

## Experiments on Synchronizing Automata

ADAM ROMAN

Institute of Computer Science, Jagiellonian University  
Prof. Stanisława Łojasiewicza 6, 30-348 Cracow, Poland  
e-mail: [roman@ii.uj.edu.pl](mailto:roman@ii.uj.edu.pl)

**Abstract.** This work is motivated by the Černý Conjecture – an old unsolved problem in the automata theory. We describe the results of the experiments on synchronizing automata, which have led us to two interesting results. The first one is that the size of an automaton alphabet may play an important role in the issue of synchronization: we have found a 5-state automaton over a 3-letter alphabet which attains the upper bound from the Černý Conjecture, while there is no such automaton (except Černý automaton  $\mathcal{C}_5$ ) over a binary alphabet. The second result emerging from the experiments is a theorem describing the dependencies between the automaton structure  $S$  expressed in terms of the so-called merging system and the maximal length of all minimal synchronizing words for automata of type  $S$ .

### 1. Introduction and motivation

We define *an automaton* as a triple  $\mathcal{A} = (Q, A, \delta)$ , where  $Q$  is a finite set of states,  $A$  is an alphabet and  $\delta : Q \times A \rightarrow Q$  is a transition function defining the action of an automaton. We can interpret the automaton as a directed labelled graph in which vertices represent states and labelled arrows represent  $\delta$ . By  $A^*$  we denote the set of all words over  $A$ . A transition function can be naturally extended on the set  $2^Q \times A^*$ :

$$\forall P \subseteq Q, \forall a \in A, \forall w \in A^* \quad \delta(P, aw) = \bigcup_{p \in P} \delta(\delta(p, a), w). \quad (1)$$

In the following we deal only with deterministic, strongly connected and complete automata. This means that  $\delta$  is a total function and the following condition holds:

$$\forall p, q \in Q \exists w \in A^* : \delta(p, w) = q.$$

If there is a word  $w \in A^*$  such that for all states  $p, q \in Q$  we have  $\delta(p, w) = \delta(q, w)$  (or, equivalently,  $|\delta(Q, w)| = 1$ ), then we say that  $w$  *synchronizes*  $\mathcal{A}$  and we call  $\mathcal{A}$  a *synchronizing automaton*. If there is no shorter word, we call  $w$  the *minimal synchronizing word* for  $\mathcal{A}$ . We denote by  $m(\mathcal{A})$  the length of a minimal synchronizing word for  $\mathcal{A}$ . The set of all synchronizing words (not necessarily minimal) is denoted by  $S(\mathcal{A})$ . In the following we use the simplified notion for  $\delta$  definition. If  $|Q| = n, |A| = m$  and  $\delta(i, j) = q_i^j$  for  $i = 1, 2, \dots, n, j = 1, 2, \dots, m$ , then we write  $\delta = (q_1^1 \dots q_n^1) \dots (q_1^m \dots q_n^m)$ .

Let  $\mathcal{A} = (Q, A, \delta)$ . If  $|Q| = n$ , then we say that  $\mathcal{A}$  is an  $n$ -state automaton. The famous Černý Conjecture [5] states that if  $\mathcal{A}$  is an  $n$ -state synchronizing automaton, then  $m(\mathcal{A}) \leq (n - 1)^2$ . The Conjecture was stated in 1964 and is still open. Černý proved that for each  $n$  there exists an  $n$ -state automaton with a minimal synchronizing word of length exactly  $(n - 1)^2$ , therefore the upper bound from the Conjecture can be reached for each  $n \geq 1$ . These special automata are so-called Černý automata. By  $\mathcal{C}_n$  we denote an  $n$ -state Černý automaton.

The Černý Conjecture seems to be only a nice combinatorial puzzle, but in fact the synchronization theory has many important applications in industry (particular in so-called ‘part orienters’), bio-computing (the reset problem), network theory, theory of codes etc. Therefore, the problem is of general interest.

The synchronization problem can be easily formulated as a pure mathematical problem from the semigroup theory. Notice that each  $a \in A$  represents the function  $f_a : Q \rightarrow Q$ . All  $f_a$ ’s, together with the composition operation form a semigroup  $S = (F, \circ)$  of functions generated by  $\{f_a : a \in A\}$ . Finding the minimal synchronizing word in a finite automaton is equivalent to the problem of finding the minimal-length composition  $f = f_{i_1} \circ f_{i_2} \circ \dots \circ f_{i_k}$  of elements from  $S$ , such that  $f$  is a constant map (see [20]).

The reader is referred to [2, 4, 12, 14, 18] for more details on the role of the synchronization problem and to [8, 14, 19, 22] for polynomial algorithms for finding the shortest possible synchronizing words. The paper [8] contains also the NP-completeness proof for the problem of finding the synchronizing word of a given length. The Černý Conjecture turned to be true for some special cases (see for example [2, 3, 6, 7, 8, 11, 22]) but in general case it is still open. The best known upper bound for  $m(\mathcal{A})$ , where  $\mathcal{A}$  has  $n$  states, is  $\frac{n^3 - n}{6}$  [9, 13, 16].

The rest of the paper is organized as follows. In Section 2 we introduce

the notion of merging states and merging systems for finite automata. In Section 3 we explain why we refine the set of analyzed automata only to these of particular merging types. This is done due to some heuristic reasoning and also in order to shorten the computing time. Section 4 contains the experimental results and the theorem emerged from the experiments, describing the dependencies between some merging types and the  $m(\mathcal{A})$  values for  $\mathcal{A}$  being of that types. In Section 5 we show that the size of an alphabet may play an important role in the synchronization theory – we give an example of a 5-state automaton (found by the computer) over a 3-letter alphabet, which does not contain  $\mathcal{C}_5$  as its subautomaton and attains the upper bound from the Černý Conjecture. We also discuss the importance of this result. The source code of the application, its executable version and the detailed results of computations can be found in [17].

## 2. Classification of finite automata

Let  $\mathcal{A} = (Q, A, \delta)$  be an automaton. By  $\delta_a^{-1}(q)$  we denote the set of states incoming to  $q$  under the letter  $a$ :  $\delta_a^{-1}(q) = \{p \in Q : \delta(p, a) = q\}$ .

DEFINITION 1. We say that  $q \in Q$  is the *merging state* of degree  $k$  for  $a \in A$ , if  $|\delta_a^{-1}(q)| = k$ .

