International Journal of Foundations of Computer Science
Vol. 22, No. 2 (2011) 331–344

DOI: 10.1142/S0129054111008076

MAGIC NUMBERS AND TERNARY ALPHABET*

GALINA JIRÁSKOVÁ†
Mathematical Institute, Slovak Academy of Sciences,
Grešákova 6, 040 01 Košice, Slovakia
jiraskov@saske.sk
http://<im3.saske.sk/∼jiraskov>

Received 30 October 2009
Accepted 14 July 2010
Communicated by Volker Diekert and Dirk Nowotka

A number α, in the range from n to 2^n, is magic for n with respect to a given alphabet size s, if there is no minimal nondeterministic finite automaton of n states and s input symbols whose equivalent minimal deterministic finite automaton has α states. We show that in the case of a ternary alphabet, there are no magic numbers. For all n and α satisfying $n \leqslant \alpha \leqslant 2^n$, we define an n-state nondeterministic finite automaton with a three-letter input alphabet that requires exactly α deterministic states.

Keywords: Deterministic and nondeterministic finite automata; minimal automata.

1991 Mathematics Subject Classification: 68Q19, 68Q45

1. Introduction

In 1997, at the Third Conference on Developments in Language Theory, Iwama, Kambayashi, and Takaki [10] stated the question of whether there always exists a minimal nondeterministic finite automaton of n states whose equivalent minimal deterministic automaton has α states for all integers n and α satisfying $n \leqslant \alpha \leqslant 2^n$. The question has also been considered by Iwama, Matsuura, and Paterson in [11], where a number α with $n \leqslant \alpha \leqslant 2^n$ is called "magic", if there is no nondeterministic finite automaton of n states that needs α deterministic states. In these two papers, it has been shown that if $\alpha = 2^n - 2^k$ or $\alpha = 2^n - 2^k - 1$, where $0 \leqslant k \leqslant n/2 - 2$, or if $\alpha = 2^n - k$, where $2 \leqslant k \leqslant 2n - 2$ and some coprimality condition holds, then α is not magic. The authors defined corresponding nondeterministic finite automata over a two-letter alphabet, and mentioned that if we allow more input symbols, then the problem becomes easier.

*This paper has been presented at the Thirteenth International Conference on Developments in Language Theory (DLT 2009) held in Stuttgart, Germany on June 30 - July 3, 2009.
†Research supported by VEGA grant 2/0111/09.

In the case when the alphabet size is allowed to grow exponentially with n, appropriate automata have been described for all values of n and α by the author in [12]. It has been shown that a $2n$-letter alphabet would be enough; however, in this case, corresponding automata were given only implicitly. For a binary alphabet, all numbers between n and $n^2/2$ have been proved to be non-magic in [12].

The explicit constructions of nondeterministic finite automata using $n+2$ input symbols have been presented by Geffert [6]. He also considered a binary alphabet, and provided a description of binary nondeterministic automata requiring α deterministic states for each value of α in the range from n to $2^{n^{1/3}}$.

The problem has been solved for a fixed four-letter alphabet by Jirásek, Jirásková, and Szabari [14] by describing, for all n and α such that $n \leqslant \alpha \leqslant 2^n$, a minimal nondeterministic finite automaton of n states with a four-letter input alphabet that needs α deterministic states. This means that in the case of a four-letter alphabet, there are no magic numbers.

Let us note that in the unary case, all numbers between $\mathrm{e}^{(1+o(1))\cdot\sqrt{n \ln n}}$ and 2^n are magic since every n-state unary nondeterministic finite automaton can be simulated by an $\mathrm{e}^{(1+o(1))\cdot\sqrt{n \ln n}}$-state deterministic finite automaton [15, 4, 7]. Moreover, Geffert proved in [7] that there are many more magic than non-magic numbers in the range from n to $\mathrm{e}^{(1\pm o(1))\cdot(\sqrt{n \ln n})}$ in the unary case.

Here we continue this research and study the ternary case. We show that neither in this case do magic numbers exist, and give explicit constructions of appropriate nondeterministic finite automata with a three-letter input alphabet. Surprisingly, the constructions and proofs are even easier than in the case of a four-letter alphabet [14]. The question of whether or not there are magic numbers for a binary alphabet remains open.

Some partial results for the binary case have recently been obtained by Matsuura and Saito in [16] and by the author in [13]. The first paper shows that, with some exceptions, all numbers from 2^n to $2^n - 4n$ are non-magic, while in the the second one, all numbers from n to $2^{n/3}$ are proved to be non-magic.

To conclude this section we mention two more related works. Magic numbers for symmetric difference nfa's have been studied by Zijl [20], and similar problems for nonterminal complexity of some operations on context-free languages have been investigated by Dassow and Stiebe [5].

2. Preliminaries

In this section, we give some basic definitions, notations, and preliminary results used throughout the paper. For further details, we refer the reader to [18, 19].

Let Σ be a finite alphabet and Σ^* the set of all strings over the alphabet Σ including the empty string ε. The length of a string w is denoted by $|w|$. A *language* is any subset of Σ^*. We denote the cardinality of a finite set A by $|A|$ and its power-set by 2^A.

A *deterministic finite automaton* (dfa) is a quintuple $M = (Q, \Sigma, \delta, s, F)$, where Q is a finite non-empty set of states, Σ is a finite non-empty input alphabet, δ is the transition function that maps $Q \times \Sigma$ to Q, s is the initial (start) state, $s \in Q$, and F is the set of accepting (final) states, $F \subseteq Q$. In this paper, all deterministic automata are assumed to be complete, that is, the next state $\delta(q, a)$ is defined for each state q in Q and each symbol a in Σ. The transition function δ is extended to a function from $Q \times \Sigma^*$ to Q in a natural way. A string w in Σ^* is accepted by dfa M if state $\delta(s, w)$ is an accepting state of dfa M. The *language accepted by* dfa M, denoted $L(M)$, is the set of strings accepted by dfa M, that is, $L(M) = \{w \in \Sigma^* \mid \delta(s, w) \in F\}$.

