View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com

ScienceDirect

Theoretical Computer Science 376 (2007) 30–41

Theoretical Computer Science

www.elsevier.com/locate/tcs

Synchronizing automata with a letter of deficiency 2

D.S. Ananichev, M.V. Volkov*, Yu.I. Zaks

Department of Mathematics and Mechanics, Ural State University, 620083 Ekaterinburg, Russia

Abstract

We present two infinite series of synchronizing automata with a letter of deficiency 2 whose shortest reset words are longer than those for synchronizing automata obtained by a straightforward modification of Černý's construction.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Deterministic finite automaton; Synchronizing automaton; Reset word; Černý conjecture; Deficiency of a transformation

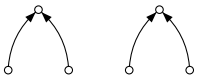
0. Background and motivation

Let $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ be a *deterministic finite automaton* (DFA), where Q is the state set, Σ stands for the input alphabet, and $\delta : Q \times \Sigma \to Q$ is the transition function defining an action of the letters in Σ on Q. The action extends in a unique way to an action $Q \times \Sigma^* \to Q$ of the free monoid Σ^* over Σ; the latter action is still denoted by δ. The DFA $\mathscr{A}$ is called *synchronizing* if there exists a word $w \in \Sigma^*$ whose action resets $\mathscr{A}$, that is it leaves the automaton in one particular state no matter which state in Q it starts at: $\delta(q_1, w) = \delta(q_2, w)$ for all $q_1, q_2 \in Q$. Any such word w is said to be a *reset word* for the DFA.

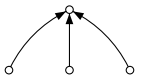
It is rather natural to ask how long a reset word for a given synchronizing automaton may be. The problem is known to be NP-complete (see [4] or [12]), but on the other hand, there are some upper bounds on the minimum length of reset words for synchronizing automata with a given number of states. The best such bound known so far is due to Pin [11] (it is based on a combinatorial theorem conjectured by Pin and then proved by Frankl [5]): for each synchronizing automaton with n states, there exists a reset word of length at most $(n^3 - n)/6$. In 1964 Černý [2] produced for each $n > 1$ a synchronizing automaton $\mathscr{C}_n$ with n states whose shortest reset word has length $(n-1)^2$ and conjectured that these automata represent the worst possible case, that is, every synchronizing automaton with n states can be reset by a word of length $(n-1)^2$. By now this simply looking conjecture is arguably the most long-standing open problem in the combinatorial theory of finite automata. The reader is referred to the survey [9] for an interesting overview of the area and its relations to multiple-valued logic and symbolic dynamics; applications of synchronizing automata to robotics are discussed in [4,6]. (A more recent survey [13] contains a detailed account of algorithmic and complexity issues in the field but unfortunately omits some important references.)

* Corresponding author.
E-mail addresses: Dmitry.Ananichev@usu.ru (D.S. Ananichev), Mikhail.Volkov@usu.ru (M.V. Volkov), zaksjulia@r66.ru (Yu.I. Zaks).

0304-3975/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2007.01.010



The action of a bactrian letter.



The action of a dromedary letter.

Fig. 1. Two kinds of letters of deficiency 2.

There are many papers where the Černý conjecture is proved for various restricted classes of synchronizing automata (cf. [4,3,8,1,14], to mention a few recent advances only). On the other hand, there are only very few examples of "slowly" synchronizing automata, that is automata whose shortest reset words have lengths close to the Černý bound. In fact, it seems that the only infinite series of n-state synchronizing automata with shortest reset words of length $O(n^2)$ that has appeared in the literature so far is the Černý series $\mathscr{C}_n$, $n = 2, 3, \ldots$. Of course, one can obtain more examples by some slight modifications of the Černý automata (we shall discuss this later) but in general "slowly" synchronizing automata turn out to be rather exceptional. This observation is supported not only by numerous experiments (see [15] for a description of certain noteworthy experimental results in the area) but also by probabilistic arguments. Indeed, if Q is an n-element set (with n large enough), then, on average, any product of $2n$ randomly chosen transformations of Q is known to be a constant map; cf. [7]. Restated in automata-theoretic terms, this fact implies that a randomly chosen DFA with n states and a sufficiently large input alphabet tends to be synchronizing, and moreover, the length of its shortest reset word does not exceed $2n$.

In the present paper we construct two new infinite series of "slowly" synchronizing automata. In contrast with the Černý series, in our automata one of the letters acts as a transformation of deficiency 2. (Recall that the *deficiency* of a transformation φ of a finite set Q is the difference $|Q|-|\varphi(Q)|$.) Since, in the presence of such a letter, synchronization speeds up, one cannot expect the lengths of shortest reset words for our automata to reach the Černý bound. However, surprisingly, our examples turn out to synchronize more slowly than automata with a letter of deficiency 2 derived in a natural way from the Černý automata.

Besides enlarging our supply of examples, there are various additional motivations for studying synchronizing automata with a letter of deficiency 2. For instance, we recall that the best upper bound known so far for the minimum length $\ell(n)$ of reset words for synchronizing automata with n states is cubic. Clearly, finding a quadratic upper bound for $\ell(n)$ would constitute a major step towards a proof of the Černý conjecture. It can be easily verified that if a quadratic in n function $f(n)$ provides an upper bound for the minimum length of reset words for n-state synchronizing automata with a letter of deficiency 2, then the function $4f(n)$ can serve as an upper bound for $\ell(n)$. Thus, approaching the problem through automata with a letter of deficiency 2 might be a reasonable strategy. However we shall not touch this approach in the present paper.

