Journal of Mathematical Sciences, Vol. 192, No. 3, July, 2013

SYNCHRONIZING RANDOM AUTOMATA ON A 4-LETTER ALPHABET

Yu. I. Zaks* **and E. S. Skvortsov*** UDC 519.713.4

The paper deals with the synchronization of a random automaton that is sampled uniformly at random from the set of all automata with n states and m letters. We show that for $m = 4$, the probability that a random automaton is synchronizing is larger than a positive constant. Bibliography: 9 titles.

1. Introduction and the main result

Let $\mathcal{A} = (Q, \Sigma, \delta)$ be a (*deterministic finite*) *automaton*, where Q is a finite set of states, Σ is a finite input alphabet, and $\delta : Q \times \Sigma \to Q$ is a transition function that defines the action of letters from Σ on states from Q. This action can be naturally extended to an action of words over the alphabet Σ; the result of applying a word w to a state $\mathbf{q} \in Q$ is denoted by $\mathbf{q}w$:

$$\mathbf{q}w = \begin{cases} \mathbf{q} & \text{if } \ w \ \text{ is the empty word,} \\ \delta(\mathbf{q}w', a) & \text{if } \ w = w'a \ \text{ for a word } \ w' \ \text{ and a letter } \ a \in \Sigma. \end{cases}$$

A word w is called a *reset word* for an automaton $\mathcal{A}$ if it sends all states of this automaton to the same state, i.e., $\mathbf{q}_1 w = \mathbf{q}_2 w$ for any $\mathbf{q}_1, \mathbf{q}_2 \in Q$. An automaton $\mathcal{A}$ is called *synchronizing* if it has a reset word.

Synchronizing automata have many applications in different areas: robotics, testing of systems and protocols, symbolic dynamics, etc. (see the surveys [4, 6, 9]). There is a number of interesting open problems concerning them, one of which is to obtain a bound on the length of the shortest reset word in terms of the number of states of the automaton. The best currently known upper bound for an automaton with n states, equal to $(n^3 - n)/6$, was obtained by Pin [5] in 1983. The conjecture, suggested by Černý in the mid 1960s, that this length does not exceed $(n-1)^2$ is proved for some special classes of automata, but remains an open problem in the general case.

In practice, *slowly synchronizing automata*, i.e., automata with the shortest reset word of length $\Theta(n^2)$, are exceptionally rare.[1] From the viewpoint of practical applications of synchronizing automata, it is of importance to study the behavior of the shortest reset word on the average. The results of numerical experiments (see, e.g., [7]) show that it differs substantially from the behavior in extreme cases, which is traditionally the main focus of study in papers on synchronizing automata.

In what follows, by a random variable we mean a discrete random variable, without explicitly mentioning this.

Now we give a rigorous definition of a random automaton and formulate the questions about its synchronization properties we are interested in.

Consider a set of states Q and an alphabet Σ. Pick a transition function δ uniformly at random from the set of totally defined functions $\{\delta : Q \times \Sigma \to Q\}$. The obtained triple (Q, Σ, δ) determines a *random finite deterministic automaton*. It is worth mentioning that a random automaton can be constructed as follows: for each state $q \in Q$ and each letter $a \in \Sigma$, pick $q' = \delta(q, a)$ uniformly at random from Q. By "picking uniformly at random" we mean that every object can be picked with equal probability.

We consider the following problems.

(1) What size of the input alphabet guarantees that almost all automata over an alphabet of this size are synchronizing, and what is the most probable length of the shortest reset word for such automata? (By "almost all automata" we mean a fraction of automata that tends to 1 as $n \to \infty$. We will also say that an assertion satisfied for almost all objects holds "with high probability.")

(2) What size of the input alphabet guarantees that almost all automata over an alphabet of this size are synchronizing and satisfy the Černý conjecture?

(3) What size of the input alphabet guarantees that a random automaton over an alphabet of this size is synchronizing with finite probability? (By "finite" probability we mean a probability bounded from below by a positive constant as $n \to \infty$.)

*Institute of Mathematics and Computer Sciences, Ural State University, Ekaterinburg, Russia, e-mail: yuzaks@gmail.com, skvortsoves@googlemail.com.

[1] For a long time, the only known infinite series of such automata was that constructed by Černý [3]. The first series of slowly synchronizing automata substantially different from Černý's ones were obtained quite recently [1, 2].

Translated from *Zapiski Nauchnykh Seminarov POMI*, Vol. 402, 2012, pp. 83–90. Original article submitted August 25, 2012.

