1958) (Hopcroft & Ullman 1979):

**Theorem (Myhill-Nerode)** *For a regular language $L$, the number of equivalences classes of $\equiv_L$ is equal to the number of states of the minimal DFA accepting $L$.*

We will be using this theorem in our proofs to follow.

## 3 Regular Languages with Linear State Complexity upon determinization

Let $\Sigma = \{0, 1, \ldots, k-1\}$ be an alphabet of $k$ symbols, we define the following language:

$$L_{k,m}=\{ux \mid \; x \in \sigma^+, \sigma \in \Sigma,\; u \in \Sigma^*, \text{ and } |u| \equiv m-1(mod\ m)\}.$$

The following NFA $\mathcal{A}_{k,m} = \langle S, \Sigma, \delta, s_I, F\rangle$ accepts $L_{k,m}$ and has $m + k$ states.

1. $S = \{s_0, s_1, \ldots, s_{m+k-1}\}$ and $\Sigma = \{0, 1, \ldots, k-1\}$.
2. $s_I = \{s_0\}$ and $F = \{s_m, s_{m+2}, ..., s_{m+k-1}\}$.
3. $\delta(s_i, \sigma) = \begin{cases} \{s_{i+1}\} & \text{if } \; i < m-1,\; \sigma \in \Sigma \\ \{s_0, s_{m+\sigma} + 1\} & \text{if } \; i = m-1,\; \sigma \in \Sigma \\ \{s_i\} & \text{if } \; i \geq m,\; \sigma = i - m - 1 \end{cases}$

At state $s_{m-1}$ the automaton $\mathcal{A}_{k,m}$ nondeterministically guesses the form of the remaining word to be either $\sigma^+$ or $u\sigma^+$ ($|u| \equiv m-1(mod\, m)$), where $\sigma \in \Sigma$. The nondeterministic automaton $\mathcal{A}_{2,4}$ is shown in Figure 1. It is not hard to see that $\mathcal{A}_{k,m}$ has $n + k$ transitions.

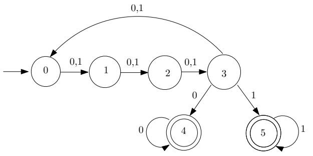


Figure 1: The nondeterministic automaton $\mathcal{A}_{2,4}$

**Lemma 1.** *NFA $\mathcal{A}_{k,m}$ with $m+k$ states is a minimal NFA accepting $L_{k,m}$.*

*Proof.* Let NFA $\mathcal{C}= \langle S', \Sigma, \delta', s'_I, F'\rangle$ be a minimal NFA accepting $L_{k,m}$. For a word $w$ where $w = (10)^*(0)$, $|w| = m-1$ and a symbol $\sigma \in \Sigma$, the word $w \cdot \sigma \cdot \sigma \in L_{k,m}$, and there is an accepting run for $\mathcal{C}$ in $w \cdot \sigma \cdot \sigma$. Let $s(w \cdot \sigma) = \{s \in S' \mid s \in \delta'^+(s_0, w \cdot \sigma) \wedge \delta'^+(s, \sigma) \cap F' \neq \emptyset\}$. Assume for the sake of contradiction that $s(w\cdot\sigma) \cap s(w\cdot\alpha) \neq \emptyset$ where $\sigma \neq \alpha$ and $\sigma, \alpha \in \Sigma$. Then the word $w \cdot \sigma \cdot \alpha \notin L_{k,m}$ will be accepted, and we have reached a contradiction. Hence there are $k$ symbols in the alphabet $\sigma$, NFA $\mathcal{C}$ has at least $k$ states.

Let $p_0, p_1, \ldots, p_{m+1}$ be tha accepting run of $\mathcal{A}_{k,m}$ on $w \cdot 0$ where $w = (10)^*(0)$ and $|w| = m-1$. Since no word of length less than $m$ is in the language, the state $p_m$ is an accepting state and states $p_0, p_1, \ldots, p_{m-1}$ are non-accepting states. If there exist $p_i$ and $p_j$ with $i < j \leq m$ such that $p_i = p_j$. Then a cycle of length $i$ where $i < m$ exists, without running through the cycle, a word $u$ where $|u| < m$ is accepted. However no word of length less than $m$ is in $L_{k,m}$. Hence we have reached a contradiction and $p_0, p_1, \ldots, p_m$ are all distinct states. If there exists $p_i$ $(i < m)$ such that $p_i \in s(w \cdot \alpha)$ for some $\alpha \neq 0$. Then the word $w \cdot \alpha \cdot u$ where $|u| < m$ will be accepted. However $w \cdot \alpha \cdot u$ is not in $L_{k,m}$. Hence the states $\{p_0, p_1, \ldots, p_{m-1}\} \cap s(w \cdot \alpha) = \emptyset$ for $\alpha \in \Sigma$ and $\mathcal{C}$ has another $m$ states. Overall, $\mathcal{C}$ has at least $m + k$ states. □

The following DFA $\mathcal{B}_{k,m} = \langle S', \Sigma, \delta', s'_I, F'\rangle$ accepts $L_{k,m}$ with $(k+1)m + (1-k)$ states. Intuitively, the automaton by reading an input string $w$ counts the lengths of the prefixes of $w$ modulo $m$, and once the length equals $m-1$ modulo $m$ the automaton starts verifying that the rest of the string is from $\sigma^+$ for some $\sigma \in \Sigma$.

1. $S = \{s'_0, s'_1, \ldots, s'_{k-1}\} \cup \{s'_{0,1}, \ldots, s'_{0,k-1}\} \cup \ldots \cup \{s'_{k-1,1}, \ldots, s'_{k-1,k-1}\} \cup \{s'_F\}$.
2. $s'_I = s'_0$
3. $F' = \{s'_F\} \cup \{s'_{i,j} \mid i \leq k-1 \textit{ and } 1 \leq j \leq k-1\}$.
4. For $\sigma \in \Sigma$, we have the following transition functions:
$\delta'(s'_i, \sigma) = \begin{cases} s'_{i+1} & \text{if } i < m-1 \\ s'_{\sigma,1} & \text{if } i = m-1 \end{cases}$
$\delta'(s'_{i,j}, \sigma) = \begin{cases} s'_{i,j+1} & \text{if } \sigma = i \text{ and } 1 \leq j < k-1 \\ s'_j & \text{if } \sigma \neq i \text{ and } i \leq j < k-1 \\ s'_F & \text{if } j = k-1 \end{cases}$

The automaton $\mathcal{B}_{2,4}$ is shown in Figure 2.

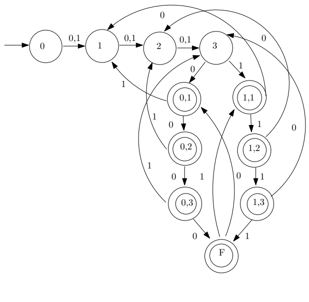


Figure 2: The deterministic automaton $\mathcal{B}_{2,4}$

**Lemma 2.** *The minimal DFA recognizing $L_{k,m}$ has exactly $(k+1)m + (1-k)$ states.*

*Proof.* For the proof we use Myhill-Nerode theorem and count the number of $\equiv_{L_{k,m}}$ equivalence classes. First consider a word $x \in \Sigma^*$ where $|x| \geq m$, we can write it in the form $x = u \cdot w$ where $u, w \in \Sigma^*$ and $|u| \equiv m-1(mod\ m)$ and $1 \leq |w| \leq m$. There are two cases for the word $w$:

*Case 1*: $w \in \sigma^i$ where $\sigma \in \Sigma$ and $1 \leq i \leq m$. For this case we want to show that the number of $\equiv_{L_{k,m}}$ equivalence classes is $k \cdot (m-1) + 1$. To show this we distinguish the following two possibilities for $w = \sigma^i$:

1. $|w| < m$: Consider any other word $x'$ such that $x'$ is of the form $u' \cdot w'$ where $|u'| = m-1$ modulo $m$ and $w'$ is of the form $\alpha^j$ with $\alpha \in \Sigma$ and $1 \leq j \leq$

$m$. Then either $|w| = |w'|$ or $|w| \neq |w'|$. First we consider the case when $|w| = |w'|$. In this case it must be that $\sigma \neq \alpha$. It is not hard to see that for $z = \sigma^{m-|w|}$ we have $x' \cdot z \in L_{k,m}$. However, $x' \cdot z \notin L_{k,m}$ because $|\alpha^i \sigma^{m-i}| = m$ and $\alpha \neq \sigma$. Hence, $x \not\equiv_{L_{k,m}} x'$. Now we consider the case when $|w| \neq |w'|$. Without loss of generality, we may assume $|w| > |w'|$. Next consider $z = \beta^{m-|w|+1}$, where $\beta \in \Sigma$ with $\beta \neq \alpha$. For this $z$ we have $x \cdot z \in L_{k,m}$ because it is of the form $u\sigma^i\beta^{m-i}\beta$ and $|u\sigma^i\beta^{m-i}| = m - 1$ modulo $m$. However, $x' \cdot z \notin L_{k,m}$. Thus, this possibility proves that there are exactly $k \cdot (m - 1)$ number of $\equiv_{L_{k,m}}$ equivalence classes represented by the words of the form $x = u \cdot w$ where $|u| \equiv m - 1 (mod\ m)$ and $w = \sigma^i$ with $\sigma \in \Sigma$ and $1 \leq i \leq m$.

