# Size of nondeterministic and deterministic automata for certain languages

Raitis Ozols, Rūsiņš Freivalds,
Laura Mančinska, Māris Ozols

sm01022@lanet.lv, rusinsf@latnet.lv,
violeta@navigator.lv, marozols@yahoo.com

Department of Computer Science, University of Latvia,
Raiņa bulvāris 29, Rīga, LV-1459, Latvia

**Abstract.** In the theory of automata the question about difference between the size of deterministic and nondeterministic automata which recognize the same language is of great importance. However, this problem has been studied mainly in case when input alphabet consists of at least 2 letters. In this paper some special kind of languages in one letter alphabet will be discussed and the estimate of the number of states required for deterministic and nondeterministic automata to accept these languages will be made. For one of these languages nondeterministic automaton with $\leq \lceil \sqrt{n} \rceil + 1$ states can be built, but for other with $\leq 1{,}8 \cdot \dfrac{\ln^2 n}{\ln \ln n} + 0{,}85 \cdot \dfrac{\ln^2 n}{(\ln \ln n)^2}$ states, where $n$ is the number of states required for the corresponding deterministic automaton.

**Conference:** FCS'05.

**Key words:** deterministic automata, nondeterministic automata, states.

## 1. Introduction

In automata theory there is a question about the difference between size (number of states) of deterministic finite automata (DFA) and nondeterministic finite automata (NFA), which recognize the same language. There are some results related to this question. The following theorem is well known: for every nondeterministic automaton with $n$ states which recognizes some language $L$ exists deterministic automaton with no more than $2^n$ states which recognizes the same language $L$. It is also known that such languages exists for which NFA with $n$ states can be built, but corresponding DFA requires exactly $2^n$ states. Such languages are known for many letter alphabets (even for one consisting of 3 letters). This problem for single letter alphabet has not been studied yet.

In this article we study NFAs with single letter alphabet. We demonstrate that such languages exists for which NFA requires respectively $\leq \lceil \sqrt{n} \rceil + 1$ and $\leq 1{,}8 \cdot \dfrac{\ln^2 n}{\ln \ln n} + 0{,}85 \cdot \dfrac{\ln^2 n}{(\ln \ln n)^2}$ states while corresponding DFA requires more than $n$ states.

## 2. Notations

DFA($L$) – the smallest possible number of states for DFA which recognizes language $L$.
NFA($L$) – the smallest possible number of states for NFA which recognizes language $L$.
$\mathbf{N} = \{1; 2; 3; \ldots\}$ – set of all natural numbers.
$\mathbf{N}_0 = \{0; 1; 2; 3; \ldots\}$ – set of all natural numbers and zero.
$P_n$ – $n$-th prime ($P_1 = 2$, $P_2 = 3$, $P_3 = 5$, etc.).

LCM($a_1, a_2, ..., a_n$) – least common multiple of natural numbers $a_1, a_2, ..., a_n$ (the smallest natural number which we can divide by any of numbers $a_1, a_2, ..., a_n$).

$a^n$ – letter $a$ repeated $n$ times ($a^0 = \varepsilon$ is empty word).

$|v|$ – length of word $v$ ($|a^n| = n$).

In this paper only languages in single letter alphabet $\Sigma = \{a\}$ will be discussed ($a$ is the only letter in alphabet). Let's define the languages we are going to study:

$\mathbf{A}_n$ ($n\in\mathbf{N}_0$) – the set of all languages $L$, which satisfy: $a^n\notin L$ and $m > n \Rightarrow a^m\in L$. Set $\mathbf{A}_0$ contains exactly one language, $\mathbf{A}_1$ – exactly two languages, etc. $\mathbf{A}_n$ contains $2^n$ languages.

$\mathbf{B}_n$ ($n\in\mathbf{N}$) – the set of all languages $L$, which satisfy: $0 \le m \le n–1 \Rightarrow a^m\in L$ and $a^n\notin L$ and $L$ is regular. For each $n$ set $\mathbf{B}_n$ is infinite.