In [8], we gave partial answers to the first two questions for automata with n states and $m(n)$ letters (the number of letters depends on the number of states). In this paper, we address the third question and show that a random automaton with the alphabet size independent of the number of states is synchronizing with finite probability. Our main result is the following theorem.

Theorem 1. *There exists a constant $p_0 > 0$ such that for every positive integer n, the random automaton $\mathcal{A} = (Q, \Sigma, \delta)$ with $|Q| = n$ and $|\Sigma| = 4$ is synchronizing with probability greater than p_0.*

Note that experimental results suggest that in fact a much stronger assertion holds: the random automaton $\mathcal{A} = (Q, \Sigma, \delta)$ with $|Q| = n$ and $|\Sigma| \geq 2$ is synchronizing with probability that tends to 1 as $n \to \infty$. However, at the moment, Theorem 1 is the best result we can prove.

2. The proof of the main result

First we prove the following technical lemma.

Lemma 1. *Let X be a random variable taking values in the interval $[0, 1]$. Then*

$$\mathbf{P}\left(X \geq \mathbf{E}\left(X\right)/2\right) \geq \frac{\mathbf{E}\left(X\right)}{2 - \mathbf{E}\left(X\right)}.$$

Proof. Let c be a constant such that $0 < c < 1$. In the definition of the expectation of X, we estimate the values of X that do not exceed c by c, and the values of X that exceed c, by 1. This yields the following inequality:

$$\mathbf{E}\left(X\right) \leq c\mathbf{P}\left(X \leq c\right) + \left(1 - \mathbf{P}\left(X \leq c\right)\right),$$

or

$$\mathbf{P}\left(X \leq c\right) \leq \frac{1 - \mathbf{E}\left(X\right)}{1 - c},$$

or

$$\mathbf{P}\left(X \geq c\right) \geq \frac{\mathbf{E}\left(X\right) - c}{1 - c}.$$

Taking c equal to $\mathbf{E}\left(X\right)/2$, we complete the proof. □

Let $\mathbf{u}, \mathbf{v}$ be a pair of states of the random automaton $\mathcal{A} = (Q, \Sigma, \delta)$. For this pair, we define a process Roomba whose aim is to find a word $w = a_1 \cdots a_k$ such that $\mathbf{u}w = \mathbf{v}w$.

At the first step of the process, we pick a letter $a_1 \in \Sigma$ at random and jump from $\mathbf{u}_0 = \mathbf{u}$ to $\mathbf{u}_1 = \mathbf{u}_0 a_1$ and from $\mathbf{v}_0 = \mathbf{v}$ to $\mathbf{v}_1 = \mathbf{v}_0 a_1$. If $\mathbf{u}_1 = \mathbf{v}_1$, then the process halts and outputs the word $w = a_1$; otherwise it continues.

At the mth step, we find ourselves in states $\mathbf{u}_{m-1}$ and $\mathbf{v}_{m-1}$. (We may have already visited one or both of them at previous steps.) Choose a letter a_m that has not yet been applied to at least one of the states $\mathbf{u}_{m-1}$ or $\mathbf{v}_{m-1}$. If we can choose such a letter, then we use it to jump from the states $\mathbf{u}_{m-1}$ and $\mathbf{v}_{m-1}$ to the states $\mathbf{u}_m = \mathbf{u}_{m-1}a_m$ and $\mathbf{v}_m = \mathbf{v}_{m-1}a$, respectively, as at the first step. We say that this is a *key jump* of the process. If $\mathbf{u}_m = \mathbf{v}_m$, then the process halts and outputs the word $w = a_1 a_2 \ldots a_m$; otherwise it continues.

If we cannot choose such a letter (which means that we have already applied all letters of the alphabet to each of the states $\mathbf{u}_{m-1}$, $\mathbf{v}_{m-1}$), then we do the following. Perform a breadth-first search for a state to which at least one letter has not been applied and which can be reached from the states $\mathbf{u}_{m-1}$ and $\mathbf{v}_{m-1}$ simultaneously. If it cannot find such a state, our process fails. Otherwise we have found a state that can be reached, say, from the state $\mathbf{u}_{m-1}$ via a word $z \in \Sigma^*$. Jump from $\mathbf{u}_{m-1}$ and $\mathbf{v}_{m-1}$ to $\mathbf{u}_m = \mathbf{u}_{m-1}z$ and $\mathbf{v}_m = \mathbf{v}_{m-1}z$, respectively, and continue the process. This part of the process will be called the *search for a word z*.

Note that the process Roomba is similar to the process Vacuum from [8], the only difference being in the rule for choosing a letter or a word for the next step. A formal description of the process Roomba is given in Fig. 1.