2. $|w| = m$: Consider any word $x'$ of the form $x' = u' \cdot w'$, where $|u'| = m - 1$ modulo $m$. It is not hard to see that $x \equiv_{L_{k,m}} x'$ since for all $z \in \Sigma^*$ we have $x \cdot z \in L_{k,m} \iff x' \in L_{k,m}$. Now we want to show that $x$ is not $\equiv_{L_{k,m}}$ equivalent to any word $y$ of the form $y = u_0 \cdot \alpha^i$ where $\alpha \in \Sigma$, $1 \leq i < m$ and $|u_0| \equiv m - 1$ $(mod\ m)$. Take $\beta \in \Sigma$ such that $\beta \neq \alpha$. Then it is clear that $x \cdot \beta \in L_{k,m}$, but $y \cdot \beta \notin L_{k,m}$.

Thus, *Case 1* proves that there are $k \cdot (m-1) + 1$ equivalence $\equiv_{L_{k,m}}$-classes.

*Case 2:* Assume that $w$ is not of the form $\sigma^i$ for $\sigma \in \Sigma$ and $1 \leq i \leq m - 1$). We want to show that there are $m$ number of $\equiv_{L_{k,m}}$ equivalence classes all distinct from the equivalence classes provided in *Case 1*.

Consider a word $x'$ of the form $x' = u' \cdot w'$, where $u'$ and $w'$ are components of $x'$ and satisfy the same conditions as the $u$ and $w$ components of $x$. Then either $|w| = |w'|$ or $|w| \neq |w'|$. First we consider the case $|w| = |w'|$. It is not hard to see that $x \equiv_{L_{k,m}} x'$ since for all $z \in \Sigma^*$ that $x \cdot z \in L_{k,m} \iff x' \cdot z \in L_{k,m}$. This is due to the choices of $u$, $u'$, $w$ and $w'$. Next we consider the case $|w| \neq |w'|$ and assume that $|w'| < |w|$. For $z = 0^{m-|w|+1}$, we have $x \cdot z \in L_{k,m}$ and $x' \cdot z \notin L_{k,m}$. Therefore $x \not\equiv_{L_{k,m}} x'$.

Now we need to show that $x$ is not $\equiv_{L_{k,m}}$ to any word from *Case 1*. Consider $y = u_0 \cdot \sigma^i$ where $u_0 \in \Sigma^*$, $1 \leq i \leq m$ and $|u_0| \equiv m - 1 (mod\ m)$. Take $z = \sigma^{m-|x|-1}$, it is not hard to see that $y \cdot z \in L_{k,m}$ but $x \cdot z \notin L_{k,m}$. Hence $y \not\equiv_{L_{k,m}} x$, and in this case $L_{k,m}$ has $m - 1$ distinct equivalence classes.

Next we consider a word $x \in \Sigma^*$ where $|x| < m$, there are two cases:

*Case 1:* We consider all $1 \leq |x| < m$. Then it is not hard to see that for all $z \in \Sigma^*$ that $x \cdot z \in L_{k,m} \iff 0^{m-1} \cdot x \cdot \in L_{k,m}$. Therefore $x \equiv_{L_{k,m}} 0^{m-1} \cdot x$ and we have already counted the equivalence classes.

*Case 2:* We consider the case when $x = \lambda$. Consider a word $x' \in \Sigma^* \backslash \{\lambda\}$. If $x' \in L_{k,m}$, we set $z = \lambda$. It is clear that $x \cdot z \notin L_{k,m}$, but $x' \cdot z \in L_{k,m}$. Therefore $x \not\equiv_{L_{k,m}} x'$. If $x' \notin L_{k,m}$, then let $i = |x| (mod\ m)$ such that $0 \leq i < m$. Now set $z = 0^{m-i+1}$. It is clear that $x \cdot z \notin L_{k,m}$ but $x' \cdot z \cdot \in L_{k,m}$ and thus $x \not\equiv_{L_{k,m}} x'$. Therefore $\lambda$ is an equivalence class on its own.

We have shown that $L_{k,m}$ has $(k + 1)m + (1 - k)$ equivalence classes. Therefore, by Myhill Nerode theorem the minimal DFA accepting $L_{k,m}$ has exactly $(k + 1)m + (1 - k)$ states. □

We now reformulate our results above in terms of linear blow-up of the determination process of nondeterministic finite automata.

**Theorem 1.** *[Linear Blow-Up Theorem 1] For every $k > 1$ there exists a regular language $L_n$ over a $k$-letter alphabet, where $n > k$, such that a minimal NFA recognizing $L_n$ needs exactly $n$ states and the minimal DFA recognizing $L_n$ needs exactly $(k+1) \cdot n - c$ states, where $c = (k+1)^2 - 2$. Moreover, the minimal NFA recognizing $L_n$ needs $O(n)$ transitions.*

*Proof.* The language $L_n$ is $L_{k,m}$ where $n = k + m$. Lemma 1 shows that this language requires exactly $n$ states to be recognized by a minimal NFA. Theorem 2 shows that this language requires exactly $(k+1) \cdot n - c$ states to be recognized by a minimal DFA. From the definition of the NFA recognizing $L_{k,m}$, it is not hard to see that it has exactly $n + k$ states. □

One would like to sharpen the theorem above to build a regular language $L_n$ such that the minimal NFA recognizing $L_n$ has exactly $n$ states and the minimal DFA recognizing $L_n$ has exactly $k \cdot n$ states. Below we present another class of languages in which this sharpness can be achieved for infinitely many $n$.

Let $\Sigma = \{0, 1\}$ and $k, m \in \mathbb{N}^+$. We define the following language

$$U_{k,m} = \{u \cdot 0 \cdot w \mid u, w \in \Sigma^*,\ |u| \geq m \ and\ |w| = k\}.$$

Intuitively, $U_{k,m}$ is the set of all words $v$ such that $|v| \geq (m+k+1)$ and the $k+1$th letter from the right is 0.

The following NFA $\mathcal{A}_{k,m} = \langle S, \Sigma, \delta, s_I, F \rangle$ accepts $U_{k,m}$ with $m + k + 2$ states.

1. $S = \{s_0, s_1, ..., s_{m+k+1}\}$, $\Sigma = \{0, 1\}$.
2. $s_I = s_0$, $F = \{s_{m+k+1}\}$.
3. $\delta(s_i, \sigma) = \begin{cases} s_{i+1} & \text{if } 0 \leq i < m \text{ or} \\ & m < i \leq m + k \text{ and } \sigma \in \Sigma \\ s_i & \text{if } i = m,\ \sigma \in \Sigma \\ s_{i+1} & \text{if } i = m \text{ and } \sigma = 0 \end{cases}$

Intuitively, the automaton $\mathcal{A}_{k,m}$, after processing the prefix of an input word of length greater than $m$, nondeterministically guesses that the rest of the string has length $k$ once a 0 is read. Then the automaton verifies that the guess was correct.

The nondeterministic automaton $\mathcal{A}_{3,6}$ which has 11 states is shown in Figure 3. It is not hard to see that $\mathcal{A}_{k,m}$ has $m + k + 2$ transitions.

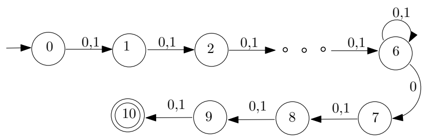


Figure 3: NFA $\mathcal{A}_{3,6}$ with 11 states

**Lemma 3.** *A minimal NFA accepting $U_{k,m}$ has exactly $m + k + 2$ states.*

*Proof.* Assume for a contradiction that there exists an NFA $\mathcal{C} = \langle S', \Sigma, \delta', s'_I, F' \rangle$ accepting $U_{k,m}$ with at most $m + k + 1$ states. Consider $0^{m+1}w \in \Sigma^*$ where $|w| = k$. Let $r = p_0, p_1, ..., p_{m+k+1}$ be an accepting run of $\mathcal{C}$ on $0^{m+1}w$. There are $m+k+2$ states in this run and hence there is at least one state $p$ appearing twice in $r$. Thus, a cycle of length smaller than $m + k + 1$ exists. Let string $v_0, v_1 \in \Sigma^*$ be such that

$p \in \delta'(s'_0, v_0) \cap \delta'(p_i, v_1)$, and string $v_2$ be such that $\delta(p_i, v_2) \cap F' \neq \emptyset$. Therefore $\delta'(s'_0, v_0 v_2) \cap F' \neq \emptyset$. However, $|v_0 v_2| < m+k+1$ and therefore $v_0 v_2 \notin U_{k,m}$. Hence we have reached a contradiction. □