$C_n$ ($n\in\mathbf{N}$) – language $L$, which satisfies: $\forall\ m\in\mathbf{N}_0\ m \ne n \Rightarrow a^m\in L$ and $a^n\notin L$. It is easy to see that for every $n$ language $C_n$ is unique.

## 3. Number of States Required for DFAs

**Theorem 1.** If $n\in\mathbf{N}_0$ and $L\in\mathbf{A}_n$ then DFA($L$) = $n$+2.

**Proof.** Assume, it is possible to construct a DFA, which recognizes $L\in\mathbf{A}_n$. When automaton receives words $a^0, a^1, \ldots, a^{n+1}$ as input, it reaches states $q_0, q_1, \ldots, q_{n+1}$. Let us assume that two of these states are equivalent: $\exists i, j$: $0 \le i < j \le n+1$ and $q_i = q_j$. It means, both words $a^ia^{n-i}$ and $a^ja^{n-i}$ ($a^n$ and $a^{n+j-i}$) will lead automaton to the same state. It is not possible, because $a^n\notin L$, but $a^{n+j-i}\in L$ ($n$+$j$–$i$>$n$). Therefore our assumption was wrong and automaton contains at least $n$+2 states, because $\forall i, j$ $i \ne j \Rightarrow q_i \ne q_j$ and therefore $|\{q_0; q_1; ...; q_{n+1}\}| = n+2$.

Now let us prove, that DFA($L$) $\le n$+2. It is sufficient to show for each $L\in\mathbf{A}_n$, how to construct an automaton, which recognizes $L$ and contains no more than $n$+2 states.

$L\in\mathbf{A}_0$

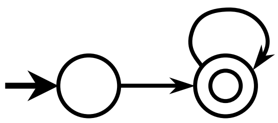

Figure 1.

$L\in\mathbf{A}_n$, $n\ge1$

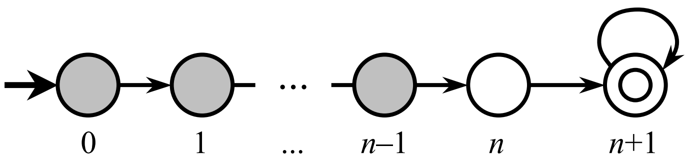


Figure 2.

If $L\in\mathbf{A}_0$, we can use automaton shown in Figure 1. If $L\in\mathbf{A}_n$ ($n\ge1$) then we can use automaton shown in Figure 2 (some of grey states must be choose as accepting, according to language $L$). Inequalities DFA($L$) $\ge n$+2 and DFA($L$) $\le n$+2 implies that DFA($L$) = $n$+2.

**Theorem 2.** If $n\in\mathbf{N}$ and $L\in\mathbf{B}_n$, then DFA($L$) $\ge n$+1.

**Proof.** Let us assume, there exists a NFA which recognizes the language $L\in\mathbf{B}_n$. When automaton receives words $a^0, a^1, \ldots, a^n$ as input, it reaches states $q_0, q_1, \ldots, q_n$. Let us assume that two of these states are equivalent: $\exists i, j$: $0 \le i < j \le n$ and $q_i = q_j$. Then words $a^ia^{n-j}$ and $a^ja^{n-j}$ ($a^{n+i-j}$ and $a^n$) will lead automaton to the same state. It is not possible, because $a^{n+i-j}\in L$ ($n$+$i$–$j$<$n$), but $a^n\notin L$. Therefore our assumption was wrong and automaton contains at least $n$+1 state, because $\forall i, j$ $i \ne j \Rightarrow q_i \ne q_j$ and therefore $|\{q_0; q_1; ...; q_n\}| = n+1$.

## 4. Number of States Required for NFAs

Now we will construct NFAs, which recognizes languages from sets $\mathbf{A}_n$ and $\mathbf{B}_n$ and show that they require significantly less states than corresponding DFAs.

**Theorem 3.** For each $n\in\mathbf{N}$ there exists language $L_n$ from set $\mathbf{A}_n$ for which it is possible to build a NFA with $k(n)=\lceil\sqrt{n}\rceil+1$ states which recognizes $L_n$.