A *nondeterministic finite automaton* (nfa) is a quintuple $M = (Q, \Sigma, \delta, s, F)$, where Q, Σ, s, and F are defined in the same way as for a dfa, and δ is a nondeterministic transition function that maps $Q \times \Sigma$ to 2^Q. The transition function can be naturally extended to the domain $Q \times \Sigma^*$, and then to the domain $2^Q \times \Sigma^*$. A string w in Σ^* is accepted by nfa M if the set $\delta(s, w)$ contains an accepting state of nfa M. The *language accepted by* nfa M is the set of strings accepted by nfa M, that is, $L(M) = \{w \in \Sigma^* \mid \delta(s, w) \cap F \neq \varnothing\}$.

Two automata are said to be *equivalent* if they accept the same language. A (non)deterministic finite automaton M is called *minimal* if every (non)deterministic finite automaton equivalent to M has at least as many states as M. It is well-known that a dfa $(Q, \Sigma, \delta, s, F)$ is minimal if *(i)* all of its states are reachable from the initial state (that is, for each state q, there is a string w such that $\delta(s, w) = q$), and *(ii)* no two distinct states are equivalent (states p and q are said to be equivalent if for all strings w in Σ^*, state $\delta(p, w)$ is accepting if and only if state $\delta(q, w)$ is accepting). Each regular language has a unique minimal dfa, up to isomorphism. However, the same result does not hold for nfa's.

Every nondeterministic finite automaton $M = (Q, \Sigma, \delta, s, F)$ can be converted to an equivalent deterministic finite automaton $M' = (2^Q, \Sigma, \delta', s', F')$ using an algorithm known as the "subset construction" [17] in the following way. Every state of dfa M' is a subset of state set Q. The initial state of dfa M' is the singleton set $\{s\}$. The transition function δ' is defined by $\delta'(R, a) = \delta(R, a)$ for each state R in 2^Q and each symbol a in Σ. A state R in 2^Q is an accepting state of dfa M' if it contains an accepting state of nfa M. We call automaton M' the *subset automaton* corresponding to nfa M. Notice that the subset automaton need not be minimal, since some of its states may be unreachable or equivalent. The following two lemmata help us to prove reachability and inequivalence of states of subset automata in the next section.

Lemma 1. *Let $\mathcal{R}$ be a family of some reachable subsets in the subset automaton corresponding to an nfa $(Q, \Sigma, \delta, s, F)$ such that*

(1) the initial subset $\{s\}$ is in $\mathcal{R}$,

(2) for each subset S in $\mathcal{R}$ and each symbol a in Σ, the subset $\delta(S, a)$ is in $\mathcal{R}$.
Then the family $\mathcal{R}$ consists of all reachable subsets *of the subset automaton.*

Proof. Let S be a reachable subset in the subset automaton. Then there is a string w in Σ^* such that $S = \delta(\{s\}, w)$. We prove by induction on $|w|$ that S is in $\mathcal{R}$.

The basis, $|w| = 0$, holds true since the initial subset $\{s\}$ is in family $\mathcal{R}$ by (1). Assume that every subset that can be reached from the initial subset by a string of length k is in $\mathcal{R}$. Let $S = \delta(\{s\}, w)$ and $|w| = k + 1$. Then there is a symbol a in Σ and a string v of length k such that $w = va$. Let $T = \delta(\{s\}, v)$. Then $S = \delta(T, a)$. By the induction hypothesis, subset T is in family $\mathcal{R}$. Hence by (2), subset S is in family $\mathcal{R}$ as well, and our proof is complete. $\square$

Lemma 2. *Let M be an nfa such that for each state q of M, there exists a string w_q which is accepted by M started in state q but is not accepted by M started in any other state. Then no two different states of the subset automaton corresponding to nfa M are equivalent.*

Proof. Let S and T be two different states in the subset automaton corresponding to nfa M. Then there is a state q of M that is in one of these two sets but not in the other. Without loss of generality, let q be in S. This means that the string w_q is accepted by the subset automaton from state S. Since w_q is not accepted by M started in any state different from q, this string is not accepted by the subset automaton from state T, and so distinguishes S and T. $\square$

To prove that an nfa is minimal we use a fooling-set lower-bound technique [1, 2, 3, 8, 9]. Let us recall the definition of a fooling set, and the lemma from [2] describing this lower-bound technique.

Definition 3. *A set $\{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$ of pairs of strings is said to be a* fooling set *for a regular language L if for every i and j in $\{1, 2, \ldots, n\}$,*

(F1) the string $x_i y_i$ is in L, and

(F2) if $i \neq j$, then at least one of the strings $x_i y_j$ and $x_j y_i$ is not in L.

Lemma 4 ([2], Lemma 1) *Let $\mathcal{A}$ be a fooling set for a regular language L. Then every nfa for L needs at least $|\mathcal{A}|$ states.* $\square$

The following lemma from [14] shows that each integer can be expressed as a sum of powers of 2 decreased by 1 if, when necessary, the smallest summand can be taken twice. We will use this lemma later in our constructions.

Lemma 5 ([14], Lemma 2) *Let k be a positive integer. Then for each integer m such that $1 \leqslant m < 2^k$, one of the following three cases holds:*

$$
\begin{aligned}
m &= 2^k - 1,\\
m &= (2^{k_1} - 1) + (2^{k_2} - 1) + \cdots + (2^{k_{\ell-1}} - 1) + (2^{k_\ell} - 1),\\
m &= (2^{k_1} - 1) + (2^{k_2} - 1) + \cdots + (2^{k_{\ell-1}} - 1) + 2 \cdot (2^{k_\ell} - 1),
\end{aligned}
$$

where $1 \leqslant \ell \leqslant k - 1$, and $k - 1 \geqslant k_1 > k_2 > \cdots > k_\ell \geqslant 1$. $\square$

3. Main Results

The aim of this section is to show that in the case of a three-letter alphabet, there are no magic numbers, which means that each value in the range from n to 2^n can be the size of the minimal dfa equivalent to a minimal n-state nfa defined over a three-letter alphabet. Let us start with an example.