Next we show the minimal number of states a DFA requires to accept $U_{k,m}$ is $2^{k+1} + m$. Intuitively, the deterministic automaton needs to remember the first $m$ states of the NFA $\mathcal{A}_{k,m}$. Afterwards, once 0 is read, the DFA needs to remember all the strings of length at most $k$. This is formally proven in the lemma:

**Lemma 4.** *The minimal DFA recognizing $U_{k,m}$ has $2^{k+1} + m$ states.*

*Proof.* For the proof we use Myhill-Nerode Theorem and count the number of $\equiv_{U_{k,m}}$ equivalence classes. Consider a word $x \in \Sigma^*$, there are three cases:

*Case 1:* $|x| \leq m$: Consider any other word $y \in \Sigma^*$ where $|y| \leq m$. There are two possibilities, either $|x| = |y|$ or $|x| \neq |y|$. First we consider the case $|x| \neq |y|$. Without loss of generality we may assume $|x| > |y|$. The word $x \cdot 1^{m-|x|} \cdot 0^{k+1} \in U_{k,m}$ and $y \cdot 1^{m-|x|} \cdot 0^{k+1} \notin U_{k,m}$. Hence, corresponding to each $0 \leq i \leq m$ we have one distinct equivalence class giving us $m+1$ classes.
Now consider $|x| = |y|$. It is clear for $z \in \Sigma^*$ that $x \cdot z \in U_{k,m} \iff y \cdot z \in U_{k,m}$. Thus, $x \equiv_{U_{k,m}} y$ and we have already counted the equivalence classes. There is a total of $m+1$ equivalence classes in this case.

*Case 2:* $m+1 \leq |x| \leq m+k+1$: In this case either $x = u \cdot 0 \cdot w$ or $x = u \cdot 1 \cdot w$ where $u, w \in \Sigma^*$ and $|u| = m$. First we consider $x = u \cdot 0 \cdot w$. Take any other word $y = u' \cdot 0 \cdot w'$ where $u', w' \in \Sigma^*$ *and* $|u'| = m$. If $w \neq w'$, then let $w_1$ be the suffix such that $w = w_0 \cdot \sigma \cdot w_1$ and $w' = w'_0 \cdot \sigma' \cdot w_1$ ($\sigma, \sigma' \in \Sigma$ and $\sigma \neq \sigma'$). Without loss of generality we may assume $\sigma = 0$ and $\sigma' = 1$. It is clear that $x \cdot 1^{k-i+1} \in U_{k,m}$ and $y \cdot 1^{k-i+1} \notin U_{k,m}$, and hence $x \not\equiv_{U_{k,m}} y$. Therefore $U_{k,m}$ has another $2^0 + 2^1 + \ldots + 2^k = 2^{k+1} - 1$ equivalence classes. When $w = w'$ for all $z \in \Sigma^*$ it is clear that $x \cdot z \iff y \cdot z$ and we have already counted the equivalence classes.

Next we consider $x$ of the form $u \cdot 1 \cdot w$ ($u, w \in \Sigma^*$ and $|u| = m$). If $w \in 1^*$, then it is clear that $x \equiv_{U_{k,m}} u$. Otherwise, let $w_1$ be such that $w = 1^* \cdot 0 \cdot w_1$. Then $x \equiv_{U_{k,m}} 0^{m+1} \cdot w_1$. In both cases we have already counted the equivalence classes. Thus $U_{k,m}$ has another $2^{k+1} - 1$ equivalence classes in Case 2.

*Case 3:* $|x| > m + k + 1$: We first consider $x \in U_{k,m}$. Then $x = u \cdot 0 \cdot w$ where $u, w \in \Sigma^*$ and $|w| = k$. It is clear that $x \equiv_{U_{k,m}} 0^{m+1} \cdot w$. Now we consider $x \notin U_{k,m}$, then $x = u \cdot 1 \cdot w$ where $u, w \in \Sigma^*$ and $|w| = k$. If $w \in 1^*$ then $x \equiv_{U_{k,m}} u$, else $w$ can be written as $1^* \cdot 0 \cdot w_1$ where $w_1 \in \Sigma^*$. It is clear that $x \equiv_{U_{k,m}} 0^{m+1} \cdot w_1$. In this case, we have already counted the equivalence classes.

From the above arguments, we have shown that $U_{k,m}$ has $2^{k+1} + m$ equivalence classes. Hence, by the Myhill-Nerode theorem, the minimal DFA accepting $U_{k,m}$ has $2^{k+1} + m$ states. □

Let $p$ be a natural number. We fix $m = \frac{2^{k+1} - pk - 2p}{p-1}$ and assume that $m$ is also a natural number. For instance, when $p = 2$ we have $m = 2^{k+1} - 2k - 4$. For such chosen $m$ and $p$ we have the following theorem that sharpens Theorem 3.

**Theorem 2** (Linear Blow-Up Theorem 2)**.** *For every $n = k + m$ there exists a regular language $L_n$ over the binary alphabet such that the minimal NFA recognizing $L_n$ needs exactly $n$ states and the minimal DFA needs exactly $p \cdot n$ states. The minimal NFA recognizing $L_n$ has $O(n)$ transitions.*

*Proof.* The desired language $L_n$ is $U_{k,m}$. We have shown in Lemma 3 that $n = m + k + 2$. Furthermore, we have shown in Theorem 4 that the minimal DFA accepting $U_{k,m}$ needs $m + 2^{k+1}$ states. Since $m = \frac{2^{k+1} - pk - 2p}{p-1}$, the minimal DFA accepting $U_{k,m}$ needs $p(m+k+2)$ states. From the definition of the NFA for $U_{k,m}$, it is not hard to see that it has $O(n)$ transitions. □

## 4 Regular Languages with Polynomial State Complexity upon determinization

Let $\Sigma = \{0, 1, \ldots, (k-1)\}$ be an alphabet of $k$ symbols. For $m \in \mathbb{N}^+$, we define the following languages:

$$V_k = 0^* 1^* \ldots (k-1)^*.$$

$$V_{k,m} = \{u \mid u \in V_k \textit{ and } |u| = m\}.$$

$$R_{k,m} = \{u \cdot 0 \cdot w \mid u \in V_k \textit{ and } w \in V_{k,m}\}.$$

**Lemma 5.** *The number of states sufficient for a NFA accepting $R_{k,m}$ is $km + 2$.*

*Proof.* The following NFA $\mathcal{A}_{k,m}$ accepts $R_{k,m}$ with $km + 2$ states. Let $\mathcal{A}_{k,m} = \langle S, \Sigma, \delta, s_I, F \rangle$ be such that:

1. $S = \{s_0, s_1, \ldots, s_{k-1}\} \cup s_k \cup \{s_{0,1}, \ldots, s_{0,m-1}\} \cup \ldots \cup \{s_{k-1,1}, \ldots, s_{k-1,m-1}\} \cup \{s_F\}$.
2. $s_I = s_0$ and $F = \{s_F\}$.
3. For $\sigma \in \Sigma$ and $s \in S$, the transition function is:

$$\delta(s_i, \sigma) = \begin{cases} \{s_\sigma, s_k\} & \text{if } i \leq k-1 \text{ and } \sigma \geq i \\ \{s_{\sigma,1}\} & \text{if } i = k \end{cases}$$

$$\delta(s_{i,j}, \sigma) = \begin{cases} \{s_{i,j+1}\} & \text{if } j < m-1 \text{ and } i = \sigma \\ \{s_{\sigma,j+1}\} & \text{if } j < m-1 \text{ and } i < \sigma \\ \{s_F\} & \text{if } j = m-1 \text{ and } i \leq \sigma \end{cases}$$

□

**Lemma 6.** *For a fixed $k$, the NFA recognizing $R_{k,m}$ has $O(n)$ transitions, where $n = km + 2$.*

*Proof.* Let NFA $\mathcal{A}_{k,m} = \langle S, \Sigma, \delta, s_I, F \rangle$ be the NFA recognizing $R_{k,m}$ as defined in Lemma 5. For a state $s \in S$, let $t(s)$ be the number of states $s$ has transitions to. First we consider the case $t(s_i)$ where $i \leq k$. For state $s_i$ where $i \leq k-1$, state $s_i$ has transitions to states $s_i, s_{i+1}, \ldots, s_k$ and hence $t(s_i) = k + 1 - i$. Therefore the total number of transitions $s_0, s_1, \ldots, s_{k+1}$ have are

$$\sum_{i=0}^{k-1} t(s_i) = (k+1) + k + \ldots + 2 = \frac{k(k+3)}{2}$$

Furthermore, it is not hard to see that $t(s_k) = k$. Hence we have accounted for $\frac{k(k+3)}{2} + k$ transitions.