1. Main results and a discussion

Let $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ be a DFA with $|Q| \geq 3$. If a letter $a \in \Sigma$ is such that the transformation of the set Q induced by the action of a has deficiency 2, then exactly one of the two following situations happens.

1. There exist four different states $q_1, q_2, q_3, q_4 \in Q$ such that

$$\delta(q_1, a) = \delta(q_2, a) \neq \delta(q_3, a) = \delta(q_4, a).$$

In this situation we say that a is a *bactrian letter*.

2. There exist three different states $q_1, q_2, q_3 \in Q$ such that

$$\delta(q_1, a) = \delta(q_2, a) = \delta(q_3, a).$$

In this case we call a a *dromedary letter*.

Fig. 1 illustrates these notions and explains the terminology.

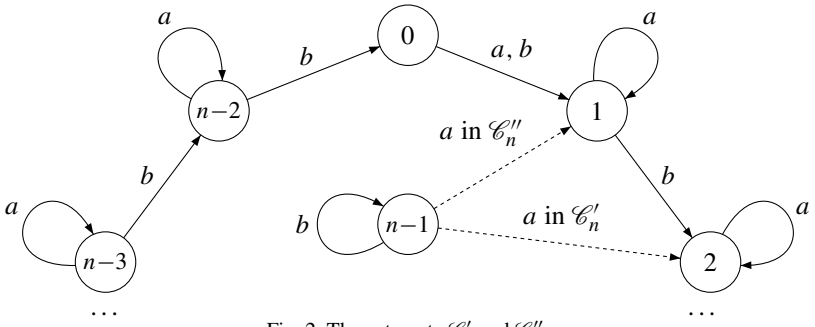


Fig. 2. The automata $\mathscr{C}'_n$ and $\mathscr{C}''_n$.

An easy way to obtain slowly synchronizing automata with a letter of deficiency 2 of either type consists in modifying the Černý automata. Namely, consider the Černý automaton $\mathscr{C}_{n-1}$ whose states are the residues modulo $n-1$ and whose input letters a and b act as follows:

$$\delta(m,a)=\begin{cases}1 & \text{for } m=0,\\ m & \text{for } 1<m<n-1;\end{cases}\qquad \delta(m,b)=m+1\,(\mathrm{mod}\ n-1).$$

We add to $\mathscr{C}_{n-1}$ an extra state denoted $n-1$ and then extend the transition function by letting $\delta(n-1,a)=2$, $\delta(n-1,b)=n-1$. This gives an n-state DFA $\mathscr{C}'_n$ in which a becomes a bactrian letter. Similarly, if we extend δ by defining $\delta(n-1,a)=1$, $\delta(n-1,b)=n-1$, we obtain another n-state DFA $\mathscr{C}''_n$ in which a is a dromedary letter. Both modifications are shown in Fig. 2.

It can be verified that the word $(ab^{n-2})^{n-3}a$, which resets the automaton $\mathscr{C}_{n-1}$, resets also both $\mathscr{C}'_n$ and $\mathscr{C}''_n$ and is in fact the shortest reset word for each of these automata. Hence $(n-2)^2$, i.e. the length of this word, turns out to be a lower bound for the minimum length of reset words for n-state synchronizing automata with a letter of deficiency 2 of either type. By analogy with the Černý conjecture, one may think that the bound is tight. However, as our results show, this is not the case.

Our first result significantly improves the lower bound for synchronizing automata with a bactrian letter:

Theorem 1.1. *For each odd $n>3$, there exists a synchronizing automaton $\mathscr{B}_n$ with n states and two input letters, one of which is bactrian such that the shortest reset word of $\mathscr{B}_n$ is of length $(n-1)(n-2)$.*

The proof of Theorem 1.1 is presented in Section 2. In our opinion, this proof is of independent interest as it involves a trick which, to the best of our knowledge, has not appeared in synchronization proofs so far.

It seems that the restriction on the parity of the quantity of states in Theorem 1.1 is essential. If n is even, then the construction used to design the automaton $\mathscr{B}_n$ still works but produces an automaton which is not synchronizing. For $n=6$ we have found a synchronizing automaton with two input letters including one bactrian and with the shortest reset word of length $(6-1)(6-2)=20$ but already for $n=8$ our best bactrian example has the shortest reset word of length $39<(8-1)(8-2)=42$. These examples are also presented in Section 2.

Now consider the dromedary case. Here we are also able to slightly improve the lower bound coming from the 'Černý-like' example $\mathscr{C}''_n$ but in contrast with the bactrian case we need three input letters this time.

Theorem 1.2. *For each $n>4$, there exists a synchronizing automaton $\mathscr{D}_n$ with n states and three input letters, one of which is dromedary such that the shortest reset word of $\mathscr{D}_n$ is of length $(n-2)^2+1$.*

The proof of Theorem 1.2 shares some ideas with the proof of Theorem 1.1 but is more bulky. It is presented in Section 3.

For $n=5$ and $n=6$, we have found some dromedary examples (again with three input letters) whose shortest reset words are one letter longer than those of respectively $\mathscr{D}_5$ and $\mathscr{D}_6$. These examples indicate that there may exist a series of n-state synchronizing automata with three input letters including one dromedary whose shortest reset words are of length $(n-2)^2+2$ but we have not managed to find such a series so far.