Notice that it is possible for a state to be of degree 0. The set  $M = \delta_a^{-1}(q)$  is called the *merging system* (for  $a$ , of  $q$  and of degree  $k$ ) and  $q$  is called the merging state for  $M$ . The merging system  $M$ , such that  $|M| > 1$ , will be called a *proper* one.

Let  $M = \delta_a^{-1}(q)$  for a given  $q \in Q$  and  $a \in A$ . Note that two situations can take place:  $q \in M$  or  $q \notin M$ . We will distinguish them: if  $q \in M$  (resp.  $q \notin M$ ), then  $q$  will be called the *internal merging state* (resp. *external merging state*). Let  $q$  be the external merging state and let  $P = \{p_1, \dots, p_k\}$  be its merging system for  $a \in A$ . Then  $q$  may belong to another merging system  $R = \{r_1, \dots, r_t\}$  ( $P \cap R = \emptyset$ ) for the same letter  $a$  but it cannot be the merging state for  $R$ . In other words – if  $s$  is a merging state for  $R$ , then  $q \neq s$  only if  $q \in R$ . This property flows directly from the definition of the merging state and from the automaton determinism: each state, for a given letter, can be a merging state only for one merging system (see Propositions 1 and 2). In the following we assume that the merging systems will always be considered for a fixed letter  $a \in A$ , but first let us introduce the generalized notion of a merging type.

DEFINITION 2. Let  $\mathcal{A} = (Q, A, \delta)$ , where  $|Q| = n$ ,  $|A| = m$ , has  $\lambda_t^i$  external merging systems of degree  $t$  and  $\mu_t^i$  internal merging systems of degree  $t$  for  $a_i$ ,  $i = 1, \dots, m$ . Let the maximal degrees of these merging systems do not exceed  $k_i$ . Then such automaton will be described in the following way:

$$\mathcal{A} \sim [1^{\lambda_1^1} 1_*^{\mu_1^1} \dots k_1^{\lambda_{k_1}^1} k_{1*}^{\mu_{k_1}^1}]_{a_1} \dots [1^{\lambda_1^m} 1_*^{\mu_1^m} \dots k_m^{\lambda_{k_m}^m} k_{m*}^{\mu_{k_m}^m}]_{a_m}.$$

The above notation will be called a *merging type* of  $\mathcal{A}$ .

We will use the following convention: if there are not any proper merging systems for a given letter or if the merging system is of degree 1, this part of notation will be omitted. This convention will be used if there is no need to distinguish external and internal merging systems of degree 1. For example, we will write  $[2_*^3 3_*^2 3_*^1]_a$  instead of  $[1^3 1_*^2 2_*^3 3_*^2 3_*^1]_a [1^{12} 1_*^8]_b$ .

The following propositions are true. The first one shows that each state is involved in exactly one merging system, the second one gives the relation between a transition function and a merging system in terms of the so-called letter deficiency [1] (for the sake of simplicity we give here the version of the merging for one letter only).

PROPOSITION 1. If  $[1^{\lambda_1} 1_*^{\mu_1} \dots k^{\lambda_k} k_*^{\mu_k}]_a$  is the merging type for an  $n$ -state automaton  $\mathcal{A} = (Q, A, \delta)$ , then  $\sum_{i=1}^k i \cdot (\lambda_i + \mu_i) = |Q| = n$ .

PROPOSITION 2. Let  $\mathcal{A} = (Q, A, \delta)$  be of type  $[1^{\lambda_1} 1_*^{\mu_1} \dots k^{\lambda_k} k_*^{\mu_k}]_a$ . Then  $|Q| - |\delta(Q, a)| = \sum_{i=2}^k (i-1)(\lambda_i + \mu_i)$ .

### 3. Merging states and synchronizing words

In the next section we present the results of computer experiments, which aim was to compute the longest word among the minimal synchronizing words for automata of certain merging types. We are interested only in automata with one merging system, because this is probably the sufficient condition which should be fulfilled by an  $n$ -state automata in order to find the minimal synchronizing word of the maximal length for a given  $n$ . Namely, we strongly believe in the following conjectures.

CONJECTURE 1. If  $\mathcal{A}$  is an  $n$ -state synchronizing automaton with  $k$  merging systems  $S_1, S_2, \dots, S_k$  ( $k > 1$ ) of degree greater than 1, then there

exists an  $n$ -state synchronizing automaton  $\mathcal{B}$  with  $l < k$  merging systems  $S_{i_1}, S_{i_2}, \dots, S_{i_l}$  (being the  $l$ -subset of  $\{S_1, \dots, S_k\}$ ), such that  $m(\mathcal{A}) \leq m(\mathcal{B})$ .

**CONJECTURE 2.** If  $\mathcal{A}$  is an  $n$ -state synchronizing automaton with a merging system of degree  $k > 2$ , then there exists a synchronizing  $n$ -state automaton  $\mathcal{B}$  with a merging system of degree  $l < k$  such that  $m(\mathcal{A}) \leq m(\mathcal{B})$ .

According to these two conjectures, the automata considered in the experiment are intuitively “difficult” to synchronize: one of the input letters merges a small number of states while all other input letters simply permute the states.

It intuitively seems that the first conjecture should be true. Notice that there are automata with more than one proper merging systems, for which the  $m()$  value is greater than  $m(\mathcal{B})$ , where  $\mathcal{B}$  is an automaton with fewer proper merging systems. For example,  $\mathcal{A} = (\{1, 2, 3, 4, 5\}, \{a, b, c\}, \delta)$  defined by the transition function  $\delta = (23111)(23114)(21453)$  has two external merging systems of degrees 2 and 3 and  $m(\mathcal{A}) = 11$ , while  $\max_{\mathcal{B}} m(\mathcal{B}) = 9$ , where each  $\mathcal{B}$  has only one proper external merging system of degree 3 (see Tab. 1). But it does not refute Conjecture 1: for  $\mathcal{B}$  we can choose the external merging system of degree 2. In such class of 5-state automata there exists an automaton with the minimal synchronizing word of length 15 (see Tab. 1).