**Proof.** Let us examine a special kind of NFA show in Figure 3. It contains $k$ states enumerated from 0 to $k$–1; $q_0$ is both – an initial and an accepting state. One can easily trace the subsets being reached by this automaton for different length of input word (Table 1). Automaton accepts word of length $n=(k–2)\cdot k+1$, but all longer words does not. Thus it recognizes language $L_{(k–2)\cdot k+1}$.

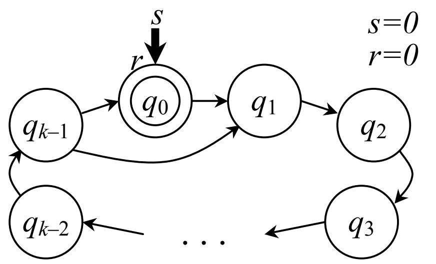

Figure 3.

Table 1.

| Length of input word | Reached states | Does NFA accept |
|---|---|---|
| 0 | 0 | yes |
| $k$ | 0,1 | yes |
| $m\cdot k$ | 0, 1, ..., $m$ | yes |
| $(k–2)\cdot k$ | 0, 1, ..., $k$–2 | yes |
| $(k–2)\cdot k+1$ | 1, 2, ..., $k$–1 | no |
| $> (k–2)\cdot k+1$ | 0, 1, ..., $k$–1 | yes |

Table 2.

| $s$ | $r$ | $n$ |
|---|---|---|
| 0 | 0 | $(k–2)\cdot k+1$ |
| 1 | 0 | $(k–2)\cdot k$ |
| $m$ | 0 | $(k–2)\cdot k+1–m$ |
| $k$–1 | 0 | $(k–3)\cdot k+2$ |
| $k$–1 | $k$–1 | $(k–3)\cdot k+1$ |
| $k$–1 | $k–m$ | $(k–3)\cdot k+2–m$ |
| $k$–1 | 3 | $(k–3)\cdot(k–1)+2$ |
| $k$–1 | 2 | $(k–3)\cdot(k–1)+1$ |

Now we will try to build similar automaton with the same number of states, but for other $n$. Let us generalize the automaton show in Figure 3, by choosing arbitrary initial state $q_s$ and arbitrary accepting state $q_r$, where $r, s\in\{0, 1, ..., k–1\}$. One can decrease $n$ for this automaton, by smoothly changing $r$ and $s$ (as shown in Table 2). As we can see $n$ can take any value from interval $\{(k–3)\cdot(k–1)+2, \ldots, (k–2)\cdot k+1\}$ for a fixed $k$ (size of automaton). There is no need to further extend the Table 2., because we have already gained $(k–3)\cdot(k–1)+1$ as the value of $n$, which equals to upper bound of the next interval (when automaton has $k$–1 state and $s = r = 0$). Thus these intervals cover all natural numbers, and for each $n$ the corresponding $k$ can be found.

In order to find $k$ for a given $n$ (determine the interval to which $n$ belongs), let us denote the interval's endpoints by $n_{min}$ and $n_{max}$. $n_{\min}=(k–3)\cdot(k–1)+2$ or $k(n_{\min})=\sqrt{n_{\min}-1}+2$. This function is monotonously increasing thus for all $n$ from the same interval the integer part of it will be the same. So we can write that $k(n)=\lfloor\sqrt{n-1}\rfloor+2$. As well as $n_{\max}=(k-2)\cdot k+1$, $k(n_{\max})=\sqrt{n_{\max}}+1$ and $k(n)=\lceil\sqrt{n}\rceil+1$. Of course it means that $k=O(\sqrt{n})$.

When we have found $k$, expressions for $r$ and $s$ are as follows: if $n\geq(k–3)\cdot k+2$ then $r$=0 and $s=(k–2)\cdot k+1–n$, otherwise (if $n<(k–3)\cdot k+2$) $r=3+n–((k–3)\cdot(k–1)+2)=1+n–(k–3)\cdot(k–1)$ and $s=k–1$.

Note: as we have shown earlier for these languages $\mathrm{DFA}(L_n)>n$, thus equivalent NFA requires significantly less states.