Example 6. Consider the five-state nfa A_5 with the input alphabet $\{a, b, c\}$ shown in Figure 1. The automaton has states $1, 2, 3, 4, 5$, and state 5 is both the initial state and sole accepting state. On inputs a and b, each state i goes to state $i + 1$, except for state 5 which goes to itself. Moreover, on input b, each state goes to state 1. There is only one transition on input c, which goes from state 4 to state 5.

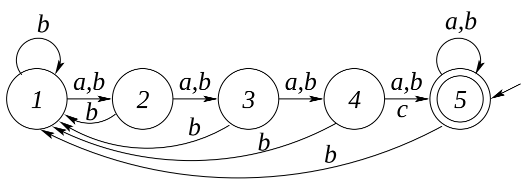

Fig. 1. The nondeterministic finite automaton A_5.

Let A_5' be the corresponding subset automaton, shown in Figure 2. To keep the figure transparent, we label the states of the dfa without commas and brackets, omit transitions on a and b in some accepting states, and also all transitions on c except for the transition from state $\{4\}$. We can see that state $\{5\}$ goes to a subset containing state 5 by each string in $\{a, b\}^*$. Moreover, each such subset is reachable from $\{5\}$ by a string in $\{a, b\}^*$. We prove this property in the general case later.

Next, consider states $\{2\}, \{2, 3\}, \{2, 3, 4\}$, depicted by a red (grey) color in the figure. Notice that from state $\{2\}$, seven subsets of $\{1, 2, 3, 4\}$, none of which contains state 5, can be reached by a string over $\{a, b\}$. Three more such subsets can be reached from state $\{2, 3\}$, and one more subset from state $\{2, 3, 4\}$.

Thus, if we would like to have a five-state nfa requiring, for example, $17 + 7 + 3 + 1$ $(= 28)$ deterministic states —let us call it $B_{5,28}$— we can construct it from nfa A_5 by adding transitions on symbol c from states $1, 2$, and 3 to subsets $\{2\}$, $\{2, 3\}$, and $\{2, 3, 4\}$, respectively. To see that nfa $B_{5,28}$ is minimal let us show that the set $\{(\varepsilon, ba^3c), (b, a^3c), (ba, a^2c), (ba^2, ac), (ba^3, c)\}$ is a fooling set of size five for language $L(B_{5,28})$. The string ba^3c is accepted by nfa $B_{5,28}$, and so (F1) from Definition 3 holds. Since no string in a^*c and no string ba^rc with $r \neq 3$ is accepted by $B_{5,28}$, (F2) holds as well. By Lemma 4, nfa $B_{5,28}$ is minimal. Next, the empty string is accepted by $B_{5,28}$ only from state 5, while the string $a^{4-i}c$ is accepted only from state i for $i = 1, 2, 3, 4$. By Lemma 2, no two different states of the subset automaton corresponding to nfa $B_{5,28}$ are equivalent. The following subsets are reachable in the subset automaton:

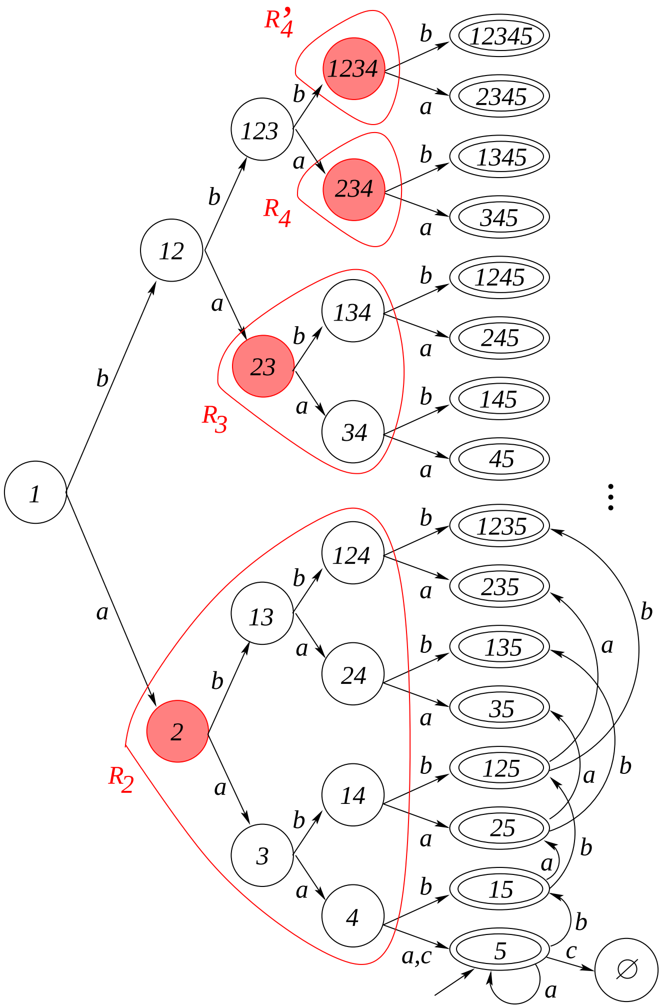

Fig. 2. The deterministic finite automaton A'_5.

- the empty set and all subsets containing state 5 (17 subsets),
- state $\{1, 5\}$ goes by c to state $\{2\}$, from which 7 subsets of $\{1, 2, 3, 4\}$ can be reached,
- state $\{2, 5\}$ goes by c to state $\{2, 3\}$, from which 3 more subsets of $\{1, 2, 3, 4\}$ can be reached,
- state $\{3, 5\}$ goes by c to a new subset $\{2, 3, 4\}$,
- no other subset is reachable.