Now we consider a state $s_{i,j}$ where $i \leq k-1$ and $j \leq m-1$. The state $s_{i,j}$ has transitions to states $s_{i,j+1}, \ldots, s_{k-1,j+1}$ and hence $t(s_i, j) = k - i$. Hence we have

$$\sum_{i=0}^{k-1} \sum_{j=1}^{m-1} t(s_{i,j}) = (m-2)(1+\ldots+k) = \frac{(k+1)k(m-1)}{2}$$

Note that state $s_F$ has no transitions.

Hence in total $\mathcal{A}_{k,m}$ has $\frac{k+1}{2}km - c$ transitions, where $c = \frac{k(k+1)}{2}$. Therefore the $\mathcal{A}_{k,m}$ recognizing $R_{k,m}$ has $O(n)$ states.

□

The nondeterministic automaton $\mathcal{A}_{2,3}$ is shown in Figure 4.

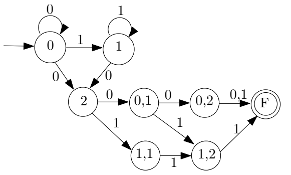


Figure 4: NFA $\mathcal{A}_{2,3}$ with 8 states

**Lemma 7.** *For the language $V_{k,m}$, where $1 \leq m$ and $k$ is the size of the alphabet, the cardinality of $V_{k,m}$ is $\prod_{i=1}^{k-1} \frac{(m+i)}{i}$.*

*Proof.* We show this by an induction on $k$. For the base case $k = 2$, it is not hard to see that

$$|V_{2,m}| = (m+1) = \textstyle\prod_{i=1}^{2-1} \frac{(m+i)}{i}.$$

Assume it is true for $k = n - 1$ that

$$|V_{n-1,m}| = \textstyle\prod_{i=1}^{n-2} \frac{(m+i)}{i}.$$

Words in $V_{n,m}$ are of the form $u_{n-1,m-i} \cdot n^i$ where $0 \leq i \leq m$ and $u_{n-i} \in V_{n-1,m-i}$. Thus, the cardinality of $V_{n,m}$ is:

$$|V_{n,m}| = |V_{n-1,m}| + |V_{n-1,m-1}| + ... + |V_{n-1,0}| = \textstyle\prod_{i=1}^{n-2} \frac{(m+i)}{i} + \ldots + \prod_{i=1}^{n-2} \frac{(0+i)}{i} = \prod_{i=1}^{n-1} \frac{(m+i)}{i}.$$

□

**Lemma 8.** *For the language $V_{k,m}$, where $1 \leq m$ and $k$ is the size of the alphabet*

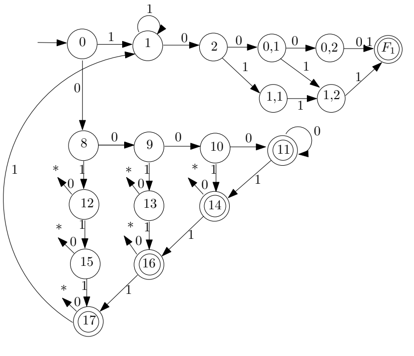
$$\textstyle\sum_{i=0}^{m} |V_{k,i}| = \prod_{i=1}^{k} \frac{(m+i)}{i}.$$

*Proof.* We have previously shown in Lemma 7 that $|V_{k,m}| = \prod_{i=1}^{k-1} \frac{(m+i)}{i}$. Hence we have the following relation: $\sum_{i=0}^{m} |V_{k,i}| = |V_{k,0}| + |V_{k,1}| + ... + |V_{k,m}| = \prod_{i=1}^{k-1} \frac{(i+0)}{i} + \ldots + \prod_{i=1}^{k-1} \frac{(i+m)}{i} = \prod_{i=1}^{k} \frac{m+i}{i}$. □

**Lemma 9.** *The minimal DFA accepting $R_{k,m}$ needs $\prod_{i=1}^{k} \frac{m+i}{i} + (km+3)$ states.*

*Proof.* We count the number of distinct $\equiv_{R_{k,m}}$-equivalence classes. $R_{k,m}$ is represented by the regular expression $0^* \cdot 1^* \cdot \ldots \cdot k^* \cdot 0 \cdot u$ where $u \in V_{k,m}$. Consider a word $x \in \Sigma^*$, there are two cases:

*Case 1*: $\exists z \in \Sigma^*$ such that $x \cdot z \in R_{k,m}$. There are the following sub-cases:



All transitions marked $*$ go to State 2.

Figure 5: Determinstic Automaton $\mathcal{B}_{2,3}$

1. $x \in 0 \cdot u$ such that $u \in V_{k,i}$ $(0 \leq i \leq m)$: In this case, corresponding to every $u \in V_{k,i}$ we have a distinct $\equiv_{R_{k,m}}$-equivalence class containing the word $0 \cdot u$. Consider distinct words $0 \cdot u$ and $0 \cdot u'$, where $u \in V_{k,n}$ and $u' \in V_{k,j}$ $(0 \leq n, j \leq m)$. Without loss of generality we may assume that $n \leq j$.

   First, consider the case when $n < j$. If $j = m$, then $0 \cdot u' \in R_{k,m}$ but $0 \cdot u \notin R_{k,m}$ and thus $0 \cdot u \not\equiv_{R_{k,m}} 0 \cdot u'$. If $j \neq m$ then let $\sigma \in \Sigma$ be the last symbol that occurs in $u'$. Then $0 \cdot u' \cdot \sigma^{m-j} \in R_{k,m}$ but $0 \cdot u \cdot \sigma^{m-j} \notin R_{k,m}$ and hence $0 \cdot u \not\equiv_{R_{k,m}} 0 \cdot u'$.

   Next consider the case when $n = j$. Let $u = 0^+ \cdot v$ and $u' = 0^+ \cdot v'$ where $v, v' \in (\Sigma \backslash \{0\})^*$. Since $u \neq u'$, we have $|v| \neq |v'|$. Without loss of generality, assume that $|v| > |v'|$. Then $0 \cdot u' \cdot \sigma^{m-|v'|} \in R_{k,m}$ but $0 \cdot u \cdot \sigma^{m-|v'|} \notin R_{k,m}$ where $\sigma$ is the last symbol of $v'$. Thus $0 \cdot u \not\equiv_{R_{2,m}} 0 \cdot u'$.

   By Lemma 8, we have $\sum_{i=0}^{m} |V_{k,i}| = \prod_{i=1}^{k} \frac{(m+i)}{i}$ and therefore we have $\prod_{i=1}^{k} \frac{(m+i)}{i}$ distinct equivalence classes corresponding to each word of the form $0 \cdot u$.

2. $x \in v \cdot 0 \cdot u$ where $v \in (1^* \cdot \ldots \cdot k^*) \backslash \{\lambda\}$ and $u \in V_{k,i}$ $(0 \leq i \leq m)$: Let $x = v \cdot 0 \cdot l^j$ and consider another word $w \in v \cdot 0 \cdot p^j$ where $l, p \in \Sigma \setminus \{0\}$ and $l < p$ $(1 \leq j \leq m-1)$. Then $x \not\equiv_{R_{k,m}} w$ since $x \cdot l^{m-j} \in R_{k,m}$ but $w \cdot l^{m-j} \notin R_{k,m}$. Also $x, w \not\equiv_{R_{k,m}} 0 \cdot u'$ $(u' \in V_{k,i})$ since $0 \cdot u' \cdot 0 \cdot 0^m \in R_{k,m}$ but $x \cdot 0 \cdot 0^m \notin R_{k,m}$ (similarly $w \cdot 0 \cdot 0^m \notin R_{k,m}$). Hence, corresponding to each $1 \leq j \leq m-1$ we have $k$ distinct equivalence classes giving us $k \cdot (m-1)$ equivalence classes. We further have one equivalence class for all words of the form $v \cdot 0$. For $u \in V_{k,m}$, all $x \in v \cdot 0 \cdot u$ form another equivalence class.

   Note that $x \in v \cdot 0 \cdot u \cdot \sigma$ $(\sigma \in \Sigma$ and $u \in V_{k,i}$ for $0 \leq i \leq m-1)$ is equivalent to all words of the form $v \cdot 0 \cdot \sigma^{i+1}$ and we have already counted these equivalence classes.

   Thus, there are a total of $k \cdot (m-1) + 2$ distinct equivalence classes in this case.

3. $x \in 0^+ \cdot v$ where $v \in (1^* \cdot \ldots \cdot k^*) \setminus \{\lambda\}$: If $|x| \leq m+1$, then $x$ is of the form $0 \cdot u$ for $u \in V_{k,|x|-1}$ and we have already counted the equivalence class corresponding to $x$.

   If $|x| > m+1$, let $x = 0 \cdot 0^n \cdot v$ such that $n \geq 0$. If $|v| > m$, then it is easy to see that $x \equiv_{R_{k,m}} \sigma^{|v|}$

where $\sigma$ is the last symbol of $x$. Therefore let $|v| \leq m$. If $n + |v| \leq m$, then $x$ is of the form $0 \cdot u$ where $u = 0^n \cdot v \in V_{k,n+|v|}$. If $n + |v| > m$, then $x \equiv_{R_{k,m}} 0 \cdot 0^{m-|v|} \cdot v$. In either case we have already counted the equivalence classes corresponding to $x$. Hence, all $x \in 0^+ \cdot v$ belong to previously enumerated equivalence classes.