Now let us examine another kind of NFAs. Let Aut($a_1$, $a_2$, ..., $a_k$ | $n$) denote NFA shown in Figure 4. ($a_i \in \mathbf{N}$, $k \in \mathbf{N}$ and $n \in \mathbf{N}$). It has accepting initial state and $k$ arrows coming out of it. The $i$–th arrow points to a cycle containing $a_i$ states where $i \in \{1, 2, ..., k\}$. Therefore in total automaton has

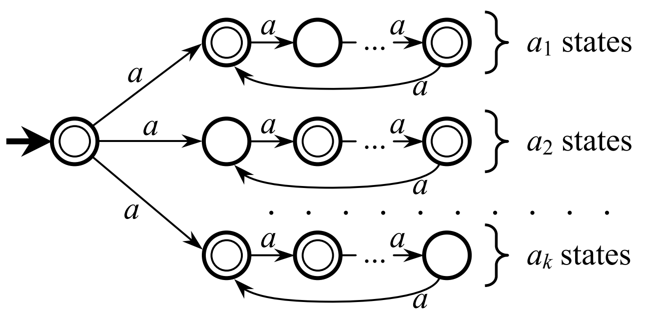


Figure 4.

$1 + a_1 + a_2 + ... + a_k$ states. In each cycle all states are accepting, except one. The nonaccepting state in the $i$-th cycle can be determined in the following way: one should determine which state in $i$-th cycle the automaton has reached after reading word $a^n$.

Now we should find out which words are recognized by the above mentioned automata Aut($a_1$, $a_2$, ..., $a_k$ | $n$) depending on the values of $a_i$ and $n$.

**Theorem 4.** An automaton Aut($a_1$, $a_2$, ..., $a_k$ | $n$) does not accept word $v$ if and only if both conditions are satisfied:

1) $|v| > 0$;

2) $(|v| - n)$ is divisible by LCM($a_1$, $a_2$, ..., $a_k$).

**Proof.** For an automaton to not accept word $v$, it is necessary that $|v| > 0$. It is also necessary for $|v| - n$ to be divisible by $a_i$ (otherwise $v$ will be accepted in cycle of length $a_i$). Of course if $a_i$=1 then $|v| - n$ is divisible by 1. Therefore $|v| - n$ is divisible by $a_1$, $a_2$, ..., $a_k$. From here follow that $|v| - n$ is divisible by LCM($a_1$, $a_2$, ..., $a_k$). It is easy to understand that these conditions are sufficient. Therefore the theorem has been proven.

**Remark.** In theorems 5, 6, 7 and 10 $P_n$ stands for $n$-th prime number.

**Theorem 5.** For every $n \in \mathbf{N}$ there exists NFA which recognize some language from set $\mathbf{B}_n$ and automata has form Aut($P_1$, $P_2$, ..., $P_k$ | $n$) where $k$ is some natural number.

**Proof.** Let's choose smallest natural number $k$ such as $P_1 \cdot P_2 \cdot ... \cdot P_k \geq n$. Product $P_1 \cdot P_2 \cdot ... \cdot P_k$ we denote with $X$. Then automaton Aut($P_1$, $P_2$, ..., $P_k$ | $n$) will recognize some language from the set $\mathbf{B}_n$. Really, LCM($P_1$, $P_2$, ..., $P_k$) = $P_1 \cdot P_2 \cdot ... \cdot P_k = X \geq n$. According to previously proven theorem automaton does not recognize word $v$ if $|v| > 0$ and $(|v| - n)$ is divisible by $X$. If for some word $v$ $|v| < n$ and $|v| - n$ is divisible by $X$ then $n - |v| \geq X \geq n \Rightarrow |v| \leq 0$. There do not exist words for which $|v| < 0$ and word $|v| = 0$ is accepted. Therefore all words which have $|v|<n$ will be accepted and word $|v| = n$ will not be accepted.

According to theorem 5 we can choose the smallest natural $k$ such as $P_1 \cdot P_2 \cdot ... \cdot P_k \geq n$. Further we will use this fact to estimate the number of states for corresponding NFA.

Further we will need the estimate of $n$-th prime. Theorem 6 provides us the needed estimate.