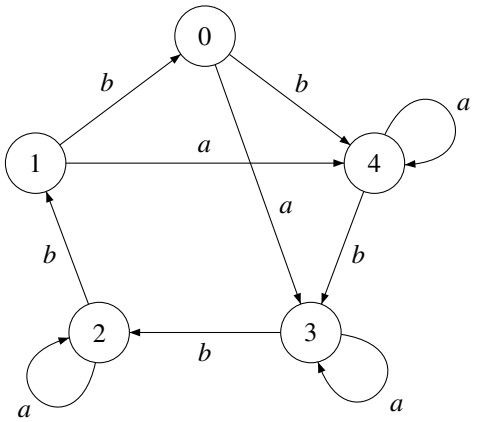


Fig. 3. The automaton $\mathscr{B}_5$.

2. The automata $\mathscr{B}_n$

Let $n = 2k + 1$, $k > 1$. The states of the automaton $\mathscr{B}_n$ are the residues modulo n and its input letters a and b act as follows:

$$\delta(m, a) = \begin{cases} m - 2 \,(\mathrm{mod}\, n) & \text{for } m = 0, 1, \\ m & \text{for } 1 < m < n; \end{cases} \qquad \delta(m, b) = m - 1 \,(\mathrm{mod}\, n).$$

Observe that a is a bactrian letter in $\mathscr{B}_n$. The smallest automaton in the series is shown in Fig. 3.

The next lemma can be straightforwardly checked and we omit its proof.

Lemma 2.1. *Let $n = 2k + 1$, $k > 1$. Then the word*

$$(ab^{2k-1})^{k-1}ab^{2k-2}(ab^{2k-1})^{k-1}a \tag{1}$$

is a reset word for the automaton $\mathscr{B}_n$.

The length of the word (1) is $2k(k-1) + 2k - 1 + 2k(k-1) + 1 = 2k(2k-1) = (n-1)(n-2)$. We observe in passing that $\mathscr{B}_n$ has yet another reset word of the same length.

To complete the proof of Theorem 1.1, it remains to show that the length of each reset word for $\mathscr{B}_n$ is at least $(n-1)(n-2)$. For this, we use a solitaire-like game on the underlying graph of $\mathscr{B}_n$. Assume that some of the states of $\mathscr{B}_n$ are covered with pairwise distinct *coins* as shown in Fig. 4. Each *move*, that is the action of a letter $c \in \{a, b\}$, makes the coins slide along the arrows labelled c so that a state m will be covered with a coin after the move if and only if there exists a state ℓ such that $\delta(\ell, c) = m$ and ℓ was covered with a coin before the move. If two coins happen to arrive at the same state m, then from the structure of $\mathscr{B}_n$ we conclude that $c = a$, $m = n - 1$ or $m = n - 2$ and both m and $m + 2 \,(\mathrm{mod}\, n)$ held coins before the move. Then we retain the coin that had covered m before the move and delete the coin arriving from $m + 2 \,(\mathrm{mod}\, n)$. Fig. 5 demonstrates how the position shown in Fig. 4 changes after a single action of a letter.

Suppose that initially all the states of the automaton $\mathscr{B}_n$ are covered with coins and let a word $w \in \{a, b\}^*$ (that is the sequence of its letters) act on this initial position. It is easy to see that after completing this action coins cover precisely the states in the image of the transformation $\delta(_, w)$. In particular, if w is a reset word for $\mathscr{B}_n$, then after the action of w only one coin survives.

Now we can explain the idea of our proof of Theorem 1.1. Given a reset word w and an initial distribution P_0 of n coins on the states of $\mathscr{B}_n$, let P_i, $0 \le i \le |w|$, stand for the position that arises when we apply the prefix of w of length i to the position P_0. To each position P_i, we shall assign an integer parameter $\mathrm{wg}(P_i)$ (called the *weight* of the position) such that the following three conditions will be satisfied:

(i) $\mathrm{wg}(P_0) \ge (n-1)^2$;
(ii) $\mathrm{wg}(P_{|w|}) \le n - 1$;

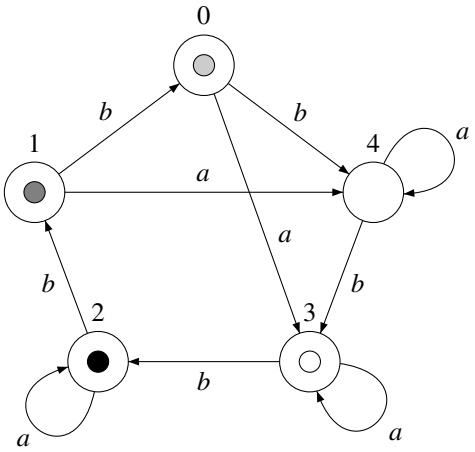


Fig. 4. A position on $\mathscr{B}_5$.