Computations support the second conjecture: for 4, 5 and 6-state automata with one merging system  $P$  of degree greater than 2 the computer found that for all of them, if  $q^*$  is the merging state and  $R = \{q \in Q : \delta_a^{-1}(q) = \emptyset\}$ , then there always exist  $p \in P, r \in R$  such that if we change the value  $\delta(p, a)$  from  $q^*$  into  $r$ , then the length of a minimal synchronizing word for the new automaton is greater than the length of such a word for the initial automaton. This method fails for  $n = 7$ , but Conjecture 2 is still true (the other transformation  $\delta$  must be used). If the above conjectures are true, then in order to prove the Černý Conjecture it would be enough to prove it only for automata with some special, simple merging type:

**PROPOSITION 3.** If Conjectures 1 and 2 are true, then Černý Conjecture holds iff  $m(\mathcal{A}) \leq (n - 1)^2$  for each  $\mathcal{A}$  of merging type [2] or [2\*].

*Proof.* Let  $\mathcal{A}_k$  be an automaton with  $k$  merging systems. Then, according to Conjecture 1, we can build the sequence of automata  $\mathcal{A}_k, \mathcal{A}_{k-1}, \dots, \mathcal{A}_1$  such that  $\mathcal{A}_i$  has  $i$  merging systems and  $m(\mathcal{A}_i) \geq m(\mathcal{A}_j)$  for  $i < j$ . Then, applying the Conjecture 2 to the automaton  $\mathcal{A}_1 = \mathcal{A}_1^t$  with one merging system of degree  $t$  we can again build the sequence  $\mathcal{A}_1^t, \dots, \mathcal{A}_1^2$  such that  $\mathcal{A}_1^i$  has one merging system of degree  $i$  and  $m(\mathcal{A}_1^i) \geq m(\mathcal{A}_1^j)$  for  $i < j$ . For a given  $\mathcal{A}_k$  we have found an automaton  $\mathcal{A}_1^2$  such that  $m(\mathcal{A}_1^2) \geq m(\mathcal{A}_k)$ .

Tab. 1. Results for certain merging types

Merg.	2	3	4	5	6	7	8	9
2*	1,1,1	4,4,4	9,9,9	16,16,16	25,25	36,36	49	64
2		4,4,4	9,9,9	<b>15,16</b> ,16	25,25	32,32	44	58
3*		1,1,1	4,4,4	9,9,9	16,16,16	25,25	36	49
3			4,4,4	9,9,9	17,17,17	25,25	33	44
4*			1,1,1	4,4,4	9,9,9	16,16	25	36
4				4,4,4	10,10,10	<b>18,20</b>	28	37
5*				1,1,1	4,4,4	9,9	16	25
5					4,4,4	11,11	21	31
6*					1,1,1	4,4,4	9,9	16
6						4,4,4	12,12	22
7*						1,1,1	4,4,4	9
7							4,4,4	13
8*							1,1,1	4,4,4
8								4,4,4
9*								1,1,1

#### 4. Experiments and results

In this section we present the experimental results on the lengths of the minimal synchronizing words for automata with one merging system, but of different degrees. The computations were done for  $2 \leq n \leq 9$ ,  $2 \leq |A| \leq 4$  and for the merging types  $[2], [2_*], \dots, [(n-1)], [(n-1)_*]$ , where  $n$  is the number of states and  $|A|$  is the alphabet size. For a given  $n$ ,  $A$  and proper merging system  $M$  all  $n$ -state automata over  $A$  with one proper merging system  $M$  were generated. For each of them the minimal synchronizing word was computed and, as the result, the length of the longest one was returned. We denote this value by  $\mathcal{M}(T)$ , where  $T$  is a merging type.

In Tab. 1 the  $\mathcal{M}(T)$  values for automata of certain classes are shown. These classes are defined by the merging types. Each table cell has one, two or three values. These are the experiment results for different alphabet sizes: the first (resp. second, third) one is the result for a binary (resp. 3-letter, 4-letter) alphabet. In some cells the values for a 3 or 4-letter alphabet are missing. This is due to the very long time needed for the computations to be finished. We discuss below the experimental results, emphasizing the role of an alphabet size in the synchronization process.

In most cases there is no difference between the values for various alphabet

sizes. Some irregularities were bolded. Special attention should be paid to the bolded values 15, 16 for  $n = 5$  and merging type [2]. We will discuss this case in the next section.

The following propositions are true for  $n$ -state automata with arbitrary alphabets. They establish the 'boundary' values from the Tab. 1.

PROPOSITION 4.  $\mathcal{M}([n_*]) = 1$ .

PROPOSITION 5.  $\mathcal{M}([n - 1]) = 4$ .

PROPOSITION 6.  $\mathcal{M}([(n - 1)_*]) = 4$ .

Proposition 4 is trivial. We omit the very easy proofs for Propositions 5 and 6.

In Tab. 1 there is also a regularity for merging types  $[n - 2]$ . The following theorem says that this dependence indeed holds for each  $n$ :

THEOREM 1. For the family of  $n$ -state automata of type  $[n - 2]$  over a binary alphabet the following equality holds:

$$\mathcal{M}([n - 2]) = \begin{cases} 9 & \text{for } |Q| = 4 \\ n + 4 & \text{for } |Q| \geq 5. \end{cases}$$

The proof (very technical) of the nontrivial equality from Theorem 1 is given in the Appendix. It is a well-known fact ([13], Corollary 3) that for an  $n$ -state synchronizing automaton and for an arbitrary  $m$ -subset of  $Q$  there exists a word  $w$  of length  $\binom{n-m+2}{2}$  such that  $|\delta(P, w)| < |P|$ . Notice, that in the synchronizing automaton of type  $[n - 2]$  there exists  $a$  such that  $|Q.a| = 3$ . Using the above fact we could bound the length of the minimal synchronizing word:  $\mathcal{M}([n - 2]) \leq 1 + \binom{n-3+2}{2} + \binom{n-2+2}{n} = \frac{n^2-3n+6}{2}$ . Theorem 1 states, that this bound can be reduced into a linear one.

There are two important reasons for introducing the notion of a merging type. They are closely related with the  $\mathcal{A}_5$  automaton described in the next section:

- the notion of a merging type allows us to reduce the computation time. Instead of doing the computations for each automaton and checking the type for each of them it is enough to fix an automaton type and perform the computations only for all automata of this type (these automata can be generated in a simple way) with no need to check the type for each automaton;
- the notion of a merging type allows us to distinguish the Černý automata from the non-Černý ones – in fact this allowed us to find the

important  $\mathcal{A}_5$  automaton – its type is  $[2_*]$ , while all Černý automata are of type  $[2]$  (see the next section).