Hence the total number of reachable states is $17 + 7 + 3 + 1$ $(= 28)$.

If we would like to get a five-state nfa $B_{5,17+7+7}$ requiring 31 deterministic states, we would define transitions on symbol c by $1 \xrightarrow{c} \{2\}$, $2 \xrightarrow{c} \{1,2\}$, and $4 \xrightarrow{c} \{5\}$, while in the case of nfa $B_{5,17+7+3+3}$ requiring 30 deterministic states we would have $1 \xrightarrow{c} \{2\}$, $2 \xrightarrow{c} \{2,3\}$, $3 \xrightarrow{c} \{1,2,3\}$, and $4 \xrightarrow{c} \{5\}$. A problem could arise if we would like to reach $17+7+3+1+1$ states in the subset automaton; however, in this case we will use $4 \xrightarrow{c} \{1,2,3,4\}$ instead of $4 \xrightarrow{c} \{5\}$, and the states still will not be equivalent. In such a way, using Lemma 5, we can define a five-state nondeterministic finite automaton $B_{5,17+m}$ that needs $17+m$ deterministic states for each m with $1 \leqslant m \leqslant 15$. □

Let us now generalize the above example. Our first aim is to define a ternary k-state nfa $B_{k,\beta}$ that needs β deterministic states, for each β greater than 2^{k-1}. We will use these automata later to get the whole range of complexities from n to 2^n.

To this aim define a ternary k-state nfa $A_k = (Q_k, \{a,b,c\}, \delta, k, \{k\})$, where $Q_k = \{1, 2, \ldots, k\}$, and for each i in Q_k,

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

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

$$\delta(i,c) = \begin{cases} \{k\}, & \text{if } i = k-1, \\ \varnothing, & \text{if } i \neq k-1, \end{cases}$$

that is, on inputs a and b, each state i goes to state $i+1$ except for state k which goes to itself. Moreover, on symbol b, each state also goes to state 1. Transitions on input c are defined only in state $k-1$, and this state goes to state k on symbol c. Automaton A_k is depicted in Figure 3.

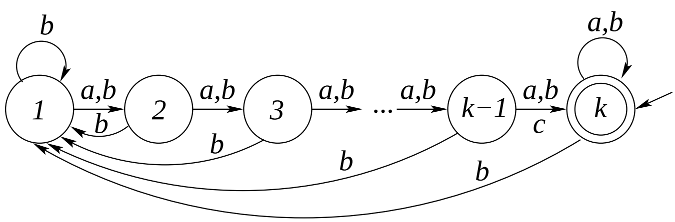

Fig. 3. The nondeterministic finite automaton A_k.

Let $A'_k = (2^{Q_k}, \{a,b,c\}, \delta', \{k\}, F')$ be the subset automaton corresponding to nfa A_k. Recall that the subset automaton contains all 2^k states, some of which may be unreachable. Since the initial state k of A_k goes to itself on a and b, and the only transition on c goes to state k, all nonempty subsets of Q_k that do not contain state k are unreachable in the subset automaton. On the other hand, let us show that all other subsets are reachable.

Lemma 7. *All subsets of state set Q_k containing state k are reachable in subset automaton A'_k from the initial state $\{k\}$ by a string over the binary alphabet $\{a, b\}$. The empty set is reachable as well.*

Proof. The empty set is reachable, since the initial state state k goes to the empty set by symbol c. We prove by induction on the size of subsets that each subset containing state k is reachable. The singleton set $\{k\}$ is reachable since it is the initial state of subset automaton A'_k. Let $1 \leqslant t \leqslant k-1$ and assume that each subset of size t containing state k is reachable by a string over $\{a, b\}$. Let $\{i_1, i_2, \ldots, i_t, k\}$ be a subset of size $t+1$ such that $1 \leqslant i_1 < i_2 < \cdots < i_t < k$. Then we have

$$\{i_1, i_2, \ldots, i_t, k\} = \delta'(\{i_2 - i_1, i_3 - i_1, \ldots, i_t - i_1, k\}, ba^{i_1 - 1}),$$

where the latter set of size t containing state k is reachable by a string over $\{a, b\}$ by the induction hypothesis. Hence the subset $\{i_1, i_2, \ldots, i_t, k\}$ is reachable by a string over $\{a, b\}$, and this completes our proof. ☐

If we look at dfa A'_5 without the dead state in our Example 1, in Figure 2 on page 336, like at a tree rooted in state $\{1\}$, then all (accepting) states that are leaves of the tree can be reached from the initial state $\{5\}$ by a string over $\{a, b\}$. The empty set is reached from state $\{5\}$ by c. Hence the dfa has 17 reachable states.

In the general case, subset automaton A'_k has $2^{k-1} + 1$ reachable states. We now continue our constructions by adding new transitions on input c to get nondeterministic automata whose corresponding subset automata have more reachable states. As a result, we will be able to construct an nfa requiring β deterministic states for each β between $2^{k-1} + 1$ and 2^k.