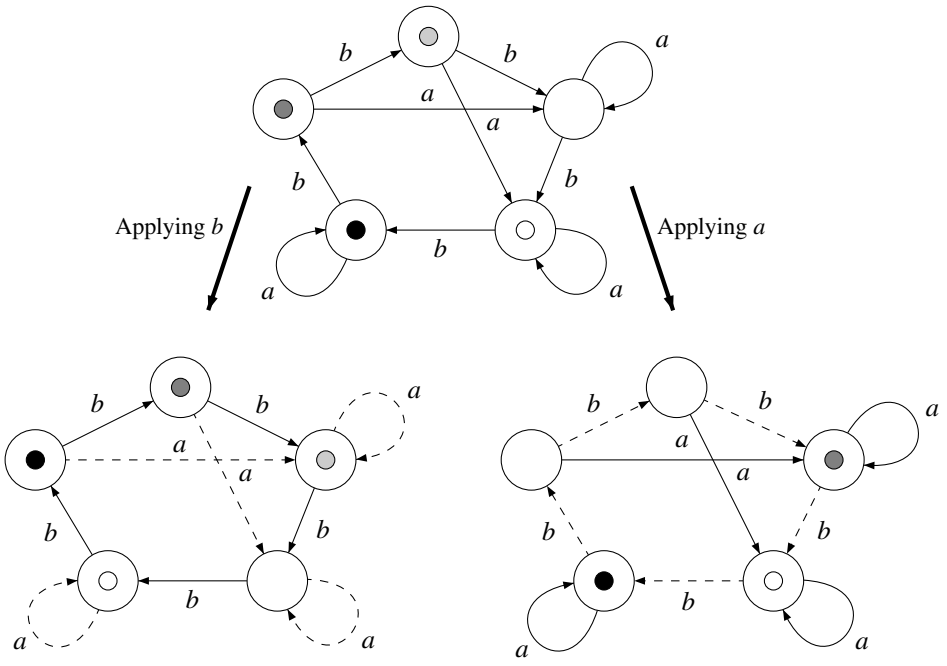


Fig. 5. Redistributing coins under the actions of b (left) and a (right).

(iii) for each $i = 1, \ldots, |w|$, the action of the ith letter of w decreases the weight of P_{i-1} by 1 at most, that is, $1 \geq \mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)$.

Clearly, if such a weight function does indeed exist, then summing up all the inequalities in (iii) and utilizing (i) and (ii), we obtain

$$|w| = \sum_{i=1}^{|w|} 1 \geq \sum_{i=1}^{|w|} \big(\mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i)\big) = \mathrm{wg}(P_0) - \mathrm{wg}(P_{|w|}) \geq (n-1)^2 - (n-1) = (n-1)(n-2),$$

as required.

It remains to construct a weight function satisfying (i)–(iii). This is by no means an easy task because some moves can delete two coins at once. It is to overcome this difficulty that we let our coins be distinguishable from each other—this allows us to make weight functions depend on reset words while a 'uniform' function serving all reset words simultaneously may not exist.

Thus, let us fix a reset word w and an initial distribution P_0 of n coins on the states of $\mathscr{B}_n$. As mentioned, the action of w on P_0 removes $n-1$ coins. We call the only coin that remains after the action the *golden* coin and denote it by G. Now fix a position P_i, $0 \le i \le |w|$. For any coin C that is present in this position, let $m_i(C)$ be the state covered with C. We denote by $d_i(C)$ the least non-negative integer such that $\delta\left(m_i(C), b^{2d_i(C)}\right) = m_i(G)$. In the 'visual' terms, $d_i(C)$ is the number of double steps on the 'main circle' of $\mathscr{B}_n$ (measured clockwise) from the state covered with C to the state covered with the golden coin. We define the *weight* of C in the position P_i as

$$\mathrm{wg}(C, P_i) = (n-1) \cdot d_i(C) + m_i(C).$$

(Observe that here we multiply and add integers and not residues modulo n.) In order to illustrate this definition, assume that the black coin in the position shown in Fig. 4 is the golden coin. Then the weight of the white coin in this position is equal to $4 \cdot 3 + 3 = 15$ because the white coin covers the state 3 and from this state one needs three double steps in the clockwise direction in order to reach the state 2 covered with the golden coin. Similarly, the weight of the dark-grey coin in Fig. 4 is $4 \cdot 2 + 1 = 9$ and the weight of the light-grey coin is $4 \cdot 4 + 0 = 16$. As for the black (=golden) coin, its weight is $4 \cdot 0 + 2 = 2$ because, by the definition, the weight of the golden coin in any position is equal to the state it covers.

Now we define the *weight* $\mathrm{wg}(P_i)$ of the position P_i as the maximum of the weights of the coins present in this position. For instance, the weight of the position shown in Fig. 4 is 16 (if, as above, one assumes that the black coin is the golden one). It remains to verify that this weight function satisfies Conditions (i)–(iii).

Condition (i): $\mathrm{wg}(P_0) \ge (n-1)^2$. In the initial position all states are covered with coins. Consider the coin C that covers the state $m_0(G) - 2 \pmod n$, that is the state in one double step clockwise after the state covered with the golden coin. Then it is easy to see that $d_0(C) = n-1$ whence $\mathrm{wg}(C, P_0) = (n-1)\cdot(n-1) + m_0(C) \ge (n-1)^2$. Since the weight of a position is not less than the weight of any coin in this position, we conclude that $\mathrm{wg}(P_0) \ge (n-1)^2$.

Condition (ii): $\mathrm{wg}(P_{|w|}) \le n-1$. In the final position only the golden coin G remains, whence the weight of $P_{|w|}$ is the weight of G. As already observed, $\mathrm{wg}(G, P_i) = m_i(G)$ for any position P_i and, clearly, $m_i(G) \le n-1$.