4. $x \in \sigma^+$ ($\sigma \in \Sigma \setminus \{0\}$): For every $\sigma \in \Sigma \setminus \{0\}$, all words of the form $\sigma^+$ form a distinct equivalence class. Consider words $l^i$ and $p^j$ such that $l, p \in \Sigma \setminus \{0\}$ and $l < p$ ($i, j \geq 1$). Then $l^i \cdot l \cdot 0^{m+1} \in R_{k,m}$ but $p^j \cdot l \cdot 0^{m+1} \notin R_{k,m}$ and $l^i \not\equiv_{R_{k,m}} p^j$. Hence we have $k-1$ equivalence classes in this case.

5. $x = \lambda$: $\lambda$ forms a distinct equivalence class.

*Case 2*: $\forall z \in \Sigma^*$ we have $x \cdot z \notin R_{2,m}$: All such $x$ form one distinct equivalence class.

From the above arguments, we can see that there are $\prod_{i=1}^{k} \frac{(m+i)}{i} + (k \cdot m + 3)$ distinct $\equiv_{R_{k,m}}$ equivalence classes. Thus, by the Myhill-Nerode theorem the minimal DFA accepting $R_{k,m}$ has exactly $\prod_{i=1}^{k} \frac{(m+i)}{i} + (k \cdot m + 3)$ states. □

The next two lemmas analyse the transition complexity of the language $R_{k,m}$ and show that the transition complexity of $R_{k,m}$ is asymptotically less than the number of transitions of the NFA of (Jirásek et al. 2007) in the worst case.

**Lemma 10.** *For $\alpha = 2^n - n + 1$, the $n$-state NFA $\mathcal{A}$ constructed in (Jirásek et al. 2007), such that the DFA complexity of $L(A)$ is $\alpha$, has $O(n^2)$ transitions where the alphabet is $\{a, b, c, d\}$.*

*Proof.* Let $\alpha = 2^n - n + 1$. In this case the states in NFA $\mathcal{A}$ constructed in (Jirásek et al. 2007) has the following transitions for symbol $d$: $\delta(1, d) = \{0, 2\}, \delta(2, d) = \{0, 2, 3\}, \ldots, \delta(n-3, d) = \{0, 2, 3, \ldots, n-2\}$ and $\delta(n-2, d)$=$\delta(n-1, d)$=$\delta(n, d) = \{0, 1, 2, 3, \ldots, n-1\}$.

It is not hard to see that $\mathcal{A}$ has $O(n^2)$ transitions for the symbol $d$. There are $O(n)$ number of transitions for the symbols $a, b, c$ and hence $\mathcal{A}$ has $O(n^2)$ transitions in total.

□

**Lemma 11.** *For $\alpha = 2^n - n + 1$, the $n$-state NFA $\mathcal{B}_{k,m}$ recognizing $R_{k,m}$ such that the DFA state complexity of $R_{k,m}$ is $O(\alpha)$ has $O(\frac{n^2}{log_2 n})$ transitions.*

*Proof.* For $k = log_n(2^n - (n-1))$, the DFA state complexity of $R_{k,m}$ is $O(\alpha)$ according to Lemma 9.

State $s_i$ where $i < k - 1$ has $k + 1 - i$ outgoing transitions and state $s_k$ has $k$ outgoing transitions. States of the form $s_{i,j}$ where $1 \leq i \leq k-1$ have $k - i$ outgoing transitions each. Hence $\mathcal{B}_{k,m}$ has $\frac{1}{2}m(k+1)(k+2)$ transitions. We have $n = km + 2$ as shown in Lemma 6 and hence $\mathcal{B}_{k,m}$ has $O(k \cdot n)$ transitions. Since $k = log_n(2^n - (n-1))$, we have $k = \frac{n}{log_2 n} - c$ ($c > 0$). Hence $\mathcal{B}_{k,m}$ has $O(\frac{n^2}{\log_2 n})$ transitions.

□

**Theorem 3** (Polynomial blow-up theorem)**.** *For every $k > 1$ there exists a regular language $L_n$ over a $k$-letter alphabet such that*

1. *The minimal NFA recognizing $L_n$ needs $n$ states, where $n > k$, and the minimal DFA recognizing $L_n$ has $O(n^k)$ states.*
2. *For $\alpha = 2^n - n + 1$, the minimal NFA recognizing $L_n$ and having a blowup of $O(\alpha)$ has $O(\frac{n^2}{log_2 n})$ transitions. This is asymptotically fewer than the $O(n^2)$ transitions required by the NFA with DFA-state complexity $\alpha$ that was described in (Jirásek et al. 2007) .*

*Proof.* As shown in Lemma 5 and Lemma 9, $km + 2$ states are sufficient for a NFA accepting $R_{k,m}$ and the minimal DFA accepting $R_{k,m}$ has $\prod_{i=0}^{k} \frac{m+i}{i} + (km+3)$ states. Since $n > k$, this implies the minimal DFA accepting $R_{k,m}$ has $O(n^k)$ states. Our desired language $L_n$ then is $R_{k,m}$ with $n = km + 2$. The second part of the theorem follows from Lemmas 10 and 11. □

## 5 Complementation

In this section, we investigate the complementation problem for NFA. The complementation operation for DFA is efficient and the DFA recognizing the complement of a $n$-state DFA has at most $n$ states. However for every $n \geq 1$ and the binary alphabet, there exists a $n$-state NFA such that the minimal NFA recognizing its complement needs $2^n$ states (Moore 1971). The authors of (Jirásková 2008) show that for every $n, m > 1$ with $\log n \leq m \leq 2^n$ there exists a $n$-state NFA such that the minimal NFA accepting its complement has $m$ states with a fixed five letter alphabet . However, in the worst case the number of transitions in the $n$-state NFA is $O(n^2)$.

For a fixed $k > 1$, we first show that for every $n > k$, there exists a $O(n)$ state NFA such that the minimal NFA for the complement has $k \cdot n + c$ states where $k$ is the alphabet size and $c$ is a constant. Next we show that for every $n, k \geq 2$, there exists a $O(n)$-state NFA such that the minimal NFA accepting its complement has between $O(n^{k-1})$ and $O(n^{2k})$ states where the alphabet is of size $k$. We would like to point out that the $n$-state NFA that we describe in this section have asymptotically fewer transitions than the NFA of (Jirásková 2008).

### 5.1 Linear blow-up

Let $\Sigma = \{0, 1, \ldots, k-1\}$ be an alphabet of $k$ symbols. Recall the language $L_{k,m}$ we defined in Section 3.

$$L_{k,m}=\{ux \mid \ x \in \sigma^+, \sigma \in \Sigma,\ u \in \Sigma^*, \text{ and } |u| \equiv m - 1 (mod\ m)\}.$$

We proved that the minimal DFA recognizing $L_{k,m}$ has exactly $(k+1)m + (1-k)$ states. Then it is clear that the DFA recognizing the complement of $L_{k,m}$ has at most $(k+1)m + (1-k)$ many states. Our goal is to show that a succinct representation of this language using NFA still needs exactly $(k+1)m + (1-k)$ many states.

**Lemma 12.** *A minimal NFA recognizing the complement of $L_{k,m}$ has at least $(k+1)m + (1-k)$ states.*

*Proof.* Let NFA $\mathcal{A}$=$\langle S, \Sigma, \delta, s_I, F\rangle$ be a minimal NFA accepting $L^c_{k,m}$, the complement of the language $L_{k,m}$. For $u, v \in \Sigma^*$, define $S(u, v) = \{s \in S \mid s \in \delta^+(s_I, u) \text{ and } \delta^+(s, v) \cap F \neq \emptyset\}$.

First, we show that $\mathcal{A}$ needs at least $k(m-1) + 1$ non-accepting states. Consider a word $0^{m-1} \cdot \sigma^i$ where $1 \leq i \leq m$ and $\sigma \in \Sigma$. There are two cases for $i$:

*Case 1*: $i < m$: Consider any other word $0^{m-1} \cdot \alpha^j$ where $1 \leq j \leq m-1$ and $\alpha \in \Sigma$. There are two possibilities for $i$ and $j$:

1. $i = j$: In this case $\alpha \neq \sigma$. It is easy to see that $0^{m-1} \cdot \sigma^i \cdot \alpha^{m-i} \in L^c_{k,m}$. Assume for a contradiction that $S(0^{m-1} \cdot \sigma^i, \alpha^{m-1}) \cap S(0^{m-1} \cdot \alpha^j, \alpha^{m-1}) \neq \emptyset$. Let $s \in S(0^{m-1} \cdot \sigma^i, \alpha^{m-1}) \cap S(0^{m-1} \cdot \alpha^j, \alpha^{m-1})$. Then $\delta^+(s_I, 0^{m-1} \cdot \alpha^j \cdot \alpha^{m-i}) \cap F \neq \emptyset$ and hence $0^{m-1} \cdot \alpha^j \cdot \alpha^{m-i}$ is accepted by $\mathcal{A}$. This is a contradiction since $0^{m-1} \cdot \alpha^j \cdot \alpha^{m-i} \notin L^c_{k,m}$.

