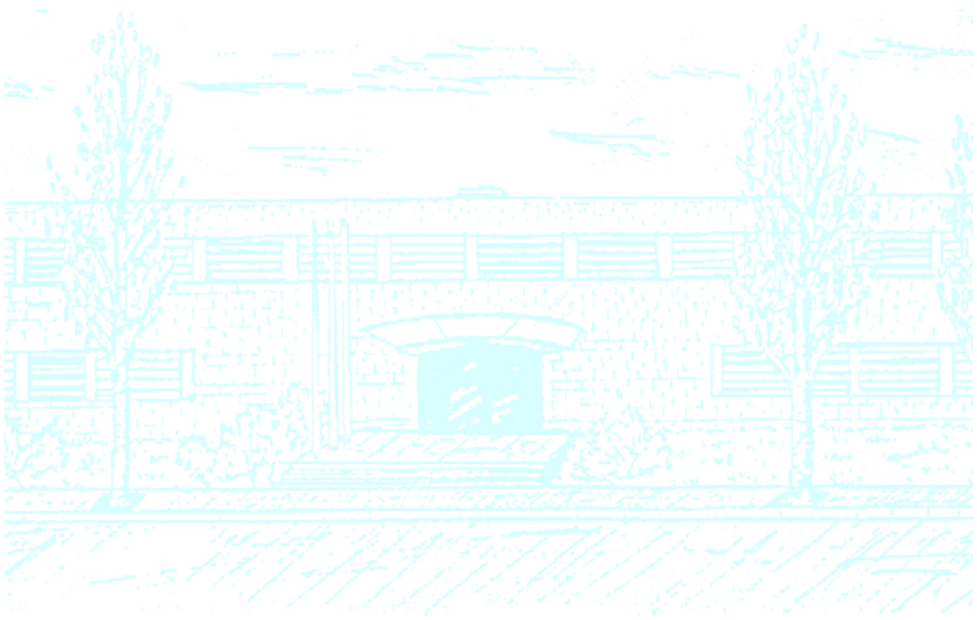
MASTER'S THESIS

Interuniversity Master in Statistics and Operations Research UPC-UB

Title:	**Study of random w-trees and automaton synchronization**
Author:	**Attila Genda**
Advisor:	**Guillem Perarnau Llobet**
Department:	**Faculty of Mathematics and Statistics**
University:	**Polytechnic University of Catalonia**
Academic year:	2022/23



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

UNIVERSITAT DE BARCELONA

Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Master thesis

Study of random w-trees and automaton synchronization

Attila Genda

Advisor: Guillem Perarnau

FME-MESIO

Dedicated to my parents in love and gratitude.

Preface

The following work is written in order to fulfill the requirements of the master degree in Statistics and Operations Research at the Polytechnic University of Catalonia. In the master thesis the synchronization of random deterministic finite automata is investigated making the topic part of the fields of combinatorics, graph theory and stochastic processes. The structure of the work is as follows. In Sec. 1 a short introduction and overview on finite deterministic automata is given. Sec. 2 focuses on the synchronization of random automata using w-trees. In Sec. 3 an algorithm is presented for determining whether the underlying directed graph generated by a word w on an automaton A is a loop-rooted tree, implying that the automaton is synchronizing, or that the underlying graph structure is not a tree, thus w does not synchronize A. Sec. 4 shows several simulation results obtained with the help of the previous algorithm. In Sec. 5 the results of Higgins on random finite mappings are presented in order to prepare the consecutive Markov chains based analysis performed in Sec. 6. In Sec. 7 the results are summarized.

Abstract

Key words: deterministic finite automata, synchronizing word, random tree, Cayley tree

MSC2000: 05C05,05C80

The synchronization properties of repeated words on deterministic finite automata were investigated numerically and analytically in this work. A polynomial ($\mathcal{O}(n^2)$) algorithm for finding short synchronizing words based on w-trees is presented. The algorithm is capable of finding synchronizing words of length proportional to $\sqrt{n \log n}$ with automata of size up to a few hundred thousand states. Experimental results on the height of w-trees is also presented showing that the mean tree height is of $\mathcal{O}\left(\sqrt{\frac{n}{k}}\right)$. Furthermore, the modeling of the random DFA synchronization process by Markov chains is carried out in this study. Three different experimentally justified estimates for the mean shortest synchronizing word length are presented as well indicating that the shortest resetting word length is proportional to the square root of the automaton size.

Notation

$\mathbb{N}$	Natural numbers
$\mathbb{Z}$	Integer numbers
$\mathbb{Q}$	Rational numbers
$\mathbb{R}$	Real numbers
$\mathbb{C}$	Complex numbers
A	Automaton represented by the 5-tuple $\langle \Sigma, \Gamma, Q, \delta, \lambda \rangle$
Σ	Input alphabet of A
Γ	Output alphabet of A
Q	Set of states
δ	Transition function $\delta : Q \times \Sigma \to Q$
λ	Next-output function $\lambda : Q \times \Sigma \to \Gamma$

Contents

1. Introduction

In computer science the concept of an automaton is probably the most fundamental abstraction for a computing device which automatically follows a predetermined sequence of operations. The emergence of the mathematical description of automata can be dated back to 1948 when John von Neumann expressed the need for a theory to describe the possibilities and limits of the already existing electric circuits [**1**]. During the next two decades the new scientific field of theoretical computing emerged having its own research agenda and textbooks. In the beginning, automata theory studied formal languages and the machines that accept them in a quasi-algebraic manner. Later, from the 1960s, formal semantics emerged as a major area of research leading to the development of extensible programming languages using the same space to store values and functions.