## 5. Automaton $\mathcal{A}_5$

In Section 1 it was told that for each  $n \geq 2$  there exists an  $n$ -state Černý automaton  $\mathcal{C}_n$  for which the conjectured upper bound  $(n-1)^2$  is attained. The transition function for  $\mathcal{C}_n$ , assuming  $Q = \{0, 1, \dots, n-1\}$  and  $A = \{a_0, a_1\}$ , is defined as follows:

$$\delta(q, a) = \begin{cases} q+1 \pmod{n} & \text{for } a = a_0, \\ q & \text{for } a = a_1 \wedge q \neq n-1, \\ 0 & \text{for } a = a_1 \wedge q = n-1. \end{cases}$$

We say that an  $n$ -state automaton  $\mathcal{A}$  is *C-free* if  $\mathcal{C}_n$  is not a subautomaton of  $\mathcal{A}$ . For  $|A| = 2$  it just means that  $\mathcal{A}$  is not isomorphic with  $\mathcal{C}_n$  (up to alphabet relabeling) and for  $|A| > 2$  – that for each  $a, b \in A$ ,  $\mathcal{A}|_{\{a,b\}}$  is not isomorphic with  $\mathcal{C}_n$ , where  $\mathcal{A}|_B$  denotes  $\mathcal{A}$  with the alphabet restricted to  $B \subset A$ .

It is an interesting fact that for some  $n$  there exist C-free automata (over a binary alphabet) with the minimal synchronizing words of length  $(n-1)^2$ , but it is hard to find the examples. Till now we know only 8 C-free automata [21].

For a binary alphabet it is easy to find the examples of C-free automata for  $n = 2, 3$ ; for  $n = 4$  the example was given by Černý himself in [6]; there is no such automaton for  $n = 5$ ; for  $n = 6$  the example was given by J. Kari [10].

In Fig. 1 a 5-state C-free automaton  $\mathcal{A}_5 = (Q_{\mathcal{A}_5}, A_{\mathcal{A}_5}, \delta_{\mathcal{A}_5})$  over the 3-letter alphabet is shown. It can be verified that the minimal synchronizing word for  $\mathcal{A}_5$  is  $w = abcacacbcaacabca$  and  $|w| = 16$ , so the upper bound  $(n-1)^2 = (5-1)^2 = 16$  is reached. The proof, in the form of the power-set automaton  $\mathcal{P}(\mathcal{A}_5)$  for  $\mathcal{A}_5$ , is shown in Fig. 2 (for the sake of simplicity only ‘forward’ arrows are presented). States of  $\mathcal{P}(\mathcal{A}_5)$  are identified with the subsets of  $Q_{\mathcal{A}_5}$  and the transition function is defined in a natural way (see equation (1)). The following lemma is a classical result [6] establishing the connection between automaton synchronization and the existence of some paths in its power automaton:



LEMMA 1.  $\mathcal{A}$  is synchronizing iff there exists a path in  $\mathcal{P}(\mathcal{A})$  leading from the state  $\{Q\}$  to the state labelled by a one-element set.

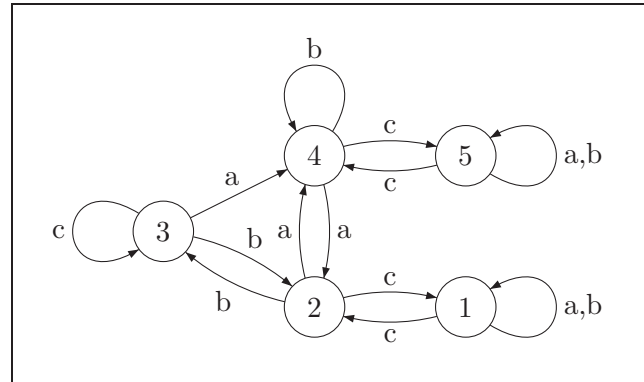


Fig. 1. Automaton  $\mathcal{A}_5$

Let  $l$  be the labelling function for the power automaton's edges. If  $e_1, e_2, \dots, e_t$  is the shortest path from  $Q$  to some "singleton" state  $\{q\}$ , then  $l(e_1)l(e_2)\dots l(e_t)$  is the minimal synchronizing word for  $\mathcal{A}$ . In Fig. 2 this path is bolded.

Let us now define  $m(n, l)$  as the maximal value among the lengths of minimal synchronizing words for  $n$ -state C-free automata over an  $l$ -letter alphabet. Using this notion we have:

THEOREM 2.  $m(3, 2) = 4$ ,  $m(4, 2) = 9$ ,  $m(5, 2) = 15$ ,  $m(6, 2) = 25$ .

These values were found by the computer in the experiment described in Section 4. The second and the fourth statement of Theorem can also be found in [10] and [6], resp. The result from Section 3 can be also formulated in the terms of  $m$ :

THEOREM 3.  $m(5, 3) = 16$ .

Notice that the computer examined *all* 5-states automata over a 3-letter alphabet and each of them had a minimal synchronizing word of length not exceeding 16, so we can put an equality in Theorem 3 (finding the automaton  $\mathcal{A}$  for which  $m(\mathcal{A}) = k$  allows us to claim only that  $m(5, 3) \geq k$ ).

The following relation holds:

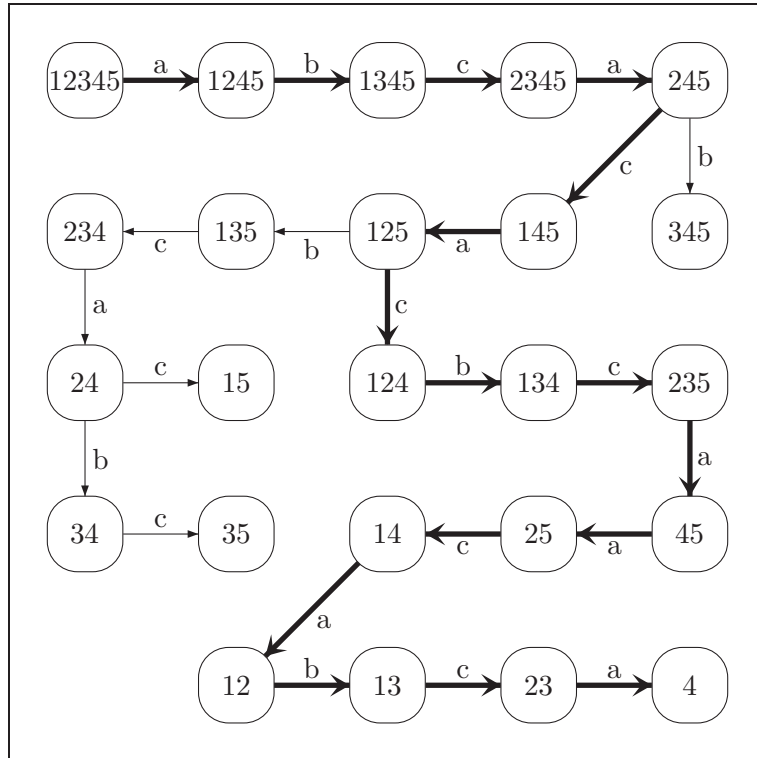


Fig. 2. Power-set automaton  $\mathcal{P}(\mathcal{A}_5)$

PROPOSITION 7.

$$l_1 < l_2 \implies m(n, l_1) \leq m(n, l_2).$$

The relation is obvious: one can take an  $l_1$ -letter automaton  $\mathcal{A} = (Q, A, \delta)$  with a minimal synchronizing word  $w$  of length  $m(n, l_1)$  and for the new  $l_2 - l_1$  letters put  $\delta(q, a_i) = \delta(Q, w)$ ,  $i = l_1 + 1, \dots, l_2$ . The new automaton over an  $l_2$ -letter alphabet possesses the same minimal synchronizing word as the initial automaton with an  $l_1$ -letter alphabet.

Theorem 3 and the fact that there is no 5-state C-free automaton over binary alphabet attaining the upper bound from the Černý Conjecture show that in some cases increasing the alphabet size results in increasing the lengths of minimal synchronizing words for C-free automata with the same number of states. Therefore in some cases the relation from Proposition 7 is sharp.

It seems to us that the alphabet size plays an important role in the synchronization problem. It is quite possible that for some  $n$ -state automaton

with the large set of states and the alphabet large enough (and, therefore, with a complex internal structure) its minimal synchronizing word could be of the length greater than  $(n - 1)^2$ , which would disprove the Černý Conjecture. Therefore, if one would like to try to find a counterexample for the Černý Conjecture, she/he probably should search it in the class of automata over large alphabets and with few small merging systems.

Moreover, it seems that the ‘exotic’ examples of automata like  $\mathcal{A}_5$  have some interesting properties in synchronization issues. For example, the Kari’s automaton was used as the counterexample to the generalized version of the Černý Conjecture proposed by Pin [15, 16]. Their existence, in view of the alphabet size, could also be an argument against the conjecture.

Till now there has been no research on the role of the alphabet size in the synchronization problem. We think that further studies in this field should be done. It can lead us to new facts about synchronization, discover some wide class of automata fulfilling the Černý Conjecture, reduce the conjecture to the simpler problem or even prove or disprove it.

## 6. Proof of Theorem 1

Let  $\mathcal{A} = (Q, A, \delta)$  be of type  $[n - 2]_a$  and let  $Q = T \cup \{p, q\}$ , where  $p$  is an external merging state of degree  $n - 2$  (for  $a$ ) and  $T$  is its merging set, i.e.  $|T| = n - 2$  and  $\delta(T, a) = p$ . For the sake of simplicity we will write  $p.a$  instead of  $\delta(p, a)$ . Consider the following cases: 1)  $p.a \in T \wedge q.a = q$ ; 2)  $p.a = q \wedge q.a \in T$ ; 3)  $p.a \in T \wedge q.a \in T$ . Because of the  $[n - 2]_a$  type of  $\mathcal{A}$  no other possibility of transforming  $p$  and  $q$  with  $a$  is allowed.

All possible cases are shown in Fig. 3. Other cases are excluded because of the automaton type or the synchronizing property. State  $T$  represents schematically the group of  $n - 2$  merging states for  $p$ . An arrow leading from  $r \notin T$  to  $T$  means that  $r$  is transformed into one of the  $n - 2$  merging states for  $p$ . By  $t_i$  we denote the states from  $T$ .

*Case 1a).* To synchronize some states into a two-element set we have to use the word  $aba$  (if  $p.ab \neq q$ ) or  $abba$  (if  $p.ab = q$ ). We have  $Q.aba = \{t_i, p\}$ ,  $Q.abba = \{t_i, p\}$ , where  $t_i = p.a$  (let us denote by  $w$  one of these two words, which has to be used). If  $w = abba$ , then in order to synchronize the automaton it is enough now to use  $babba$ , so  $m(\mathcal{A}) = 9$ . Notice that this case holds for 4-state automata. Consider now the case with  $w = aba$ . State  $t_i$  must be transformed into  $q$ , while  $p$  will be transformed into itself. To transform  $t_i$  into  $q$ , we have to use  $v = b^i$  for some  $i$ . In the worst case  $i = n - 2$ . We have  $Q.wv = \{p, q\}$ . We need to use  $ab^j a$  for some  $j$  such that