2. $i \neq j$: Let $\beta \in \Sigma \setminus \{\alpha\}$. Clearly $0^{m-1} \cdot \alpha^j \cdot \beta^{m-i+1} \in L^c_{k,m}$. Assume for the sake of contradiction that $S(0^{m-1} \cdot \sigma^i, \beta^{m-i+1}) \cap S(0^{m-1} \cdot \alpha^j, \beta^{m-i+1}) \neq \emptyset$. Then there must be an $s \in S(0^{m-1} \cdot \sigma^i, \beta^{m-i+1}) \cap S(0^{m-1} \cdot \alpha^j, \beta^{m-i+1})$ and therefore $\delta^+(s_I, 0^{m-1} \cdot \sigma^i \cdot \beta^{m-i+1}) \cap F \neq \emptyset$. Hence $\mathcal{A}$ accepts the word $0^{m-1} \cdot \sigma^i \cdot \beta^{m-i+1}$. This is a contradiction since $0^{m-1} \cdot \sigma^i \cdot \beta^{m-i+1} \notin L^c_{k,m}$.

Since we have $k(m-1)$ words of type $0^{m-1} \cdot \sigma^i$, we have shown that $\mathcal{A}$ needs at least $k(m-1)$ distinct states.

*Case 2*: $i = m$: Let $\beta \in \Sigma \setminus \{\sigma\}$. Consider any other word $0^{m-1} \cdot \alpha^j$ where $1 \leq j \leq m-1$. Then clearly $0^{m-1} \cdot \alpha^j \cdot \beta \in L^c_{k,m}$. Assume for contradiction that $S(0^{m-1} \cdot \alpha^j, \beta) \cap S(0^{m-1} \cdot \sigma^m, \beta) \neq \emptyset$. Then there must be an $s \in S(0^{m-1} \cdot \alpha^j, \beta) \cap S(0^{m-1} \cdot \sigma^m, \beta)$ and hence $\delta^+(s_I, 0^{m-1} \cdot \sigma^m \cdot \beta) \cap F \neq \emptyset$. Therefore, $\mathcal{A}$ accepts $0^{m-1} \cdot \sigma^m \cdot \beta$ which is a contradiction since $0^{m-1} \cdot \sigma^m \cdot \beta \notin L^c_{k,m}$. Hence $\mathcal{A}$ has at least one more state.

From the two cases above, we conclude that $\mathcal{A}$ has at least $k(m-1)+1$ states. We would now like to show that $\mathcal{A}$ has at least $m$ more states.

Consider a word $0^i$ for $0 \leq i \leq m-1$. Consider another word $0^j$ for $0 \leq j \leq m-1$ such that $i \neq j$. Without loss of generality assume that $i < j$. Clearly $0^i \cdot 0^{m-1-j} \cdot 0 \in L^c_{k,m}$. Assume for the sake of contradiction that $S(0^i, 0^{m-1-j} \cdot 0) \cap S(0^j, 0^{m-1-j} \cdot 0) \neq \emptyset$. Then there exists a state $s \in S(0^i, 0^{m-1-j} \cdot 0) \cap S(0^j, 0^{m-1-j} \cdot 0)$ and hence $\delta^+(s_I, 0^j \cdot 0^{m-1-j} \cdot 0) \cap F \neq \emptyset$. Thus $\mathcal{A}$ accepts $0^j \cdot 0^{m-1-j} \cdot 0$ and this is a contradiction since $0^j \cdot 0^{m-1-j} \cdot 0 \notin L^c_{k,m}$.

Now consider a word $0^i$ $(0 \leq i \leq m-1)$ and another word $0^{m-1}\alpha^j$ $(1 \leq j \leq m-1$ and $\alpha \in \Sigma)$. Clearly we have $0^i \cdot \alpha^{m-i-1} \in L^c_{k,m}$. Assume for contradiction that $S(0^i, \alpha^{m-i-1}) \cap S(0^{m-1}\alpha^j, \alpha^{m-i-1}) \neq \emptyset$ and there is $s \in S(0^i, \alpha^{m-i-1}) \cap S(0^{m-1}\alpha^j, \alpha^{m-i-1})$. Hence $\delta^+(s_I, 0^{m-1} \cdot \alpha^j \cdot \alpha^{m-i-1}) \cap F \neq \emptyset$ and therefore $\mathcal{A}$ accepts $0^{m-1} \cdot \alpha^j \cdot \alpha^{m-i-1}$. This is a contradiction since $0^{m-1} \cdot \alpha^j \cdot \alpha^{m-i-1} \notin L^c_{k,m}$.

From the above arguments, we can conclude that $\mathcal{A}$ has at least $m$ more states as required. Hence we have shown that $\mathcal{A}$ has at least $k(m-1)+1+m = (k+1)m + (1-k)$ states. □

**Theorem 4** (Linear blow-up for complementation)**.** *For every $k > 1$ there exists a regular language $L_n$ over k-letter alphabet, where $n > k$, such that*

1. *The minimal NFA recognizing $L_n$ needs exactly $n$ states and the minimal NFA recognizing the complement of $L_n$ needs exactly $(k+1)n - c$ states, where $c = (k+1)^2 - 2$.*

2. *The minimal NFA recognizing $L_n$ needs $O(n)$ transitions.*

*Proof.* The language $L_n$ is $L_{k,m}$ where $n = k + m$. We have shown in Lemma 1 a minimal NFA accepting $L_n$ needs exactly $n$ states. In Theorem 3 we have shown the minimal DFA accepting $L_n$ needs exactly $(k+1)n - c$ states, hence the complement of $L_n$ can be accepted by a $(k+1)n - c$ states DFA. Furthermore in Lemma 12, we have shown that $(k+1)n - c$ states are necessary for a minimal NFA accepting the complement of $L_n$. Hence this proves $(k+1)n - c$ states are necessary and sufficient for a minimal NFA recognizing $L^c_n$. We have shown in Theorem 1 the minimal NFA accepting $L_n$ has $O(n)$ transitions. □

### 5.2 Polynomial Blow-up

First consider the language $V_{k,m} = \{u \mid u \in 0^* \cdot \ldots \cdot (k-1)^*$ and $|u| = m\}$. Then it is clear that the following NFA $\mathcal{A} = (S, \Sigma, \delta, s_I, F)$ with $k(m-1)+2$ states recognizes $V_{k,m}$:

1. $S = \{s_0\} \cup \{s_{0,1}, \ldots, s_{0,m-1}\} \cup \ldots \cup \{s_{k-1,1}, \ldots, s_{k-1,m-1}\} \cup \{s_F\}$ and $\Sigma = \{0, 1, \ldots, k-1\}$.

2. $s_I = s_0$ and $F = \{s_F\}$.

3. For $0 \leq i < k$ and $\sigma \in \Sigma$, $\delta(s_0, \sigma) = \{s_{\sigma,1}\}$.

4. $$\delta(s_{i,j}, \sigma) = \begin{cases} \{s_{\sigma,j+1}\} & \text{if } j < m-1 \text{ and } i \leq \sigma \\ \{s_F\} & \text{if } j = m-1 \text{ and } i \leq \sigma \end{cases}$$

The NFA recognizing $V_{2,4}$ is shown in Figure 6. It is not hard to see that the NFA for $V_{k,m}$ has $O(k^2 m)$ transitions.

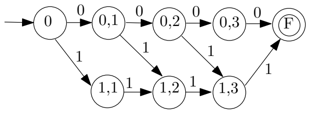


Figure 6: The NFA accepting $V_{2,4}$

Later we will need to use the following language: $G_{k,m,\alpha} = \{\beta \cdot u \mid \beta \cdot u \in V_{k,m}$ and $\beta \in \{0, \ldots \alpha - 1\}\}$ where $\alpha \in \{1, \ldots, k-1\}$. The following NFA $\mathcal{C}$ recognizes $G_{k,m,\alpha}$:

1. $S = \{s_{0,0}, \ldots, s_{0,m}\} \cup \ldots \cup \{s_{\alpha-1,0}, \ldots, s_{\alpha,m}\} \cup \{s_{\alpha,1}, \ldots, s_{\alpha,m}\} \ldots \cup \{s_{k-1,1}, \ldots, s_{k-1,m}\}$.

2. The initial states are $\{s_{\sigma,0} \mid \sigma \in \Sigma\}$ and $F = \{s_{\sigma,m} \mid \sigma \in \Sigma\}$.

3. For $i, \sigma \in \Sigma$ and $0 \leq j < m$:
$$\delta(s_{i,j}, \sigma) = \{s_{\sigma,j+1}\} \ \ \textit{where} \ \ i \leq \sigma$$

The NFA recognizing $G_{2,2,1}$ is shown in Figure 7. It is not hard to see that the NFA for $G_{k,m,\alpha}$ has $\alpha(m+1) + (k-\alpha)m$ states and $O(k^2 m)$ transitions.