**Theorem 6** (Rosser, 1938). For all $n \in \mathbf{N}$ $P_n > n \cdot \ln n$ [1].

**Theorem 7.** If $n \in \mathbf{N}$, then $P_1 \cdot P_2 \cdot ... \cdot P_n > 0{,}5 \cdot n^n$.

**Proof.** With $A_k$ we denote a statement $P_1 \cdot P_2 \cdot ... \cdot P_k > 0{,}5 \cdot k^k$. Correctness of statements $A_1, A_2, ..., A_{16}$ we can check using computer. If $k \geq 16$ then according to theorem 6

$$P_k > k \cdot \ln k \geq k \cdot \ln 16 > k \cdot e\,.$$

Let us suppose that $A_k$ is true ($k \geq 16$) and prove $A_{k+1}$. $A_k \Leftrightarrow P_1 \cdot P_2 \cdot ... \cdot P_k > 0{,}5 \cdot k^k \Rightarrow P_1 \cdot P_2 \cdot ... \cdot P_{k+1} > 0{,}5 \cdot k^k \cdot (k+1) \cdot e$. Let us prove the following inequality $0{,}5 \cdot k^k \cdot (k+1) \cdot e > 0{,}5 \cdot (k+1)^{k+1}$. We can transform this inequality to $k^k \cdot e > (k+1)^k$, $e > \left(1 + \frac{1}{k}\right)^k$. Last inequality is true (it is well known from calculus). Therefore $A_k \Rightarrow A_{k+1}$. From here follow that $A_k$ is true for all $k \geq 16$. Statements $A_1, A_2, ..., A_{16}$ are also true therefore statements $A_n$ are corrects for all natural values of $n$. Thus the theorem has been proved.

**Theorem 8.** If $A > 1$ and $x = 1{,}5 \cdot \dfrac{A}{\ln A}$ then $x \ln x > A$.

**Proof.** If $x = 1{,}5 \cdot \dfrac{A}{\ln A}$ then $x \ln x = 1{,}5 \cdot \dfrac{A}{\ln A} \cdot \ln\left(1{,}5 \cdot \dfrac{A}{\ln A}\right) =$
$= 1{,}5 \cdot \dfrac{A}{\ln A} \cdot \left(\ln 1{,}5 + \ln A - \ln\ln A\right)$. We want to prove that $1{,}5 \cdot \dfrac{A}{\ln A} \cdot (\ln 1{,}5 + \ln A - \ln\ln A) > A$ (if $A > 1$). After several transformations we gain:

$$1{,}5 \cdot \ln 1{,}5 + 0{,}5 \cdot \ln A > 1{,}5 \cdot \ln\ln A$$

(for all $A > 1$). In order to prove this inequality let us substitute A by $e^{3x}$ where $x > 0$. Then we must prove inequality $1{,}5 \cdot \ln 1{,}5 + 0{,}5 \cdot 3x > 1{,}5 \cdot \ln(3x)$ or $\ln 1{,}5 + x > \ln 3 + \ln x$ for all $x > 0$. Last inequality we can transform to

$$(\ln 1{,}5 - \ln 3 + 1) + x - 1 > \ln x \text{ and } 0{,}3068... + x - 1 > \ln x.$$

Last inequality is correct because from calculus it is known that $x > 0 \Rightarrow x - 1 \geq \ln x$.

Now it is possible to conclude that if for a given $n$ we need to find smallest $k$ such as the product of first $k$ primes is greater or equal to $n$ then we can look for $k$ such as $0{,}5 \cdot k^k \geq n$. Thus $k^k \geq 2n$ and $k \ln k \geq \ln(2n)$. According to theorem 8: $k \geq 1{,}5 \cdot \dfrac{\ln(2n)}{\ln\ln(2n)}$, $k = \left\lceil 1{,}5 \cdot \dfrac{\ln(2n)}{\ln\ln(2n)} \right\rceil$ $(n \geq 2)$.

**Theorem 9.** There exists $n_1 \in \mathbf{N}$ such as $n \geq n_1 \Rightarrow \left\lceil 1{,}5 \cdot \dfrac{\ln(2n)}{\ln\ln(2n)} \right\rceil < 1{,}6 \cdot \dfrac{\ln n}{\ln\ln n}$.