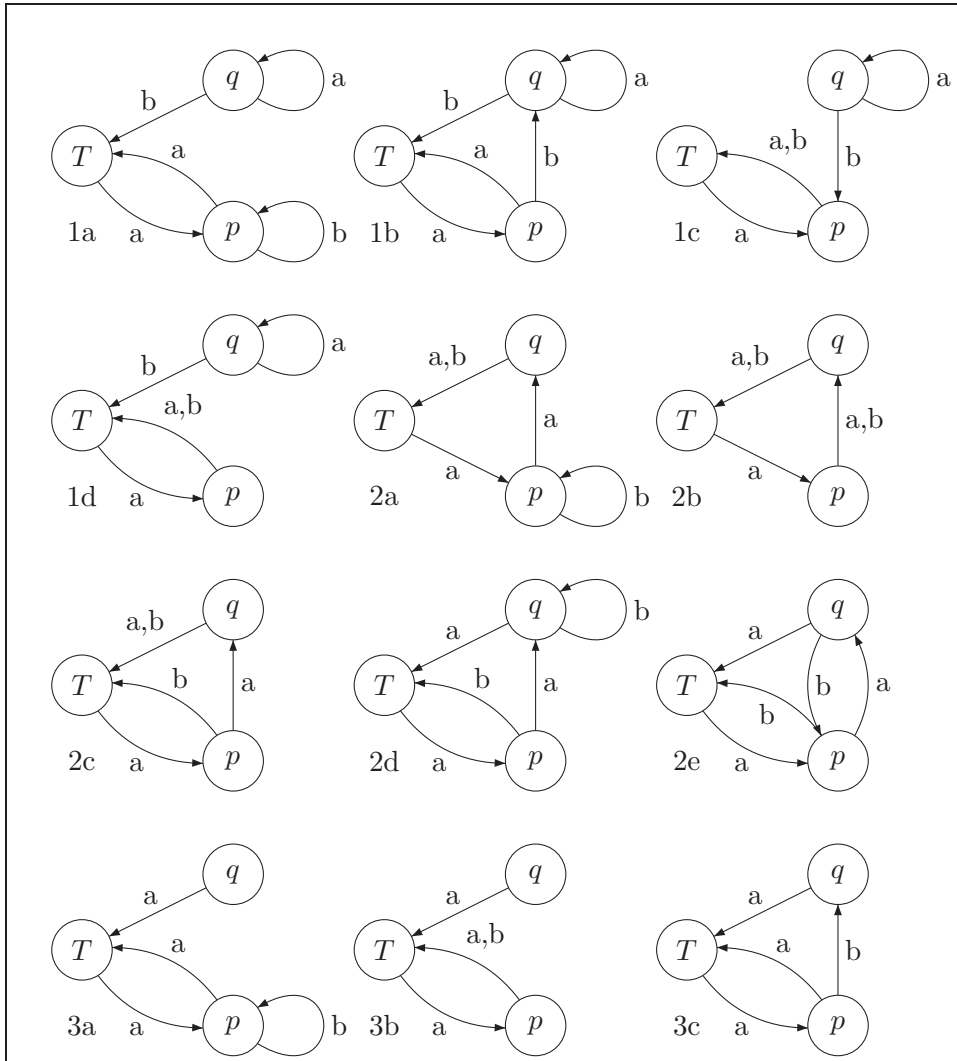


Fig. 3. All possible cases of Theorem 1

$|\{p, q\}.ab^j a| = 1$ , that is  $\{p, q\}.ab^j \in T$ . Because  $\{p, q\}.a = \{q, t_i\}$ , states  $q$  and  $t_i$  must be transformed into some states of  $T$ . If  $t_i.b = q$ , it is enough to use  $abba$ , if  $q.b = t_i$ , it is enough to use  $aba$ . Let us denote this word ( $abba$  or  $aba$ ) by  $u$ . We have  $n + 4 \leq |Q.wvu| \leq n + 5$ , but notice that  $|Q.wvu| = n + 5$  holds only when  $w = abba$ . This implies  $m(\mathcal{A}) = 9$ , so this equality holds only for a 4-state automaton. For automata with  $|Q| > 4$  we have  $m(\mathcal{A}) \leq n + 4$ .

*Case 1b).* A minimal synchronizing word must start with  $a$ . If  $q.bb = p$ ,

then  $abaaba \in S(\mathcal{A})$  and  $m(\mathcal{A}) \leq 6$ . If  $q.bb \in T$ , then  $Q.abba = \{p, q\}$  if  $(p.a).b = p$ ,  $\{p, t\}$  if  $(p.a).bb = p$  or  $p$  in other cases. If  $Q.abba = \{p, q\}$ , then  $(abba)(bba) \in S(\mathcal{A})$  and  $m(\mathcal{A}) \leq 7$ . If  $Q.abba = \{p, t\}$ , then  $(abba)(babba) \in S(\mathcal{A})$  and  $m(\mathcal{A}) \leq 9$  (notice that the very last case holds only for  $|Q| \geq 5$ ).

*Case 1c).* If  $(p.b).b \in T$ , then  $Q.abba = \{p\}$  or  $Q.abba = \{p, q\}$ . In the first case we have  $m(\mathcal{A}) \leq 4$ , in the second one we have  $(abba)(bba) \in S(\mathcal{A})$  and therefore  $m(\mathcal{A}) \leq 7$ . If  $(p.b).b = q$ , then  $(p.a).b^i \in T \forall i$ , so  $|Q.ababa| = 1$  and  $m(\mathcal{A}) \leq 5$ .

*Case 1d).* A minimal synchronizing word must start with  $aba$ .  $Q.aba = \{p, q\}$  (if  $(p.a).b = q$ ),  $\{p, t_i\}$  (if  $(p.a).b = p$ ) or  $p$  in other cases. In the first situation we have  $ababa \in S(\mathcal{A})$  and  $m(\mathcal{A}) \leq 3$ . In the second one, for  $p.b = p.a$  an automaton is not a synchronizing one, so the only possibility is  $(p.b).b \in T$  or  $(p.b).b = q$ . In the first case we have  $(aba)(bba) \in S(\mathcal{A})$  and  $m(\mathcal{A}) \leq 6$ . In the second one, we have  $(aba).(bbaba) \in S(\mathcal{A})$  and  $m(\mathcal{A}) \leq 8$ .

*Case 2a).* If  $q.a = q.b$ , then for  $\mathcal{A}$  to be synchronized  $q.bb$  must be in  $T$  and then  $(aba)(aba) \in S(\mathcal{A})$  and  $m(\mathcal{A}) \leq 6$ . If  $q.a \neq q.b$ , then consider two subcases:  $q.ab \in T$  and  $q.ab = q$ . In the first one we have  $Q.(aba)(aba) = \{p, q\}.aba = p$  and therefore  $m(\mathcal{A}) \leq 6$ . The second subcase implies  $q.bb \in T$  and we have  $Q.(abba)(abba) = \{p, q\}.abba = p$ , so  $m(\mathcal{A}) \leq 8$ .

*Case 2b).* If  $q.a = q.b$ , then necessarily  $q.ab \in T$  and  $Q.(aba)(abba) = p$ , so  $m(\mathcal{A}) \leq 7$ . If  $q.a \neq q.b$ , we consider two subcases:  $q.ab = p$  and  $q.ab \in T$ . If  $q.ab = p$ , then  $Q.(abba)(bbba) = \{p, q.a\}.bbba = p$  and  $m(\mathcal{A}) \leq 8$ . If  $q.ab \in T$ , then  $Q.(aba)(aaba) = \{p, q.a\}.aaba = p$  and  $m(\mathcal{A}) \leq 7$ .