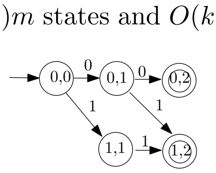


Figure 7: The NFA accepting $G_{2,2,1}$

**Lemma 13.** *For every $k, m > 1$, there exists a $O(m)$-state NFA $\mathcal{B}$ such that NFA accepting the complement of $L(\mathcal{B})$ has at least $O(m^{k-1})$ states.*

*Proof.* Consider the language $H_{k,m} = (\Sigma^* \cdot 0 \cdot y \cdot (\Sigma \setminus \{0\}) \cdot \Sigma^*) + (\Sigma^* \cdot 1 \cdot y \cdot (\Sigma \setminus \{1\}) \cdot \Sigma^*) + \ldots + (\Sigma^* \cdot (k-1) \cdot y \cdot (\Sigma \setminus \{k-1\}) \cdot \Sigma^*)$ where the following conditions hold:

1. $|y| = m$ and
2. $y = y_1 \cdot y_2$ such that $a \cdot y_1, y_2 \cdot b \in 0^* \cdot 1^* \cdot \ldots \cdot (k-1)^*$ for some symbols $a \neq b$.

Intuitively, the NFA recognizing $H_{k,m}$ behaves as follows: It guesses the position of a symbol $a \in \Sigma$ and then starts verifying whether the next $m+1$ symbols are in $V_{k,m}$. If at position $i$ in this verification, the automaton reads a symbol $\alpha$ such that the symbol at position $i-1$ is $\beta > \alpha$ then the automaton tries to verify whether the last $m-i+1$ symbols are in $V_{k,m-i+1}$ and the $(m+1)$th symbol is $b \neq a$.

Formally, let the following be the NFA's recognizing $V_{k,m}, \ldots, V_{k-i,m}, \ldots, V_{1,m}$ respectively (i.e. $\mathcal{A}^i$ recognizes $V_{k-i,m}$ where $0 \leq i \leq k-1$):

$$\mathcal{A}^0 = (S^{A^0}, \Sigma, \delta^{A^0}, s_{I^{A^0}}, F^{A^0})$$

$$\ldots$$

$$\mathcal{A}^i = (S^{A^i}, \Sigma \setminus \{0, \ldots, i-1\}, \delta^{A^i}, s_{I^{A^i}}, F^{A^i})$$

$$\ldots$$

$$\mathcal{A}^{k-1} = (S^{A^{k-1}}, \Sigma \setminus \{0, \ldots, k-2\}, \delta^{A^{k-1}}, s_{I^{A^{k-1}}}, F^{A^{k-1}})$$

Let $\mathcal{C}^0 = (S^{C^0}, \Sigma, \delta^{C^0}, s_{I^{C^0}}, F^{C^0})$ be the NFA recognizing $G_{k,m-2,k-1}$ and the following be the NFA's recognizing $G_{k,m-1,1}, \ldots G_{k,m-1,k-2}$ (i.e. $\mathcal{C}^i$ recognizes $G_{k,m,i}$ where $1 \leq i \leq k-2$) .

$$\mathcal{C}^1 = (S^{C^1}, \Sigma, \delta^{C^1}, s_{I^{C^1}}, F^{C^1})$$

$$\ldots$$

$$\mathcal{C}^{k-2} = (S^{C^{k-2}}, \Sigma, \delta^{C^{k-2}}, s_{I^{C^{k-2}}}, F^{C^{k-2}})$$

Also, let $\mathcal{C}^{k-1} = (S^{C^{k-1}}, \Sigma \setminus \{k-1\}, s\delta^{C^{k-1}}, s_{I^{C^{k-1}}}, F^{C^{k-1}})$ be the NFA recognizing $G_{k-1,m-1,k-1}$.

The following NFA $\mathcal{D}_{k,m}$ accepts $H_{k,m}$:

1. $S = \{s_0\} \cup S^{A^{k-1}} \cup \ldots \cup S^{A^0} \cup S^{C^{k-1}} \cup \ldots S^{C^1} \cup S^{C^0} \cup \{s_F\}$.
2. $I = \{s_0\}$ and $F = \{s_F\}$.
3. For every $\sigma \in \Sigma$, $\delta(s_0, \sigma) = \{s_0, s_{I^{A^\sigma}}\}$ and $\delta(s_F, \sigma) = \{s_F\}$.
4. For $s \in S^{A^i}$ $(1 \leq i \leq k)$ and $\sigma \in \Sigma$, $\delta(s, \sigma) = \delta^{A^i}(s, \sigma)$. Similarly for $s \in S^{C^j}$ $(0 \leq j \leq k-1)$.
5. For every $0 \leq i \leq k-1$, the following conditions hold:
   (a) $s_F \in \delta(s_F^{A^i}, \sigma)$ for every $\sigma \in \Sigma \setminus \{i\}$.
   (b) For $i > 0$, $s_F \in \delta(s_{\alpha,m-1}^{C^i}, \sigma)$ for every $\alpha$ in the alphabet of $C_i$ and $\sigma \in \Sigma \setminus \{i\}$. For $i = 0$, $s_F \in \delta(s_{\alpha,m-2}^{C^0}, \sigma)$ for every $\alpha \in \Sigma$ and $\sigma \in \Sigma \setminus \{0\}$.
6. For every $1 \leq i \leq k-1$, the following conditions hold:
   (a) For $\sigma \in \{0, \ldots, i-1\}$, $s_{\sigma,0}^{C^i} \in \delta(s_{I^{A^i}}, \sigma)$.
   (b) For $0 \leq j \leq k-i-1$ and $1 \leq j' \leq m-1$, $s_{\sigma,j'}^{C^i} \in \delta(s_{j,j'}^{A^i}, \sigma)$ for every $\sigma \in \{0, \ldots, j+i-1\}$.
7. For $i = 0$, the following is true:
   (a) For $1 \leq j \leq k-1$ and $1 \leq j' \leq m-1$, $s_{\sigma,j'-1}^{C^0} \in \delta(s_{j,j'}^{A^0}, \sigma)$ for every $\sigma \in \{0, \ldots, j-1\}$.

The NFA recognizing $H_{2,4}$ is shown in Figure 8. Since the NFA recognizing $V_{k,m}$ has $k(m-1)+2$ states and the NFA for $G_{k,m,\alpha}$ has $\alpha(m+1)+(k-\alpha)m$ states, it is not hard to see that the NFA recognizing $H_{k,m}$ has $O(k^2m) = O(m)$ states.

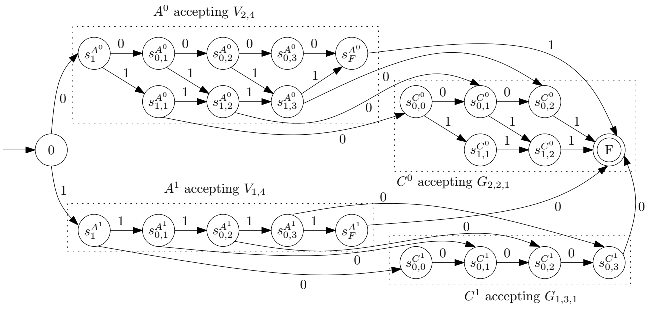


Figure 8: The NFA accepting $H_{2,4}$

Let $\mathcal{A} = (S, \Sigma, \delta, s_I, F)$ be a NFA recognizing $H_{k,m}^c$. Consider any word $w \in V_{k,m+1}$. Then $w \cdot w \in H_{k,m}^c$ since any symbols in $w \cdot w$ that are separated by $m$ positions are identical. We define $S(w) = \{s \in S \mid s \in \delta^+(s_I, w) \text{ and } \delta^+(s, w) \cap F \neq \emptyset\}$. Consider any other word $w' \in V_{k,m+1}$. Assume for the sake of contradiction that $S(w) \cap S(w') \neq \emptyset$. Then there is a state $s \in S(w) \cap S(w')$ and we have $\delta^+(s_I, w \cdot w') \cap F \neq \emptyset$ and $\delta^+(s_I, w' \cdot w) \cap F \neq \emptyset$. Hence $\mathcal{A}$ accepts $w \cdot w'$ and $w' \cdot w$.