Condition (iii): $\mathrm{wg}(P_{i-1}) - \mathrm{wg}(P_i) \le 1$ for $i = 1, \ldots, |w|$. Let us fix a coin C of maximum weight in P_{i-1}. First consider the case when the letter that causes the transition from P_{i-1} to P_i is b. Recall that $\delta(m, b) = m - 1 \pmod n$. This implies that $d_i(C) = d_{i-1}(C)$ (because the relative location of the coins does not change) and

$$m_i(C) = \begin{cases} m_{i-1}(C) - 1 & \text{if } m_{i-1}(C) > 0, \\ n-1 & \text{if } m_{i-1}(C) = 0. \end{cases}$$

We see that

$$\begin{aligned} \mathrm{wg}(P_i) \ge \mathrm{wg}(C, P_i) &= (n-1) \cdot d_i(C) + m_i(C) \ge \\ &(n-1) \cdot d_{i-1}(C) + m_{i-1}(C) - 1 = \mathrm{wg}(C, P_{i-1}) - 1 = \mathrm{wg}(P_{i-1}) - 1. \end{aligned}$$

Next suppose that the transition from P_{i-1} to P_i is caused by the action of a. Recall that a sends the states 0 and 1 to the states $n-2$ and $n-1$ respectively (that is one double step clockwise) and fixes all other states. If the coin C covers neither 0 nor 1, then $m_i(C) = m_{i-1}(C)$ and

$$d_i(C) = \begin{cases} d_{i-1}(C) & \text{if the golden coin } G \text{ covers neither 0 nor 1}, \\ d_{i-1}(C) + 1 & \text{if } G \text{ covers either 0 or 1}. \end{cases}$$

We conclude that

$$\begin{aligned} \mathrm{wg}(P_i) &\ge \mathrm{wg}(C, P_i) = (n-1) \cdot d_i(C) + m_i(C) \\ &\ge (n-1) \cdot d_{i-1}(C) + m_{i-1}(C) = \mathrm{wg}(C, P_{i-1}) = \mathrm{wg}(P_{i-1}). \end{aligned}$$

Thus, here the transition from P_{i-1} to P_i does not decrease the weight.

It remains to consider the subcase when the coin C covers either 0 or 1. As these two possibilities are analyzed with precisely the same argument, we assume that C covers 0. Then in the position P_i the state $n-2$ holds a coin C' (which may or may not coincide with C). If in the position P_{i-1} the golden coin G covers either 0 or 1, then $d_i(C') = d_{i-1}(C)$ whence

$$\mathrm{wg}(P_i) \geq \mathrm{wg}(C', P_i) = (n-1)\cdot d_i(C') + n - 2 > (n-1)\cdot d_{i-1}(C) = \mathrm{wg}(C, P_{i-1}) = \mathrm{wg}(P_{i-1}).$$

We see that here the weight even increases. Finally, if the coin G covers neither 0 nor 1, it does not move, whence $d_i(C') = d_{i-1}(C) - 1$. Therefore

$$\begin{aligned}\mathrm{wg}(P_i) &\geq \mathrm{wg}(C', P_i) = (n-1)\cdot d_i(C') + n - 2\\ &= (n-1)\cdot(d_{i-1}(C) - 1) + n - 2 = (n-1)\cdot d_{i-1}(C) - 1 = \mathrm{wg}(C, P_{i-1}) - 1 = \mathrm{wg}(P_{i-1}) - 1,\end{aligned}$$

as required.

Thus, we have verified that our weight function satisfies Conditions (i)–(iii), and this completes the proof of Theorem 1.1.

It is very tempting to conjecture that the expression $(n-1)(n-2)$ gives the exact value for the minimum length of reset words for n-state synchronizing automata with a letter of deficiency 2 when $n \geq 5$ is odd. So far we have been able to confirm this only for $n = 5$ (thus solving a question mentioned in Pin's early survey [10]).

As mentioned in Section 1, there is a synchronizing automaton $\mathscr{B}_6^\star$ with two input letters including one bactrian whose shortest reset word is of length 20, thus matching the lower bound $(n-1)(n-2)$ established in Theorem 1.1 for odd values of n. This automaton is shown in Fig. 6; its shortest reset word is $(ab^3ab^2)^2ab^4a$.

However, the example seems to be exceptional. We have exhaustively searched through all eight-state automata with two input letters of which one is bactrian and the other acts as a permutation. This search has yielded no synchronizing automaton whose shortest reset word would be of length $(8-1)(8-2) = 42$; moreover, the maximum length of shortest reset words for such automata turns out to be 39. The latter value is achieved on a unique automaton $\mathscr{B}_8^\star$ shown in Fig. 7. Its shortest reset word is $ab\cdot a^2b^3ab^3a^2b^3\cdot abab^3a\cdot(ab^2ab)^2\cdot a^2b^3a$.

3. The automata $\mathscr{D}_n$

Take an $n > 4$ and let $\mathscr{D}_n$ be the DFA with the state set $\{1, 2, \ldots, n\}$, with the input alphabet $\{a, b, c\}$ and the transition function δ defined as follows:

m	1	2	3	4	5	...	n
$\delta(m, a)$	1	1	1	4	5	...	n
$\delta(m, b)$	1	1	2	4	5	...	n
$\delta(m, c)$	4	1	4	5	6	...	3

Thus, both a and b fix each state m with $4 \leq m \leq n$ and c acts on the set $\{3, 4, \ldots, n\}$ as a cyclic shift. The automaton $\mathscr{D}_n$ is shown in Fig. 8.

Verifying the following lemma amounts to a straightforward calculation:

Lemma 3.1. *Let $n > 4$. Then the word*

$$c^2(bc^{n-1})^{n-4}bc^2 \tag{2}$$

is a reset word for the automaton $\mathscr{D}_n$.

The length of the word (2) is $n(n-4)+5 = (n-2)^2+1$ and we shall prove that this is in fact the minimum length of a reset word for $\mathscr{D}_n$. Observe that the word (2) does not involve the letter a, and therefore, it also resets the DFA obtained from $\mathscr{D}_n$ by omitting a. Thus, we see (and this is a bit surprising) that adding a letter of deficiency 2 to a synchronizing automaton in which all letters have deficiency 1 may not decrease the minimum length of reset words.