*Case 2c).* If  $(q.a).b \in T$ , then  $Q.aba = p$ ; if  $(q.a).b = p$  then  $Q.aba = \{p, q\}$  and  $(aba)(ba) \in S(\mathcal{A})$ ; if  $(q.a).b = q$ , then  $Q.aba = \{p, t\}$  and  $(aba)(aba) \in S(\mathcal{A})$ . In any case we have  $m(\mathcal{A}) \leq 6$ .

*Case 2d).* If  $(q.a).b = p$ , then  $(p.b).b \in T$  and after applying  $abbabba$  we have  $(Q.abba).bba = \{p, q.a\}.bba = p$ , so  $m(\mathcal{A}) \leq 7$ . If  $(q.a).b \in T$ , then  $(Q.aba).ba = p$  and  $m(\mathcal{A}) \leq 5$ .

*Case 2e).* If  $q.a = p.b$  and  $(q.a).b = q$ , then an automaton is not a synchronizing one; if  $(q.a).b \in T$ , then  $(Q.aba).bba = \{p, q\}.bba$  and  $m(\mathcal{A}) \leq 6$ . If  $q.a \neq p.b$  and  $(q.a).b = q$ , then  $(p.b).b \in T$  and  $(Q.abba).bba = \{p, q\}.bba = p$ , so  $m(\mathcal{A}) \leq 7$ ; if  $(q.a).b \in T$ , then  $(Q.aba).bba = \{p, q\}.bba = p$ , provided  $(p.b).b \in T$ ; if  $(p.b).b = q$ , then  $(Q.aba).bbaba = p$  and  $m(\mathcal{A}) \leq 8$ .

*Case 3a).* If  $p.b = b$ , then  $\exists t \in T : t.b = q \wedge \exists 1 \leq j \leq n-2 : p.b^j = t$ , so in the worst case  $ab^{n-2}a^2 \in S(\mathcal{A})$  and  $m(\mathcal{A}) \leq n+1$ .

*Case 3b).* Let us consider three subcases. *3b.1)*  $p.ab \in T$ . Then  $aaba \in S(\mathcal{A})$ . *3b.2)*  $p.ab = p$ . If  $p.bb \in T$ , then  $aabba \in S(\mathcal{A})$ . If  $p.bb = q$ , then  $aab^4a \in S(\mathcal{A})$  or, if  $q.b = p.a$ ,  $aabbaba \in S(\mathcal{A})$ . *3b.3)*  $p.ab = q$ . If  $p.b = p$ , then  $aabaa \in S(\mathcal{A})$ . If  $p.b \in T$ , we have again four subcases: *3b.3.1)*  $q.ab \in T$ . Then  $aababa \in S(\mathcal{A})$ . *3b.3.2)*  $q.ab = p, p.b \notin \{p.a, q.a\}$ . Then  $aabba \in S(\mathcal{A})$ . *3b.3.3)*  $q.ab = p \wedge p.b = p.a$ . Then  $q.b \neq q.a$  and  $aabbba \in S(\mathcal{A})$ . *3b.3.4)*

$q.ab = p \wedge p.b = q.a$ . Then  $q.b \neq p.a$  and  $aabbba \in S(\mathcal{A})$ . Putting all cases together, we have  $m(\mathcal{A}) \leq 7$ .

*Case 3c).* If  $p.b = q$ , then necessarily  $q.b \in T$ . Consider two subcases: *3c.1)*  $p.abb \in T$ . Then  $aabba \in S(\mathcal{A})$ . *3c.2)*  $p.abb = p$ . Then  $aabaa \in S(\mathcal{A})$ . Therefore,  $m(\mathcal{A}) \leq 5$ .

Putting all these cases together we obtain the following inequality:

$$\mathcal{M}([n - 2]) \leq \begin{cases} 9 & \text{if } |Q| = 4 \\ n + 4 & \text{if } |Q| \geq 5. \end{cases}$$

We will show that these values are reachable. For 4-state automata it is enough to take an automaton  $\mathcal{B}_4$  with the transition function  $\delta = (2131)(1342)$ . We have then  $m(\mathcal{B}_4) = 9$ . For  $n > 4$  we construct the automaton  $\mathcal{B}_{>4}$  with the following transition function:

$$\delta_{\mathcal{B}_{>4}} = (2\ 1\ 1\ \dots\ 1\ n)(1\ 3\ 4\ 5\ 6\ \dots\ n\ 2).$$

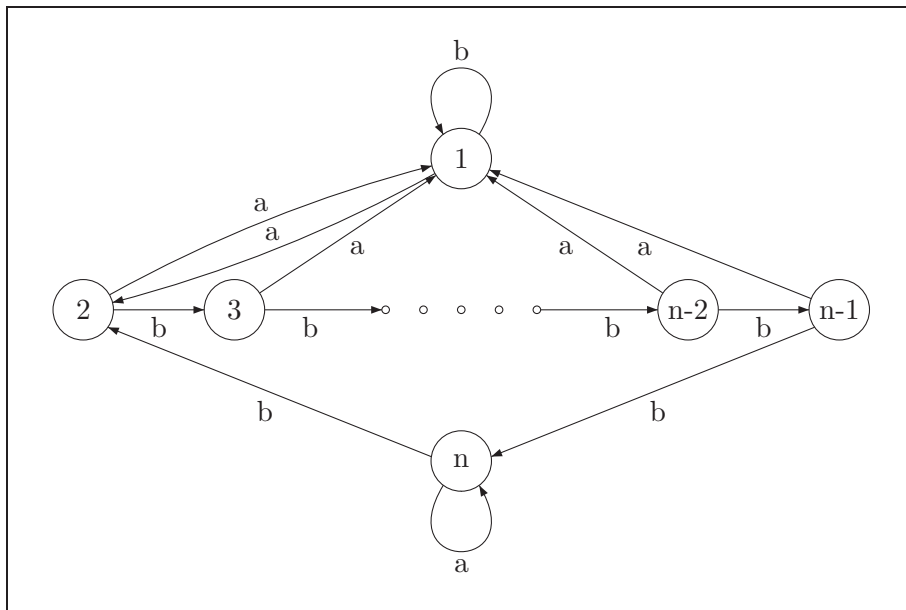


Fig. 4. Automaton  $\mathcal{B}_{>4}$

Construction of  $\mathcal{B}_4$  and  $\mathcal{B}_{>4}$  is derived from the analysis of Case 1a). We will show that  $m(\mathcal{B}_{>4}) = n + 4$ . To do this we use the power-set automaton  $\mathcal{P}(\mathcal{B}_{>4})$ . The shortest path from  $Q$  to a "singleton" state is bolded. The power-set automaton in Fig. 5 is simplified; we omitted labels on some edges, for example  $\{1, 2\}.b^i = \{1, 2\}$  for  $i \neq k(n - 2), k \in \mathbb{N}$ . However, it is

easy to check that the word labelling the bolded path is the shortest possible one and therefore it is the minimal synchronizing word for  $\mathcal{B}_{>4}$ .