To this aim consider states $\{2\}$, $\{2, 3\}$, $\{2, 3, 4\}$, $\ldots$, $\{2, 3, 4, \ldots, k-1\}$ of subset automaton A'_k. Notice that $\{2\} \subseteq \{2, 3\} \subseteq \{2, 3, 4\} \subseteq \cdots \subseteq \{2, 3, 4, \ldots, k-1\}$, which is an important property that will be crucial in the proof of our main result. Consider also all subsets of Q_k not containing state k that can be reached from these states by strings over the binary alphabet $\{a, b\}$, and let us introduce some notation. Let $2 \leqslant r \leqslant k-1$, and let

$$\mathcal{R}_1 = \big\{R \subseteq Q_k \setminus \{k\} \mid R = \delta'(\{1\}, w) \text{ for some string } w \text{ over } \{a, b\}\big\},$$
$$\mathcal{R}_r = \big\{R \subseteq Q_k \setminus \{k\} \mid R = \delta'(\{2, 3, \ldots, r\}, w) \text{ for some string } w \text{ over } \{a, b\}\big\},$$
$$\mathcal{R}'_r = \big\{R \subseteq Q_k \setminus \{k\} \mid R = \delta'(\{1, 2, 3, \ldots, r\}, w) \text{ for some string } w \text{ over } \{a, b\}\big\},$$

that is, $\mathcal{R}_1$, $\mathcal{R}_r$, and $\mathcal{R}'_r$ are the families of subsets of state set Q_k that do not contain state k and can be reached by strings over the binary alphabet $\{a, b\}$ from states $\{1\}$, $\{2, 3, \ldots, r\}$, and $\{1, 2, 3, \ldots, r\}$, respectively. In our Example 6 in Figure 2 on page 336 we have $\mathcal{R}_3 = \big\{\{2, 3\}, \{3, 4\}, \{1, 3, 4\}\big\}$, $\mathcal{R}_4 = \big\{\{2, 3, 4\}\big\}$, and $\mathcal{R}'_4 = \big\{\{1, 2, 3, 4\}\big\}$. We can see that in this example, the families $\mathcal{R}_2$, $\mathcal{R}_3$, and $\mathcal{R}_4$ are pairwise disjoint and contain $2^3 - 1, 2^2 - 1$, and $2^1 - 1$ states, respectively. Let us consider the general case.

Lemma 8. *Let $2 \leqslant r \leqslant s \leqslant k-1$ and let $\mathcal{R}_1$, $\mathcal{R}_r$, and $\mathcal{R}'_r$ be the families of states of subset automaton A'_k defined above. Then we have:*

(i) The size of family $\mathcal{R}_1$ is $2^{k-1}-1$.
(ii) The size of family $\mathcal{R}_r$ and of family $\mathcal{R}'_r$ is $2^{k-r}-1$.
(iii) Families $\mathcal{R}_r$ and $\mathcal{R}'_s$ are disjoint.
(iv) If $r < s$, then families $\mathcal{R}_r$ and $\mathcal{R}_s$ are disjoint.

Proof. First, let us prove by induction on ℓ that the set of states reachable from state $\{2,3,\ldots,r\}$ by strings over $\{a,b\}$ of length ℓ, with $0 \leqslant \ell \leqslant k-r-1$, is

$$\big\{S \cup \{2+\ell, 3+\ell, \ldots, r+\ell\} \mid S \subseteq \{1,2,\ldots,\ell\}\big\}.$$

The basis, $\ell = 0$, holds true since the only state reachable by the empty string is $\{2,3,\ldots,r\}$. Assume that $0 \leqslant \ell \leqslant k-r-2$, and that the claim holds for ℓ. Let w be a string over $\{a,b\}$ of length $\ell+1$. Then $w = va$ or $w = vb$, where v is a string of length ℓ. By the induction hypothesis, state $\{2,3,\ldots,r\}$ goes by the string v to a state $S \cup \{2+\ell, 3+\ell, \ldots, r+\ell\}$, for some subset S of the set $\{1,2,\ldots,\ell\}$. Since $\ell + r \leqslant k-2$, this state goes to state

$$\{s+1 \mid s \in S\} \cup \{2+\ell+1, 3+\ell+1, \ldots, r+\ell+1\}$$

by a, and to state

$$\{1\} \cup \{s+1 \mid s \in S\} \cup \{2+\ell+1, 3+\ell+1, \ldots, r+\ell+1\}$$

by b. Here the sets $\{s+1 \mid s \in S\}$ and $\{1\} \cup \{s+1 \mid s \in S\}$ are subsets of $\{1,2,\ldots,\ell+1\}$. Now let $S' \subseteq \{1,2,\ldots,\ell+1\}$, and let us show that the set

$$S' \cup \{2+\ell+1, 3+\ell+1, \ldots, r+\ell+1\}$$

is reachable from state $\{2,3,\ldots,r\}$ by a string of length $\ell+1$. If $1 \notin S$, then this set can be reached from the set $\{s-1 \mid s \in S\} \cup \{2+\ell, 3+\ell, \ldots, r+\ell\}$ by a. Otherwise, it can be reached from the set $\{s-1 \mid s \in S \text{ and } s \neq 1\} \cup \{2+\ell, 3+\ell, \ldots, r+\ell\}$ by b. By the induction hypothesis, both of these sets are reachable from state $\{2,3,\ldots,r\}$ by a string of length ℓ. This completes the proof of the claim.

In a similar way we can prove that the set of states reachable from state $\{1,2,\ldots,r\}$ by strings over $\{a,b\}$ of length ℓ, with $0 \leqslant \ell \leqslant k-r-1$, is

$$\big\{S \cup \{1+\ell, 2+\ell, 3+\ell, \ldots, r+\ell\} \mid S \subseteq \{1,2,\ldots,\ell\}\big\}.$$

Both states $\{2,3,\ldots,r\}$ and $\{1,2,3,\ldots,r\}$ go to a subset containing state k by every string over $\{a,b\}$ of length at least $k-r$. As a corollary, we get (iii) and (iv). Next, it follows that

$$|\mathcal{R}_r| = |\mathcal{R}'_r| = 1+2+4+\cdots+2^{k-r-1} = 2^{k-r}-1 \text{ and } |\mathcal{R}_1| = 2^{k-1}+1,$$

which proves the lemma. □

Using the results of the above lemma we are now able to give, for each β between 2^{k-1} and 2^k, a construction of a k-state nfa that needs β deterministic states.