**Proof.** Using fact that for every real $a$ $\lceil a \rceil < a + 1$ we conclude that $\left\lceil 1{,}5 \cdot \dfrac{\ln(2n)}{\ln\ln(2n)} \right\rceil < 1{,}5 \cdot \dfrac{\ln(2n)}{\ln\ln(2n)} + 1$ and for sufficiently large $n$ $\ln\ln(2n) > \ln\ln n$ therefore

$$\left\lceil 1{,}5 \cdot \frac{\ln(2n)}{\ln\ln(2n)} \right\rceil < 1{,}5 \cdot \frac{\ln(2n)}{\ln\ln(2n)} + 1 < 1{,}5 \cdot \frac{\ln(2n)}{\ln\ln n} + 1.$$

Let us show that if $n$ is sufficiently large then $1{,}5 \cdot \dfrac{\ln(2n)}{\ln\ln n} + 1 < 1{,}6 \cdot \dfrac{\ln n}{\ln\ln n}$ (it is obvious that the needed inequality follows from this inequality). The above shown inequality can be transformed in the following way $1{,}5 \cdot \ln(2n) + \ln\ln n < 1{,}6 \cdot \ln n$, $1{,}5 \cdot \ln 2 + 1{,}5 \cdot \ln n + \ln\ln n < 1{,}6 \cdot \ln n$, $1{,}5 \cdot \ln 2 + \ln\ln n < 0{,}1 \cdot \ln n$. After substituting $n$ by $e^{20x}$ we gain that $1{,}5 \cdot \ln 2 + \ln 20 + \ln x < x + x$, or $3{,}0354... + 1 + \ln x < x + x$. It can be seen that if $x > 4$ then

$$3{,}0354... < x \ \& \ 1 + \ln x \leq x \ \Rightarrow \ 3{,}0354... + 1 + \ln x < x + x.$$

If $x$ is sufficiently large then this inequality is true. Thus it can be concluded that for sufficiently large $n$ also the initial inequality holds.

**Theorem 10.** There is number $n_2 \in \mathbf{N}$ such as $n \geq n_2 \Rightarrow 1 + (P_1 + P_2 + ... + P_n) < 0{,}7 \cdot n^2 \ln n$.

**Proof.** In order to prove this theorem we will use the result $P_1 + P_2 + ... + P_n \sim \dfrac{1}{2} n^2 \ln n$ $(n \to \infty)$ (proved by Bach and Shallit, 1996 [2]). From this statement immediately follows that

$$\lim_{n\to\infty} \frac{P_1 + P_2 + ... + P_n}{0{,}5 \cdot n^2 \ln n} = 1 \quad \text{and} \quad \lim_{n\to\infty} \frac{1 + (P_1 + P_2 + ... + P_n)}{0{,}5 \cdot n^2 \ln n} = 1.$$

Therefore we can find natural number $n_2$ that $n \geq n_2 \Rightarrow \dfrac{1 + (P_1 + P_2 + ... + P_n)}{0{,}5 \cdot n^2 \ln n} < 1{,}4$. This statement is equivalent to the following statement: $\exists\, n_2 \in \mathbf{N}$ such that $n \geq n_2 \Rightarrow$ $1 + (P_1 + P_2 + ... + P_n) < 0{,}7 \cdot n^2 \ln n$ which we wanted to prove.

**Theorem 11.** There exists $n_0 \in \mathbf{N}$ such as for all $n \geq n_0$ NFA that recognizes some language from the set $\mathbf{B}_n$ with no more than $1{,}8 \cdot \frac{\ln^2 n}{\ln \ln n} + 0{,}85 \cdot \frac{\ln^2 n}{(\ln \ln n)^2}$ states can be built.