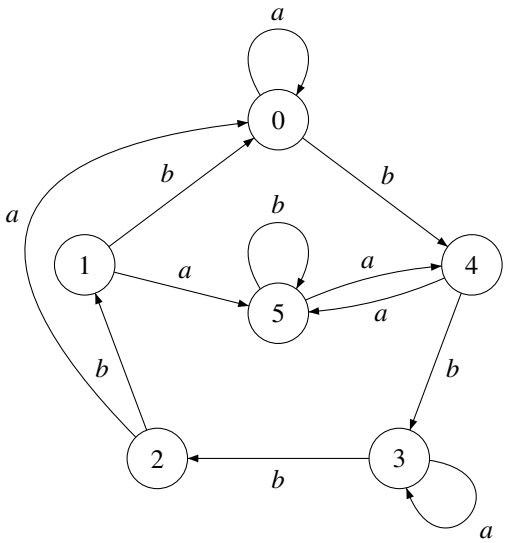


Fig. 6. The automaton $\mathscr{B}_6^\star$.

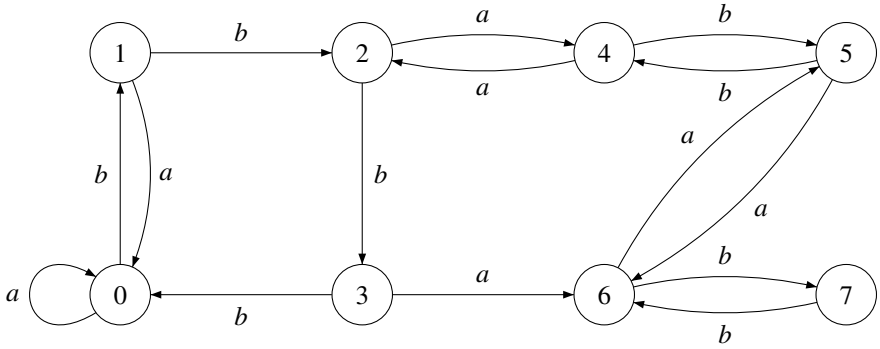


Fig. 7. The automaton $\mathscr{B}_8^\star$.

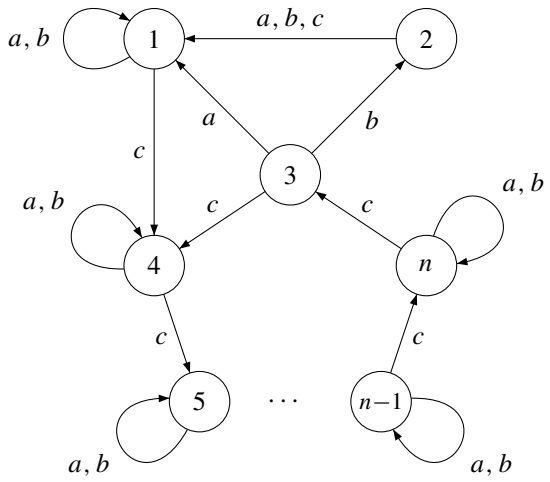


Fig. 8. The automaton $\mathscr{D}_n$.

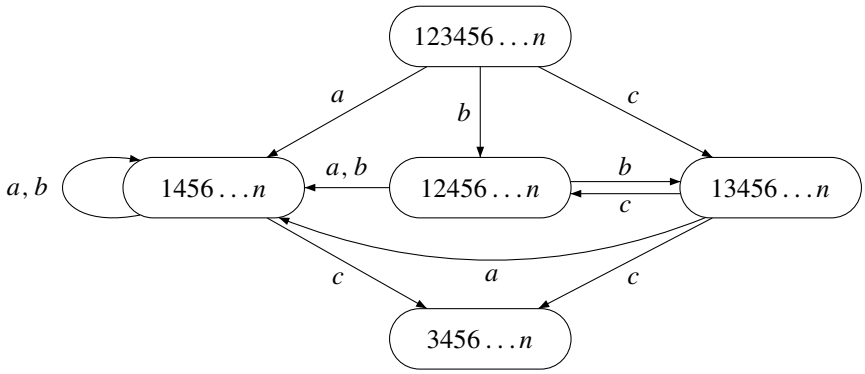


Fig. 9. A fragment of the power-set automaton of $\mathscr{D}_n$.

As in the proof of Theorem 1.1, we use a version of the solitaire-game approach. However, in contrast to Section 2, here we assume that coins used in our game are non-distinguishable. As above, a *move* is the action of a letter; a state m is covered with a coin after the move $d \in \{a, b, c\}$ if and only if there exists a state ℓ such that $\delta(\ell, d) = m$ and ℓ held a coin before the move. Thus, the rule amounts to saying that coins slide along the arrows labelled ℓ and, whenever several coins arrive at the same state, all but one of them is removed.

Suppose that initially all the states of the automaton $\mathscr{D}_n$ are covered with coins. As in the proof of Theorem 1.1, it is easy to realize that a word $w \in \{a, b, c\}^*$ resets $\mathscr{D}_n$ if and only if the action of w on this initial position removes $n - 1$ coins. Considering the 'top' part of the power-set automaton of $\mathscr{D}_n$ as shown in Fig. 9, one observes that any reset word of the minimum length should start with either ac or c^2. The action of either of these words frees the states 1 and 2 so that the remaining coins cover precisely the set $\mathcal{C} = \{3, 4, \ldots, n\}$ (that will be referred to as the *main circle* of $\mathscr{D}_n$).