Lemma 9. *For all integers k and β such that $2^{k-1}+1 \leqslant \beta \leqslant 2^k$, there exists a minimal ternary k-state nfa $B_{k,\beta}$ whose equivalent minimal dfa has β states.*

Proof. Since all k-state nfa's described in this proof require more than 2^{k-1} deterministic states, they must be minimal.

If $\beta = 2^{k-1}+1$, then we set $B_{k,\beta} = A_k$, where A_k is the nfa defined on page 337. The empty string is accepted by A_k only from state k, while for each other state i, the string $a^{k-1-i}c$ is accepted only from state i. By Lemma 2, no two different states of the corresponding subset automaton are equivalent. By Lemma 7, the subset automaton has $2^{k-1}+1$ reachable states.

If $2^{k-1}+1 < \beta \leqslant 2^k$, then $\beta = 2^{k-1}+1+m$ for an integer m such that $1 \leqslant m \leqslant 2^{k-1}-1$. By Lemma 5, one of the following three cases holds for m:

$$m = 2^{k-1} - 1 \tag{1}$$

$$m = (2^{k_1}-1) + (2^{k_2}-1) + \cdots + (2^{k_{\ell-1}}-1) + (2^{k_\ell}-1) \tag{2}$$

$$m = (2^{k_1}-1) + (2^{k_2}-1) + \cdots + (2^{k_{\ell-1}}-1) + 2\cdot(2^{k_\ell}-1) \tag{3}$$

where $1 \leqslant \ell \leqslant k-2$, and $k-2 \geqslant k_1 > k_2 > \cdots > k_\ell \geqslant 1$.

Construct a k-state nfa $B_{k,\beta} = (Q_k, \{a,b,c\}, \delta_B, k, \{k\})$ from nfa A_k by adding transitions on input c depending on m as follows:

- In case (1) holds, add the transition on c from state 1 to $\{1\}$.
- In case (2) holds, add transitions on c from state i to $\{2,3,\ldots,k-k_i\}$ for $i = 1,2,\ldots,\ell$.
- In case (3) holds, add the same transitions as in the case where (2) holds, and add also the transition on c from state $\ell+1$ to $\{1,2,3,\ldots,k-k_\ell\}$. If $\ell = k-2$, that is, if $m = (2^{k-2}-1)+(2^{k-3}-1)+\cdots+3+1+1$, then replace the transition on c from $k-1$ to k with the transition on c from $k-1$ to $\{1,2,\ldots,k-1\}$.

In all cases, the empty string is accepted only from state k, and for every other state i, the string $a^{k-1-i}c$ is accepted only from state i, except in the case where $m = (2^{k-2}-1)+(2^{k-3}-1)+\cdots+3+1+1$. In this case, the string $ca^{k-2}ca$ is accepted only from state $k-1$, since only this state goes to state 1 on symbol c. Next, the string $a^{k-1-i}ca^{k-2}ca$ is accepted only from state i for $i = 1,2,\ldots,k-2$. This means that no two different states of the corresponding subset automata are equivalent.

Set $r_i = k - k_i$ $(1 \leqslant i \leqslant \ell)$ and let $\mathcal{R}$ denote the family consisting of the empty set and all subsets of $\{1,2,\ldots,k\}$ containing state k. Let us show that the following families $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$ consist of all reachable states of the subset automaton $B'_{k,\beta}$ corresponding to nfa $B_{k,\beta}$ in cases (1), (2), and (3), respectively; recall that the families $\mathcal{R}_r$ and $\mathcal{R}'_r$ are defined before Lemma 8 on page 338:

$$\mathcal{S}_1 = \mathcal{R} \cup \mathcal{R}_1,$$

$$\mathcal{S}_2 = \mathcal{R} \cup \mathcal{R}_{r_1} \cup \mathcal{R}_{r_2} \cup \cdots \cup \mathcal{R}_{r_\ell},$$

$$\mathcal{S}_3 = \mathcal{R} \cup \mathcal{R}_{r_1} \cup \mathcal{R}_{r_2} \cup \cdots \cup \mathcal{R}_{r_\ell} \cup \mathcal{R}'_{r_\ell}.$$

The empty set is reachable since no transition on symbol c is defined in state k. All subsets containing state k are reachable by Lemma 7. In case (1) holds, state $\{1, k\}$ goes to state 1 by c, and then all the subsets in $\mathcal{R}_1$ are reachable from state 1 by strings over $\{a, b\}$. In case (2) holds, each state $\{i, k\}$, $1 \leqslant i \leqslant \ell$, goes by c to state $\{2, 3, \dots, k - k_i\}$, that is to state $\{2, 3, \dots, r_i\}$, from which all the subsets in $\mathcal{R}_{r_i}$ are reached by strings over $\{a, b\}$. Similarly, in case (3) holds, all subsets in $\mathcal{R}_{r_i}$, $1 \leqslant i \leqslant \ell$, are reachable, and, moreover, in this case, state $\{\ell + 1, k\}$ goes to state $\{1, 2, 3, \dots, r_\ell\}$, and so all the subsets in $\mathcal{R}'_{r_\ell}$ are reachable as well.

We now show that no other subsets are reachable. By Lemma 1, since the initial state $\{k\}$ is in $\mathcal{S}_1$ ($\mathcal{S}_2, \mathcal{S}_3$), it is enough to prove that each state S in family $\mathcal{S}_1$ ($\mathcal{S}_2, \mathcal{S}_3$, respectively) goes to a state in this family by each of the symbols a, b, c. This is quite straightforward for symbols a and b since every state in $\mathcal{R}_r$ goes either to another state in $\mathcal{R}_r$ or to a state in $\mathcal{R}$ by a and by b. In the case of symbol c, it is important to notice that we have $\delta_B(k - 1, c) = \{k\}$ or $\delta_B(k - 1, c) = \{1, 2, \dots, k - 1\}$, and $\delta_B(1, c) \subseteq \delta_B(2, c) \subseteq \cdots \subseteq \delta_B(\ell, c)$, and in case (3) holds, we also have $\delta_B(\ell, c) \subseteq \delta_B(\ell + 1, c)$. In the other states, transitions on c are not defined. It follows that for each subset S of state set Q_k, the set $\delta_B(S, c)$ is either empty, or contains state k, or is equal to $\delta_B(q, c)$ for the greatest integer q in $\{1, 2, \dots, \ell\}$ (in $\{1, 2, \dots, \ell + 1\}$, respectively) that is in S. In all three cases, the set $\delta(S, c)$ is in family $\mathcal{S}_1$ ($\mathcal{S}_2, \mathcal{S}_3$, respectively).