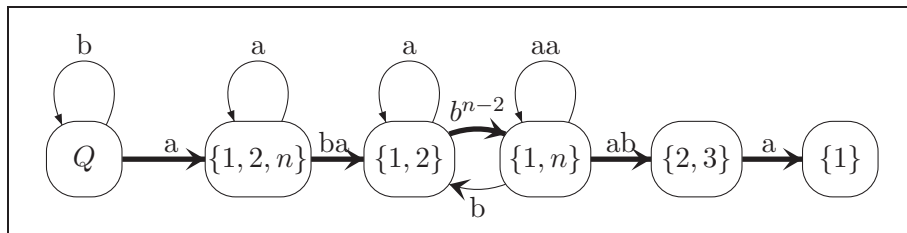


Fig. 5. Power-set automaton  $\mathcal{P}(\mathcal{B}_{>4})$

The length of the shortest path is  $|a| + |ba| + |b^{n-2}| + |ab| + |a| = n + 4$ .

By giving the examples of  $\mathcal{B}_4$  and  $\mathcal{B}_{>4}$  we have shown that the following equality holds:

$$\mathcal{M}([n - 2]) = \begin{cases} 9 & \text{if } |Q| = 4 \\ n + 4 & \text{if } |Q| \geq 5, \end{cases}$$

which ends the proof.

Automaton  $\mathcal{B}_4$ , constructed as in Case 1a), is exactly the same as the automaton given by Černý in [6] as one of the 4-state C-free automata. This is not only the single 4-state automaton of type  $[n - 2]$  reaching the upper bound from the Černý Conjecture, but also the only one 4-state C-free automaton over a binary alphabet [23].

## 7. References

- [1] Ananichev D., Volkov M.; *Collapsing words vs. synchronizing words*, Lecture Notes in Computer Science 2295, 2001, pp. 166–174.
- [2] Ananichev D., Volkov M.; *Synchronizing monotonic automata*, Lecture Notes in Computer Science 2710, 2003, pp. 111–121.
- [3] Ananichev D.S., Volkov M.V.; *Synchronizing Generalized Monotonic Automata*, Workshop on Synchronizing Automata, Turku, Finland, 2004.
- [4] Benenson Y., Adar R., Paz-Elizur T., Livneh L., Shapiro E.; *DNA molecule provides a computing machine with both data and fuel*, Proceedings of the National Academy of Sciences 100, 2003, pp. 2191–2196.

- [5] Černý J.; *Poznámka k. homogénnym experimentom s konečnými automatmi*, Mat. fyz. cas SAV 14, 1964, pp. 208–215.
- [6] Černý J., Pirická A., Rosenauerova B.; *On directable automata*, Kybernetika 7, 1971, pp. 289–298.
- [7] Dubuc L.; *Sur les automates circulaires et la conjecture de Černý*, RAIRO Informatique Théorique et Applications 32, 1998, pp. 21–34 [in French].
- [8] Eppstein D.; *Reset sequences for monotonic automata*, SIAM Journal on Computing 19, 1990, pp. 500–510.
- [9] Frankl P.; *An extremal problem for two families of sets*, European Journal of Combinatorics 3, 1982, pp. 125–127.
- [10] Kari J.; *A counter example to a conjecture concerning synchronizing words in finite automata*, EATCS Bulletin 73, 2001, p. 146.
- [11] Kari J.; *Synchronizing finite automata on Eulerian digraphs*, Lecture Notes in Computer Science 2136, 2001, pp. 432–438.
- [12] Kari J.; *Synchronization and Stability of Finite Automata*, Journal of Universal Computer Science 8(2), 2002, pp. 270–277.
- [13] Kljachko A., Rystsov I.K., Spivak M.A.; *An extremely combinatorial problem connected with the bound on the length of a recurrent word in an automata*, Kybernetika 2, 1987, pp. 16–25.
- [14] Natarajan B.K.; *An algorithmic Approach to the Automated Design of Parts Orienters*, Proceedings of the 27th Annual Symposium on Foundations of Computer Science, IEEE, 1986, pp. 132–142.
- [15] Pin J.-E.; *Le probleme de la synchronisation et la conjecture de Černý*, These de 3eme cycle, Université Paris VI, 1978.
- [16] Pin J.-E.; *On two combinatorial problems arising from automata theory*, Annals of Discrete Mathematics 17, 1983, pp. 535–548.
- [17] Roman A.; [www.ii.uj.edu.pl/~roman/publications.html](http://www.ii.uj.edu.pl/~roman/publications.html)
- [18] Roman A., Forys W.; *Lower Bound for the Length of Synchronizing Words in Partially-Synchronizing Automata*, Lecture Notes in Computer Science 4910, 2008, pp. 448–459 .
- [19] Roman A.; *Synchronizing Finite Automata with Short Reset Words*, Applied Mathematics and Computation 209, 2009, pp. 125–136.
- [20] Salomaa A.; *Compositions over a Finite Domain: from Completeness to Synchronizable Automata*, Turku Centre for Computer Science, Technical Report No 350, 2000.



- [21] Trahtman A.N.; *An efficient algorithm finds noticeable trends and examples concerning the Cerny conjecture*, Lecture Notes in Computer Science 4162, 2006, pp. 789–800.
- [22] Trahtman A.N.; *Some results of implemented algorithms of synchronization*, 10th Journées Montoises d'Informatique Théorique, Liège, Belgium, September 8–11, 2004.
- [23] Trahtman A.N.; *Computations of some examples and notable trends concerning the synchronization*, 4th Workshop in Applied and Computational Mathematics, Tel-Aviv 2006.

*Received February 24, 2010*