It is clear that if a coin situated on the main circle is eventually removed, it should first exit from $\mathcal{C}$ through the state 3. We say that two coins covering some states $\ell, m \in \mathcal{C}$ can be *properly merged* if there exists a word $v \in \{a, b, c\}^*$ (called a *merging word*) such that

- $\delta(\ell, v) = \delta(m, v)$;
- $\delta(\ell, u) \neq \delta(m, u)$ for any proper prefix u of v;
- during the action of v, each of the two coins exits from the main circle at most once.

In more formal terms, the latter condition means that at most one prefix u_1 of v satisfies $\delta(\ell, u_1) \notin \mathcal{C}$ and at most one prefix u_2 of v satisfies $\delta(m, u_2) \notin \mathcal{C}$.

Lemma 3.2. *Suppose that two coins situated on the main circle $\mathcal{C}$ can be properly merged. Let the coin that exits from $\mathcal{C}$ first be C_1 and let C_2 be the other coin. Then C_2 immediately follows C_1 (in the clockwise direction) on $\mathcal{C}$.*

Proof. Let v be the corresponding merging word. As observed above, the only way to exit from the main circle is through the state 3. Denote by u the prefix of v that brings C_1 from its initial position to 3, just before the coin leaves the main circle. Arguing by contradiction, we assume that u brings C_2 to a state $m \in \mathcal{C}$ different from n. Then under the action of the next letter $d \in \{a, b\}$ of v the coin C_1 exits from the main circle while C_2 keeps covering the state m.

If during the action of v, the coin C_2 also exits from the main circle, it must reach 3 from the state m. This is only possible under the action of a word v' in which the letter c occurs more than once. Hence such a factor v' should follow the prefix ud in v. However, then the action of v' returns the coin C_1 to a state $\ell \in \mathcal{C}$ such that $\ell \neq 3$ and no proper merging is possible.

If C_2 remains on the main circle during the action of v, then the only state at which the coins could merge is 4. In order to bring the coin C_2 from m to 4, one has to apply a word v'' that contains more than two occurrences of the letter c. We conclude that the word v decomposes as $v = udv''$. However, then the action of v'' brings the coin C_1 to a state $\ell \neq 4$, a contradiction again.

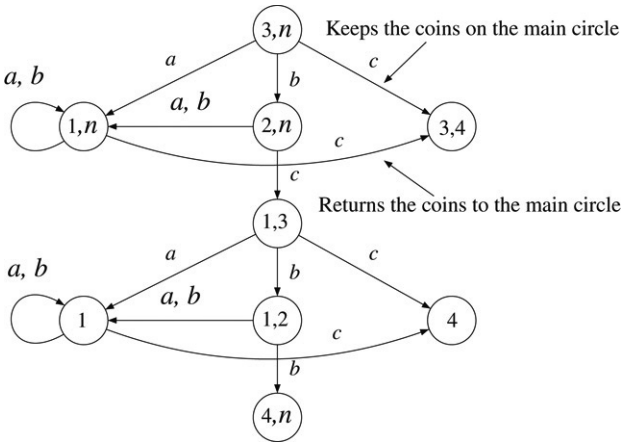


Fig. 10. Another fragment of the power-set automaton of $\mathscr{D}_n$.

In view of Lemma 3.2, we may assume that coins that can be properly merged cover the states 3 and n. (The general situation can be reduced to this partial one with a cyclic shift caused by the action of a suitable power of c.) Such coins can merge on either 4 or 1. Consider yet another fragment of the power-set automaton of $\mathscr{D}_n$ as shown in Fig. 10.

Inspecting Fig. 10 shows that the merging word should start with b because the action of c keeps the coins on the main circle without merging and a sends them to the set $\{1, n\}$ from where they can only return back to the main circle without merging. Furthermore, one can conclude that any merging word coincides with one of the following:

$$bcc,\ bcax,\ bcbax,\ bcbbx,\ bcaxc,\ bcbaxc,\ bcbbxc, \tag{3}$$

where x is a word in the language $\{a, b\}^*$. One can directly check that during the action of any of these words no coin besides the two involved in merging leaves the main circle. The coin that appears as the result of merging covers either 1 or 4. If it covers 1, the after the next application of c the coin returns to the main circle. This means that in the sequel we can analyze only coins on the main circle. Observe also that the shortest word in (3) has length 3.

It is convenient to give a name to states bearing no coins; we call them *holes*. Any sequence of adjacent holes in the main circle is called a *lacuna*. It is easy to see that lacunas can 'grow' only in the clockwise direction. Indeed, given a lacuna, let C be the coin following it in the clockwise direction. We first move C to the state 3 and then apply a word of the form (3). This makes the lacuna one hole longer (and transfers C to the state 4). Observe that if there is a coin immediately after C (in the clockwise direction), then the action causes a merging and a new hole is added to the lacuna. If a hole follows C, then the action described merely transposes C and the hole (so the next lacuna becomes one hole shorter).

If one wants to repeat the process, one first should move the coin C from the state 4 back to the to the state 3, and for this one has to apply a word u with $n-3$ occurrences of the letter c (in particular, $|u| \geq n-3$).