By Lemma 8, the families $\mathcal{R}_{r_1}, \mathcal{R}_{r_2}, \dots, \mathcal{R}_{r_\ell}$, and $\mathcal{R}'_{r_\ell}$ are pairwise disjoint and

$$|\mathcal{R}_{r_i}| = |\mathcal{R}'_{r_i}| = 2^{k - r_i} - 1 = 2^{k - (k - k_i) - 1} - 1 = 2^{k_i} - 1.$$

Hence the subset automaton corresponding to nondeterministic automaton $B_{k,\beta}$ has $1 + 2^{k-1} + m$ reachable states, and no two different states are equivalent. Since we have $\beta = 1 + 2^{k-1} + m$, our proof is complete. □

We are now able to prove our main result.

Theorem 10. *For all integers n and α such that $n \leqslant \alpha \leqslant 2^n$, there exists a minimal nondeterministic finite automaton of n states with a three-letter input alphabet whose equivalent minimal deterministic finite automaton has exactly α states.*

Proof. If $\alpha = n$, then take an n-state nfa for $\{w \in \{a, b, c\}^* \mid |w| \geqslant n - 1\}$. If $\alpha \geqslant 2^{n-1} + 1$, then take the n-state nfa $B_{n,\alpha}$ given by Lemma 9. Otherwise, let us find an integer k, such that $1 \leqslant k \leqslant n - 1$ and

$$n - (k - 1) + 2^{k-1} \leqslant \alpha < n - k + 2^k.$$

Then

$$\alpha = n - k + 1 + 2^{k-1} + m$$

for some integer m such that $0 \leqslant m < 2^{k-1}$. If we define β by $\beta = 1 + 2^{k-1} + m$, then

$$\alpha = n - k + \beta.$$

Construct an n-state nfa $C_{n,\alpha}$ from the k-state nfa $B_{k,\beta}$ described in the proof of Lemma 9 in the following way. First, add new states $k+1, k+2, \ldots, n$. State n is the initial state, and state k is the sole accepting state of nfa $C_{n,\alpha}$. Add transitions on symbol b from state j to state $j+1$ for each state j greater than k. Add transitions on symbols a and c from state $k+1$ to itself. Automaton $C_{n,\alpha}$ for $\alpha = n-k+1+2^{k-1}$ is shown in Figure 4.

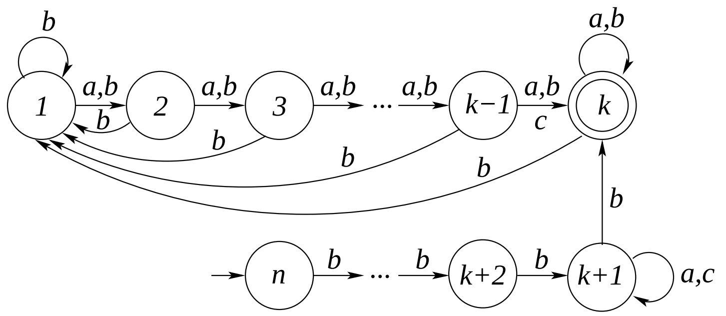

Fig. 4. The nondeterministic finite automaton $C_{n,n-k+1+2^{k-1}}$.

To prove that nfa $C_{n,\alpha}$ is minimal, consider the following set of pairs of strings
$\{(b^j, b^{n-k-1-j}a^k cbba^{k-2}ca) \mid 0 \leqslant j \leqslant n-k-1\} \cup$
$\{(b^{n-k-1}a^k cb, ba^{k-2}ca)\} \cup \ \{(b^{n-k-1}a^k cbba^i, a^{k-2-i}ca) \mid 0 \leqslant i \leqslant k-2\}$.
Let us show that this set is a fooling set of size n for language $L(C_{n,\alpha})$. The string $b^{n-k-1}a^k cbba^{k-2}ca$ is accepted by nfa $C_{n,\alpha}$ since $C_{n,\alpha}$ goes to state k by $b^{n-k-1}a^k cb$, then to state 1 by b, then to state $k-1$ by a^{k-2}, and then either goes to state k or remains in state $k-1$ by c, and finally goes to the accepting state k by a. It follows that (F1) in Definition 3 holds. On the other hand, nfa $C_{n,\alpha}$ does not accept the following strings: $b^r a^k cbba^{k-2}ca$ with $r < n-k-1$, since a cannot be read in states greater than $k+1$; $b^{n-k-1}a^k cb \cdot b^r a^k cbba^{k-2}ca$, since after reading the second a^k, the nfa is in state k and cannot read c; $b^{n-k-1}a^k cbba^s \cdot b^r a^k cbba^{k-2}ca$ for the same reason; $b^{n-k-1}a^k cb \cdot a^r ca$ with $r \geqslant 0$ and $b^{n-k-1}a^k cbba^s ca$ with $s > k-2$, since c cannot be read in state k. Thus (F2) holds, and so nfa $C_{n,\alpha}$ is minimal.