However $w$ and $w'$ are distinct words and differ for at at least one position $1 \leq p \leq m+1$. Hence $w \cdot w'$ is of the form $x_1 x_2 \ldots x_{p-1} a \ldots x_{m+1} x'_1 x'_2 \ldots x'_{p-1} b \ldots x'_{m+1}$ such that $a \neq b$. There are two cases:

1. $x_{m+1} \leq x'_1$: Since $w, w' \in V_{k,m}$, it is not hard to see that $a x_{p+1} \ldots x'_{p-1} \in 0^* \cdot 1^* \cdot \ldots \cdot (k-1)^*$ and $b \in 0^* \cdot 1^* \cdot \ldots \cdot (k-1)^*$. Hence $w \cdot w'$ is of the form $\Sigma^* \cdot a \cdot y \cdot b \cdot \Sigma^*$ where $y_1 = x_{p+1} \ldots x'_{p-1}$ and $y_2 = \lambda$ and $a \cdot y_1, y_2 \cdot b \in 0^* \cdot 1^* \cdot \ldots \cdot (k-1)^*$.
2. $x_{m+1} > x'_1$: In this case $a x_{p+1} \ldots x_{m+1} \in 0^* \cdot 1^* \cdot \ldots \cdot (k-1)^*$ and $x'_1 \ldots x'_{p-1} b \in 0^* \cdot 1^* \cdot \ldots \cdot (k-1)^*$. Hence, $w \cdot w'$ is of the form $\Sigma^* \cdot a \cdot y \cdot b \cdot \Sigma^*$ where $y_1 = x_{p+1} \ldots x_{m+1}$ and $y_2 = x'_1 \ldots x'_{p-1}$ and $a \cdot y_1, y_2 \cdot b \in 0^* \cdot 1^* \cdot \ldots \cdot (k-1)^*$.

In both cases $w \cdot w' \notin H_{k,m}^c$ but the word is accepted by $\mathcal{A}$. A very similar argument can be made for $w' \cdot w$. We have arrived at a contradiction. By Lemma 7 there are $O(m^{k-1})$ words in $V_{k,m+1}$ and hence $\mathcal{A}$ has at least $O(m^{k-1})$ states. □

In the following theorem we give an upper bound for the DFA recognizing the complement of the language $H_{k,m}$.

**Lemma 14.** *For every $k, m > 1$, the DFA recognizing the complement of the language $H_{k,m}$ has at most $O(m^{2k})$ states.*

*Proof.* We use the Myhill-Nerode theorem to prove this bound. First we observe that for any words $u, v \in H_{k,m}$, we have $u \equiv v$.

Now consider any word $w \notin H_{k,m}$ such that $w \neq \lambda$. Then $w$ must be of the form $\Sigma^* \cdot a \cdot y$ where $a \in \Sigma$ and $0 \leq |y| \leq m$. Here $y$ is the maximal length word such that $y = y_1 \cdot y_2$ and $a \cdot y_1 \in 0^* \cdot \ldots (k-1)^*$.

Consider any other word $w' \notin H_{k,m}$ such that $w' \in \Sigma^* \cdot a \cdot y$. Then it is not hard to see that $w \equiv w'$ since $w \cdot x \in H_{k,m}$ iff $w' \cdot x \in H_{k,m}$ for any $x \in \Sigma^*$. There are at most $O(m^{2k})$ words of the form $a \cdot y$ and hence there are at most $O(m^{2k})$ equivalence classes. □

**Lemma 15.** *For $n > 0$ and $\alpha = 2^n - n + 1$, the NFA $\mathcal{A}$ constructed in (Jirásková 2008) with $n$-state such that the NFA accepting the complement has $\alpha$ states has $O(n^2)$ number of transitions.*

*Proof.* The NFA $\mathcal{A}$ constructed in (Jirásková 2008) with $n$ states has exactly has an alphabet of five symbols $a, b, c, d, f$. The number of transitions for symbols $a, b, c, d$ are exactly the same as those for the NFA constructed in (Jirásek et al. 2007) which is $O(n^2)$ by lemma 10. The symbols $f$ only adds $O(n)$ number of transitions. Hence, the NFA $\mathcal{A}$ constructred in (Jirásková 2008) has $O(n^2)$ number of transitions. □

**Lemma 16.** *For $n > 0$ and $\alpha = 2^n - n + 1$, the $O(n)$-state NFA $D_{k,m}$ accepting $H_{k,m}$, such that the NFA accepting $H^c_{k,m}$ has $O(\alpha)$ states, has $O(\frac{n^2}{log_2 n})$ transitions.*

*Proof.* In order for the minimal NFA for $H^c_{k,m}$ to have $O(\alpha)$ transitions, we must have $k \in O(\frac{n}{log_2 n})$. The NFA $D_{k,m}$ has $O(k^3 m)$ number of transitions since the NFA's for $V_{k,m}$ and $G_{k,m,\alpha}$ have $O(k^2 m)$ transitions each. Also $D_{k,m}$ has $O(k^2 m)$ states by lemma 13. Hence $D_{k,m}$ has $O(kn)$ transitions where $n \in O(k^2 m)$. Since $k \in O(\frac{n}{log_2 n})$, it is clear that $D_{k,m}$ has $O(\frac{n^2}{log_2 n})$ transitions. □

The following theorem follows from lemmas 13, 14, 15 and 16 proved above.

**Theorem 5.** *For every $k, m > 1$, there exists a NFA $\mathcal{A}$ with $O(m)$ states such that:*

1. *The minimal NFA recognizing the complement of $L(\mathcal{A})$ has between $O(m^{k-1})$ and $O(m^{2k})$ states.*
2. *In the worst case, the NFA $\mathcal{A}$ has $O(\frac{n^2}{log_2 n})$ transitions which is asymptotically fewer than the $O(n^2)$ transitions of the NFA described in (Jirásková 2008).*

## References

Birget, J.-C. (1992), 'Intersection and union of regular languages and state complexity', *Inform. Process. Lett.* **43**(4), 185–190.

Câmpeanu, C., Culik, K., Salomaa, K. & Yu, S. (2001), State complexity of basic operations on finite languages, *in* 'Automata Implementation', Vol. 2214 of *Lecture Notes in Comput. Sci.*, Springer, Berlin / Heidelberg, pp. 148–157.

Gramlich, G. & Schnitger, G. (2007), 'Minimizing NFA's and regular expressions', *J. Comput. Syst. Sci.* **73**(6), 908–923.

Holzer, M. & Kutrib, M. (2003*a*), 'Nondeterministic descriptional complexity of regular languages', *Internat. J. Found. Comput. Sci.* **14**(6), 1087–1102. Selected papers from CIAA 2002 (Tours).

Holzer, M. & Kutrib, M. (2003*b*), State complexity of basic operations on nondeterministic finite automata, *in* 'Implementation and application of automata', Vol. 2608 of *Lecture Notes in Comput. Sci.*, Springer, Berlin, pp. 148–157.

Hopcroft, J. E. & Ullman, J. D. (1979), *Introduction to automata theory, languages, and computation*, Addison-Wesley Publishing Co., Reading, Mass. Addison-Wesley Series in Computer Science.

Iwama, K., Kambayashi, Y. & Takaki, K. (2000), 'Tight bounds on the number of states of DFA's that are equivalent to $n$-state NFA's', *Theor. Comput. Sci.* **237**(1-2), 485–494.

Jirásek, J., Jirásková, G. & Szabari, A. (2005), State complexity of concatenation and complementation of regular languages, *in* 'Implementation and application of automata', Vol. 3317 of *Lecture Notes in Comput. Sci.*, Springer, Berlin, pp. 178–189.

Jirásek, J., Jirásková, G. & Szabari, A. (2007), Deterministic blow-ups of minimal nondeterministic finite automata over a fixed alphabet, *in* 'Developments in language theory', Vol. 4588 of *Lecture Notes in Comput. Sci.*, Springer, Berlin, pp. 254–265.

Jirásková, G. (2005), 'State complexity of some operations on binary regular languages', *Theoret. Comput. Sci.* **330**(2), 287–298.

Jirásková, G. (2008), On the state complexity of complements, stars, and reversals of regular languages, *in* 'Developments in language theory', Vol. 5257 of *Lecture Notes in Comput. Sci.*, Springer, Berlin, pp. 431–442.

Jirásková, G. (2009), Magic numbers and ternary alphabet, *in* V. Diekert & D. Nowotka, eds, 'Developments in Language Theory', Vol. 5583 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 300–311.

Moore, F. R. (1971), 'On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata', *IEEE Trans. Comput.* **20**(10), 1211–1214.

Nerode, A. (1958), 'Linear automaton transformations', *Proc. Amer. Math. Soc.* **9**, 541–544.

Piotr Berman, A. L. (1977), *On complexity of regular languages in terms of finite automata*, Institute of Computer Science, Polish Academy of Sciences.

Rabin, M. O. & Scott, D. (1959), 'Finite automata and their decision problems', *IBM J. Res. Develop.* **3**, 114–125.

Salomaa, A., Salomaa, K. & Yu, S. (2007), 'State complexity of combined operations', *Theoret. Comput. Sci.* **383**(2-3), 140–152.

Schnitger, G. (2006), Regular expressions and NFA's without $\epsilon$-transitions, *in* 'in 23th Symposium on Theoretical Aspects of Computer Science (STACS 2006), LNCS 3884 (2006', pp. 432–443.

Yu, S. (2005), 'State complexity: recent results and open problems', *Fund. Inform.* **64**(1-4), 471–480.