**Proof.** For sufficiently large $n$ automaton Aut($P_1$, $P_2$, ..., $P_k$ | $n$) where $k = \left\lceil 1{,}5 \cdot \frac{\ln(2n)}{\ln \ln(2n)} \right\rceil$ recognizes a language from $\mathbf{B}_n$. This automaton has $1 + P_1 + P_2 + ... + P_k$ states. If $k \geq n_2$ then number of states satisfies the following inequality:

$$S = 1 + \sum_{i=1}^{k} P_i < 0{,}7 \cdot k^2 \ln k = 0{,}7 \cdot \left\lceil 1{,}5 \cdot \frac{\ln(2n)}{\ln \ln(2n)} \right\rceil^2 \cdot \ln \left\lceil 1{,}5 \cdot \frac{\ln(2n)}{\ln \ln(2n)} \right\rceil.$$

If $n \geq n_1$ and $k \geq n_2$ then

$$S < 0{,}7 \cdot \left( 1{,}6 \cdot \frac{\ln n}{\ln \ln n} \right)^2 \cdot \ln \left( 1{,}6 \cdot \frac{\ln n}{\ln \ln n} \right) = 0{,}7 \cdot 2{,}56 \cdot \frac{\ln^2 n}{(\ln \ln n)^2} \cdot \left( \ln 1{,}6 + \ln \ln n - \ln \ln \ln n \right),$$

$$S < 0{,}7 \cdot 2{,}56 \cdot \frac{\ln^2 n}{\ln \ln n} + 0{,}7 \cdot 2{,}56 \cdot \ln 1{,}6 \cdot \frac{\ln^2 n}{(\ln \ln n)^2}.$$

Using inequalities $0{,}7 \cdot 2{,}56 = 1{,}792 < 1{,}8$ and $0{,}7 \cdot 2{,}56 \cdot \ln 1{,}6 = 0{,}842... < 0{,}85$ we gain inequality

$$S < 1{,}8 \cdot \frac{\ln^2 n}{\ln \ln n} + 0{,}85 \cdot \frac{\ln^2 n}{(\ln \ln n)^2}$$

which holds for sufficiently large $n$. Thus such $n_0$ mentioned in theorem 11 can be found that for all $n \geq n_0$ this inequality holds.

**Remark.** Theorems 4, 5, ..., 11 was proved by R.Ozols.

**Theorem 12.** For every $n \geq n_0$ there exists NFA, which recognizes language $C_n$ and has at most $\lceil \sqrt{n} \rceil + 1{,}8 \cdot \frac{\ln^2 n}{\ln \ln n} + 0{,}85 \cdot \frac{\ln^2 n}{(\ln \ln n)^2} + 1$ states.

**Proof.** Such automaton can be constructed by combining NFAs mentioned in proofs of theorems 3 and 4. It is easy to see that all words $v$ such $|v| < n$ or $|v| > n$ will be accepted. If $|v| = n$ then $v$ will not be accepted because none of the automata accept it. Thus automaton constructed will have at most $\lceil \sqrt{n} \rceil + 1{,}8 \cdot \frac{\ln^2 n}{\ln \ln n} + 0{,}85 \cdot \frac{\ln^2 n}{(\ln \ln n)^2} + 1$ states. This estimation is gained by adding estimations of number of states for both automatons (from theorems 3 and 4).

## 5. References


[1] http://mathworld.wolfram.com/RossersTheorem.html (Rosser)

[2] http://mathworld.wolfram.com/PrimeSums.html (Bach and Shallit)

[3] Andris Ambainis, Rusins Freivalds: 1-way quantum finite automata: strengths, weaknesses and generalizations CoRR quant ph/9802062: (1998)

[4] Andris Ambainis, Richard F. Bonner, Rusins Freivalds, Marats Golovkins, Marek Karpinski: Quantum Finite Multitape Automata. SOFSEM 1999: 340 348

[5] Wilfried Brauer: On Minimizing Finite Automata. Bulletin of the EATCS 35: 113 116 (1988)

[6] Rusins Freivalds: Probabilistic Machines Can Use Less Running Time. IFIP Congress 1977: 839 842

[7] Rusins Freivalds: An answer to an open problem. Bulletin of the EATCS 23: 31 32 (1984)

[8] Eitan Gurari. An Introduction to the Theory of Computation. Computer Science Press, 1989.