Observe a number of useful properties of the process Roomba applied to an automaton on a two-letter alphabet.

Proposition 1. *Let $\mathcal{A} = (Q, \Sigma, \delta)$ be a random automaton with $|Q| = n$ and $|\Sigma| = 2$. Then the process* Roomba *applied to any pair of states $\mathbf{u}, \mathbf{v} \in Q$ outputs a word w after a key jump with probability $1/n$.*

INPUT: A random automaton $\mathcal{A} = (Q, \Sigma, \delta)$ and a pair of states $\mathbf{u} \in Q, \mathbf{v} \in Q$
OUTPUT: ***Failure*** or a word $w = a_1 \dots a_k$ such that $\mathbf{u}w = \mathbf{v}w$
METHOD:

```
let Δ_q ⊆ Σ, w ∈ Σ*
set Δ_q = ∅ for all q ∈ Q, w = ε
while uw ≠ vw
  if Δ_uw ∩ Δ_vw ≠ Σ, then the key jump
    choose a ∈ Σ \ (Δ_uw ∩ Δ_vw)
    set Δ_uw = Δ_uw ∪ {a}, Δ_vw = Δ_vw ∪ {a}
    set w = wa
  otherwise
    search for a word z
    set w = wz
output w
```

Fig. 1. The process ROOMBA.

Proof. By the definition of a key jump, it involves a letter a_i that has not been applied to at least one of the states $\mathbf{u}_{i-1}$ or $\mathbf{v}_{i-1}$, say $\mathbf{u}_{i-1}$. Hence we pick a state $\mathbf{u}_i$ uniformly at random from Q, and it will coincide with a given state $\mathbf{v}_i$ with probability $1/n$. □

Proposition 2. *There exists a constant $c_1 > 0$ such that for an arbitrary random automaton $\mathcal{A} = (Q, \Sigma, \delta)$ with $|Q| = n$ and $|\Sigma| = 2$, the process* ROOMBA *applied to any pair of states $\mathbf{u}, \mathbf{v} \in Q$ performs a key jump at least $c_1 n$ times with high probability, unless it outputs a word w earlier.*

Proof. First we show that an arbitrary subset of states from Q of size less than n/e has an outgoing edge with high probability. For a fixed set of states of size m with $m < n/e$, the probability that it has no outgoing edge is $(m/n)^{2m}$. The total number of such sets of size m is $\binom{n}{m} \le \left(\frac{ne}{m}\right)^m$. It follows from the Boole inequality (for brevity, the trivial estimate that bounds the probability of the union of events by the sum of their probabilities will be called the *Boole inequality*) that the probability that there exists a set of size m without outgoing edges is less than

$$\left(\frac{m}{n}\right)^{2m} \left(\frac{ne}{m}\right)^m = \left(\frac{me}{n}\right)^m \underset{n\to\infty}{\longrightarrow} 0.$$

Summing over all such m, we see that all sets of size less than $c_1 n$ for some constant c_1 with $0 < c_1 < 1/e$ have an outgoing edge with high probability. It easily follows that from every state $\mathbf{q} \in Q$ one can reach at least $c_1 n$ states with high probability. Moreover, the constant c_1 depends neither on the type of the automaton, nor on the number of its states.

Thus the breadth-first search for a word z at some step of the process ROOMBA will perform at least $c_1 n$ steps, unless it successfully outputs the answer earlier. By the finiteness of all objects, for the process as a whole there are two possibilities: either at a certain moment it performs the required number of steps in the search, or it outputs a word w earlier.

Observe that, by the definition of the search, the paths from $\mathbf{u}_{m-1}$ and $\mathbf{v}_{m-1}$ marked by the word z include only edges that have been used earlier. The fact that we encounter an unknown edge means that we could complete the search with a shorter word z. Thus we use an edge for the first time only in a key jump.

The fact that the search started at some state $\mathbf{u}_j$ has performed $c_1 n$ steps means that in the automaton there are $2c_1 n$ edges we have already examined. All these edges have been once used for the first time, i.e., a key jump has been performed at least $c_1 n$ times. □

The established properties allow us to prove the following lemma.

Lemma 2. *There exist constants $p_0 > 0$ and $c_0 > 0$ such that for every positive integer n and every random automaton $\mathcal{A} = (Q, \Sigma, \delta)$ with $|Q| = n$ and $\Sigma = \{a, b\}$, the probability of the event*

$$\frac{|\{(\mathbf{u}, \mathbf{v}) \in Q^2 \mid \exists w \ \mathbf{u}w = \mathbf{v}w\}|}{n^2} > c_0$$

is greater than p_0. In other words, a finite fraction of pairs of states in the random automaton can be synchronized with finite probability.