Consider the corresponding subset automaton $C'_{n,\alpha}$. Each of the singleton sets $\{n\}, \{n-1\}, \ldots, \{k\}$ is reachable from the initial state $\{n\}$ by a string in b^*. Then, all β reachable states of the subset automaton $B'_{k,\beta}$ are reachable in $C'_{n,\alpha}$ as well. Moreover, no other subset of set $\{1, 2, \ldots, n\}$ is reachable. Hence nfa $C'_{n,\alpha}$ has $n-k+\beta$ reachable states. Since $n-k+\beta = \alpha$, it is enough to show that no two different reachable states are equivalent. Two different subsets of $\{1, 2, \ldots, k\}$ can be distinguished by the same string as in the dfa $B'_{k,\beta}$. If $k+1 \leqslant i < j \leqslant n$, then the string b^{i-k} distinguishes states $\{i\}$ and $\{j\}$. If S is a subset of the set $\{1, 2, \ldots, k\}$ and $k+1 \leqslant i \leqslant n$, then the string $b^{i-k-1}a^k cb$ is accepted from $\{i\}$ but not from S. This concludes our proof. $\square$

4. Conclusions

In this paper, we have shown that there are no magic numbers in the ternary case. We have described a minimal n-state nondeterministic finite automaton with a three-letter input alphabet whose equivalent minimal deterministic finite automaton has exactly α states for all integers n and α satisfying $n \leqslant \alpha \leqslant 2^n$.

The question of whether there are some magic numbers in the binary case remains open. However, after investigating the ternary case, we strongly conjecture that each value in the range from n to 2^n can be reached as the size of the minimal deterministic finite automaton equivalent to a minimal *binary* nondeterministic finite automaton of n-states.

Acknowledgments

I would like to thank Alexander Szabari for his help with the computational verification of some conjectures. I am very grateful to anonymous referees of DLT 2009 for their valuable corrections and suggestions. My thanks also go to Mikhail Volkov and Janusz Brzozowski for their careful reading of the manuscript and useful remarks that have helped me to improve the presentation of this paper.

References

[1] Aho, A.V., Ullman, J.D., and Yannakakis, M.: On notions of information transfer in VLSI circuits. In: Proc. 15th STOC, pp. 133–139 (1983)

[2] Birget, J.-C.: Intersection and union of regular languages and state complexity. Inform. Process. Lett. 43, 185–190 (1992)

[3] Birget, J.-C.: Partial orders on words, minimal elements of regular languages, and state complexity. Theoret. Comput. Sci. 119, 267–291 (1993); Erratum: Available at `http://clam.rutgers.edu/~birget/poWordsERR.ps`

[4] Chrobak, M.: Finite automata and unary languages. Theoret. Comput. Sci. 47, 149–158 (1986); Erratum: Theoret. Comput. Sci. 302, 497–498 (2003)

[5] Dassow, J., Stiebe, R.: Nonterminal complexity of some operations on context-free languages. In: Geffert, V., Pighizzini, G. (eds.) 9th International Workshop on Descriptional Complexity of Formal Systems, pp. 162–169. P.J. Šafárik University of Košice, Slovakia (2007)

[6] Geffert, V.: (Non)determinism and the size of one-way finite automata. In: Mereghetti, C., Palano, B., Pighizzini, G., Wotschke D. (eds.) 7th International Workshop on Descriptional Complexity of Formal Systems, pp. 23–37. University of Milano, Italy (2005)

[7] Geffert, V.: Magic numbers in the state hierarchy of finite automata. Inform. Comput. 205, 1652–1670 (2007)

[8] Glaister, I., Shallit, J.: A lower bound technique for the size of nondeterministic finite automata. Inform. Process. Lett. 59, 75–77 (1996)

[9] Hromkovič, J.: Communication complexity and parallel computing. Springer, Heidelberg, 1997.

[10] Iwama, K., Kambayashi, Y., Takaki, K.: Tight bounds on the number of states of DFAs that are equivalent to n-state NFAs. Theoret. Comput. Sci. 237, 485–494 (2000). Preliminary version in: Bozapalidis, S. (ed.) 3rd International Conference on Developments in Language Theory. Aristotle University of Thessaloniki (1997)

[11] Iwama, K., Matsuura, A., Paterson, M.: A family of NFAs which need $2^n - \alpha$ deterministic states. Theoret. Comput. Sci. 301, 451–462 (2003)
[12] Jirásková, G.: Note on minimal finite automata. In: Sgall, J., Pultr, A., Kolman, P. (eds.) MFCS 2001. LNCS, vol. 2136, pp. 421–431. Springer, Heidelberg (2001)
[13] Jirásková, G.: On the state complexity of complements, stars, and reversals of regular languages. In: Ito M., Toyama M. (eds.) DLT 2008. LNCS, vol. 5257, pp. 431–442. Springer, Heidelberg (2008)
[14] Jirásek, J., Jirásková, G., Szabari, A.: Deterministic blow-ups of minimal nondeterministic finite automata over a fixed alphabet. Internat. J. Found. Comput. Sci. 16, 511–529 (2005)
[15] Lyubich, Yu.I.: Estimates for optimal determinization of nondeterministic autonomous automata. Sib. Mat. Zh. 5, 337–355 (1964) (in Russian)
[16] Matsuura A., Saito, Y.: Equivalent transformation of minimal finite automata over a two-letter alphabet. In: Campeanu, C., Pighizzini, G.: (eds.) 10th International Workshop on Descriptional Complexity of Formal Systems, pp. 224–232. University of Prince Edward Island, Canada (2008)
[17] Rabin, M., Scott, D.: Finite automata and their decision problems. IBM Res. Develop. 3, 114–129 (1959)
[18] Sipser, M.: Introduction to the theory of computation. PWS Publishing Company, Boston (1997)
[19] Yu, S.: Chapter 2: Regular languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages - Vol. I, pp. 41–110. Springer, Heidelberg (1997)
[20] Zijl, L.: Magic numbers for symmetric difference NFAs. Internat. J. Found. Comput. Sci. 16, 1027–1038 (2005)