Now let w be an arbitrary reset word of minimum length for the automaton $\mathscr{D}_n$. Recall that w must start with ac or c^2, and after the action of this prefix coins cover precisely the states in the main circle. Thus, at this moment there are no lacunas at all but after completing the action of w we get a lacuna with $n-3$ holes. Thus, the $n-3$ steps described above are required. The first hole emerges after an application of a word of the form (3) (whose length is at least 3), and then one has to alternate the action of a word of length at least $n-3$ with the action of a word of the form (3) $n-4$ times. Therefore,

$$|w| \geq 2 + 3 + (n-4)((n-3)+3) = n^2 - 4n + 5 = (n-2)^2 + 1,$$

as required. Theorem 1.2 is proved.

Fig. 11 shows an example of a dromedary synchronizing automaton with five states and three input letters whose shortest reset word is of length $(5-2)^2 + 2 = 11$, thus exceeding the lower bound of Theorem 1.2. In fact, we have found several examples of five-state automata of this sort.

For automata with six states the bound from Theorem 1.2 also is not tight. An example with an 18-letter reset word is shown in Fig. 12. Even though the automata in Figs. 11 and 12 appear to have a common pattern, we have not yet managed to construct a similar example with seven states.

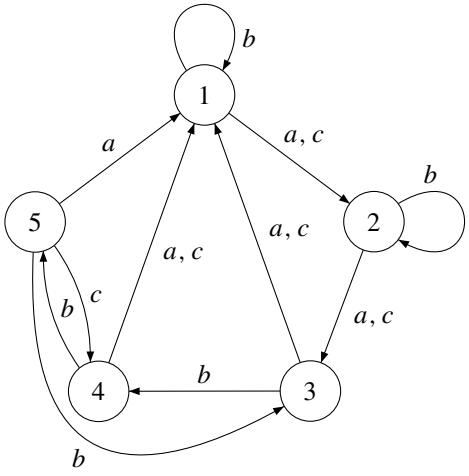


Fig. 11. A five-state automaton with the shortest reset word $ab^2c^2b^2cbc^2$.

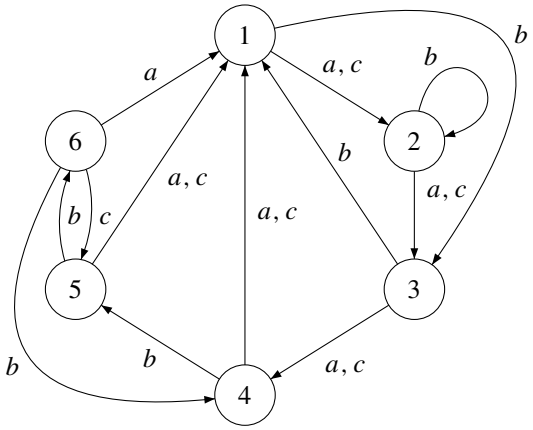


Fig. 12. A six-state automaton with the shortest reset word $ab^2cbacb^2c^2b^5c^2$.

Acknowledgement

This work was supported by the Russian Foundation for Basic Research, grant 05-01-00540.

References

[1] D.S. Ananichev, M.V. Volkov, Synchronizing generalized monotonic automata, Theoret. Comput. Sci. 330 (2005) 3–13.
[2] J. Černý, Poznámka k homogénnym eksperimentom s konecnými automatami, Mat.-Fyz. Cas. Slovensk. Akad. Vied. 14 (1964) 208–216 (in Slovak).
[3] L. Dubuc, Sur le automates circulaires et la conjecture de Černý, RAIRO Inform. Theor. Appl. 32 (1998) 21–34 (in French).
[4] D. Eppstein, Reset sequences for monotonic automata, SIAM J. Comput. 19 (1990) 500–510.
[5] P. Frankl, An extremal problem for two families of sets, European J. Combin. 3 (1982) 125–127.
[6] K. Goldberg, Orienting polygonal parts without sensors, Algorithmica 10 (1993) 201–225.
[7] P.M. Higgins, The range order of a product of i transformations from a finite full transformation semigroup, Semigroup Forum 37 (1988) 31–36.
[8] J. Kari, Synchronizing finite automata on Eulerian digraphs, Theoret. Comput. Sci. 295 (2003) 223–232.
[9] A. Mateescu, A. Salomaa, Many-valued truth functions, Černý's conjecture and road coloring, EATCS Bull. 68 (1999) 134–150.

[10] J.-E. Pin, Le probléme de la synchronisation et la conjecture de Černý, in: A. De Luca (Ed.), Non-commutative Structures in Algebra and Geometric Combinatorics, in: Quaderni de la Ricerca Scientifica, vol. 109, CNR, Roma, 1981, pp. 37–48 (in French).
[11] J.-E. Pin, On two combinatorial problems arising from automata theory, Ann. Discrete Math. 17 (1983) 535–548.
[12] A. Salomaa, Composition sequences for functions over a finite domain, Theoret. Comput. Sci. 292 (2003) 263–281.
[13] S. Sandberg, Homing and synchronizing sequences, in: M. Broy, et al. (Eds.), Model-Based Testing of Reactive Systems, in: Lect. Notes Comput. Sci., vol. 3472, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 5–33.
[14] A.N. Trahtman, The Černý conjecture for aperiodic automata, J. Autom. Lang. Comb. 11 (2006).
[15] A.N. Trahtman, An efficient algorithm finds noticeable trends and examples concerning the Černý conjecture, in: R. Královič, P. Urzyczyn (Eds.), Mathematical Foundations of Computer Science 2006, in: Lect. Notes Comput. Sci., vol. 4162, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 789–800.