Proof. Let us take a pair of states $(\mathbf{u}, \mathbf{v}) \in Q \times Q$ and try to synchronize it by the process ROOMBA. By Proposition 2, a key jump will be performed at least $c_1 n$ times for a constant c_1, unless the process successfully outputs the answer earlier. By Proposition 1, each key jump leads to synchronization with probability $1/n$. The key jumps are independent, so that, applying the Boole inequality over all key jumps, we see that a pair of states can be synchronized with probability bounded from below by a constant c_2.

Therefore, the expectation of the random variable

$$|\{(\mathbf{u}, \mathbf{v}) \in Q \times Q \mid \exists w \ \mathbf{u}w = \mathbf{v}w\}|$$

is greater than $c_2 n^2$. Applying Lemma 1 to the random variable

$$X = \frac{|\{(\mathbf{u}, \mathbf{v}) \in Q \times Q \mid \exists w \ \mathbf{u}w = \mathbf{v}w\}|}{n^2}$$

completes the proof. □

Proof of Theorem 1. Let $\Sigma = \{a, b, d, f\}$. By Lemma 2, with a finite probability p_0 there exists a subset $T \subset Q \times Q$ such that $|T| > c_0 n^2$ and every pair of states from T can be synchronized by a word w_1 over the alphabet $\{a, b\}$. We will show that for every pair of states not from T, there exists a path from it to a pair of states from T.

Consider an arbitrary pair of states $\mathbf{u}, \mathbf{v} \in Q \setminus T$. In the proof of Proposition 2 we showed that the number of states that can be reached from a fixed state with high probability is not less than $c_3 n$ for some constant c_3. Then, starting from the state $\mathbf{u}$ and using words over the alphabet $\{d, f\}$, we can reach at least $c_3 n$ states for some constant c_3. Visit all these states by a breadth-first search, simultaneously making the same jumps from the state $\mathbf{v}$. We will thus obtain $c_3 n$ pairs of states, including at least $\frac{c_3 n}{2}$ distinct ones. The probability that a set of $\frac{c_3 n}{2}$ random pairs is disjoint with T is bounded from above by $(1 - c_0)^{c_3 n/2}$, which tends to 0 as $n \to \infty$. Denote by w_2 the word in the alphabet $\{d, f\}$ that corresponds to the path leading from $\mathbf{u}, \mathbf{v}$ to a pair from T.

Applying the Boole inequality over all pairs of states, we see that with finite probability each pair of states of the automaton will be synchronized either by the word w_1 or by the word $w_2 w_1$. As is well known [3], if every pair of states of an automaton can be synchronized, then the automaton is synchronizing. □

Acknowledgments. The authors are grateful to the anonymous referee for valuable remarks.

The first author was supported by the RFBR grant 10-01-00793.

Translated by N. V. Tsilevich.

REFERENCES

1. D. S. Ananichev, V. V. Gusev, and M. V. Volkov, "Slowly synchronizing automata and digraphs," *Lect. Notes Comput. Sci.*, **6281**, 55–64 (2010).
2. D. S. Ananichev, M. V. Volkov, and Yu. I. Zaks, "Synchronizing automata with a letter of deficiency 2," *Theoret. Comput. Sci.*, **376**, 30–41 (2007).
3. J. Černý, "Poznámka k homogénnym eksperimentom s konečnými automatami," *Matematicko-fyzikalny Časopis Slovensk. Akad. Vied*, **14**, No. 3, 208–216 (1964).
4. A. Mateescu and A. Salomaa, "Many-valued truth functions, Černý's conjecture and roadcoloring," *Bull. EATCS*, **68**, 134–150 (1999).
5. J.-E. Pin, "On two combinatorial problems arising from automata theory," *Ann. Discrete Math.*, **17**, 535–548 (1983).
6. S. Sandberg, "Homing and synchronizing sequences," *Lect. Notes Comput. Sci.*, **3472**, 5–33 (2005).
7. E. Skvortsov and E. Tipikin, "Experimental study of the shortest reset word of random automata," *Lect. Notes Comput. Sci.*, **6807**, 290–298 (2011).
8. E. Skvortsov and Yu. Zaks, "Synchronizing random automata," *Discrete Math. Theor. Comput. Sci.*, **12**, No. 4, 95–108 (2010).
9. M. V. Volkov, "Synchronizing automata and the Černý conjecture," *Lect. Notes Comput. Sci.*, **5196**, 11–27 (2008).