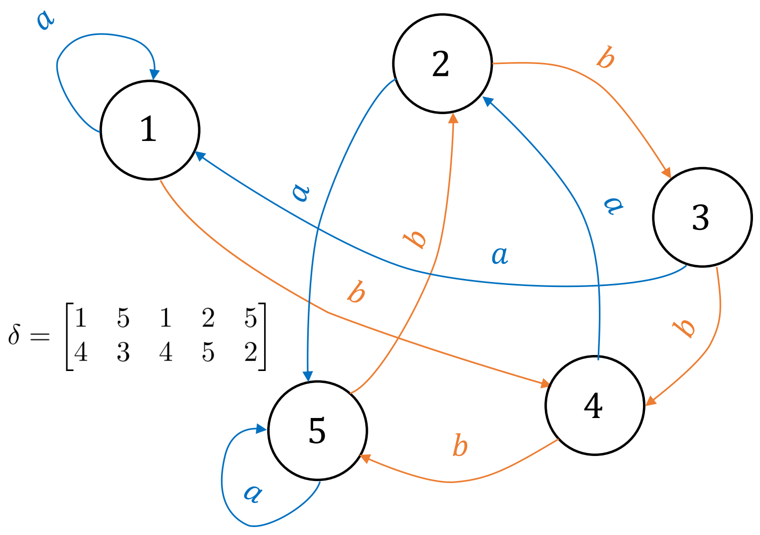
In the beginning automaton theory was considered as a branch of mathematical systems theory dealing with discrete-parameter systems. Instead of using differential calculus as in material systems, abstract algebra was used to describe automata [**2**]. Automata can be defined by restricting the definition of a system by only allowing discrete time-steps. The output of the automaton only depends on it state and input through unchanging function defining the relations within them. The automaton *runs* when processing some input, given in every time step, it passes through its *states* (*transition*) in a way defined by the *transition function.* The input is a *word* consisting of letters from the *input alphabet.* Simultaneously the automaton produces output letters (from the *output alphabet*) using the *output function.* The automaton runs as long as it has read the whole input word. The state where it *halts* is called the *final state.*

Using formal language theory to investigate the possible sequences of states, inputs and outputs a *starting state* and a set of *accepting states* can be assigned to the automaton. Depending on whether the final state is in the set of accepting state, the automaton can *accept* or *reject* a word. The set of all input sequences is called the *language recognized by the automaton* [**3**].

An automaton either has finite or infinite states, and its transitions can either be deterministic or non-deterministic. In this work, we are focusing on deterministic finite automata (DFA).

1.1. Deterministic Finite Automata. Formally, a deterministic finite automaton is a 5-tuple $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ with

- a finite set of states Q,
- a finite set of input symbols Σ (alphabet),
- a transition function $\delta : Q \times \Sigma \to Q$, with natural extension $\delta : Q \times \Sigma^k \to Q$,
- an initial state $q_0 \in Q$,
- a set of accept states $F \subseteq Q$.



$$\delta = \begin{bmatrix} 1 & 5 & 1 & 2 & 5 \\ 4 & 3 & 4 & 5 & 2 \end{bmatrix}$$

FIG. 1. A DFA with five states and two letters. The SSW is *aabaa*. When applied, all states meet in state 5. From synchronizing words of length 6 there exist multiple ones, for example *aabaab* or *baabaa*.

Having a word $w = a_1 a_2 \ldots a_n$ over the alphabet Σ the automaton A accepts the string w if there exists a set of states $r_0, r_1, \ldots, r_n \in Q$ such that the following holds:

- $r_0 = q_0$
- $r_{i+1} = \delta(r_i, a_{i+1})$, for $i = 0, 1, \ldots n-1$
- $r_n \in F$.

A DFA without starting state and without accept states is called a *transition system.*

In the following we will discuss *complete* DFA, i.e. such ones where in every state the transition to the next state is defined for every possible letter of the input sequence. Furthermore, the automata are assumed to be random, i.e., when setting up the automata's transition function every transition from a state to any state, including the current state as well, is equally possible.

1.2. Synchronization of DFA. A *synchronizing word* or *reset sequence* is a word $\omega \in \{a, b\}$ in the input alphabet that sends any state $u \in [n]$ of a DFA to a common final state $v_0 \in [n]$ irrespective of which the starting state was. Of course, not all automata can be synchronized; reasons might be, e.g., that the automaton is not connected or the transitions cause permutations. Finding such a word, especially a short one, is obviously not easy and until now trial and error has proven to be the only possible method. The problem is known to be computationally hard, thus the expected time of finding the shortest synchronizing word (SSW) increases exponentially with the number of states. On the other hand, it is easy to see that an automaton, if synchronizable, has a synchronizing word of length at most n^3: since by the pigeonhole principle every two vertex can be synchronized in at most n^2 steps (selecting two states and building pairs of vertices one finds n^2 possible combinations, thus, any word longer than n^2 would bring the two states into a configuration that has already occurred before), repeated application of the observations n times yields the n^3 bound. Theoretical research on finding the shortest reset word is mainly motivated by Černý's conjecture stating that for every synchronizing automata there is a reset word of length $< (n-1)^2$ [**4**]. Fig. 2 shows an example with 6 states for Černý's automaton. The conjecture has been proven only for a few special cases. Yet, a general cubic bound $(n^3 - n)/6$ has been established [**5, 6**] and has been reduced to $n(7n^2 + 6n - 16)/48$ by *Trahtman* [**7**]. The currently known smallest bound for general DFA is $0.1654n^3 + o(n^3)$ given by *Shitov* [**8**].

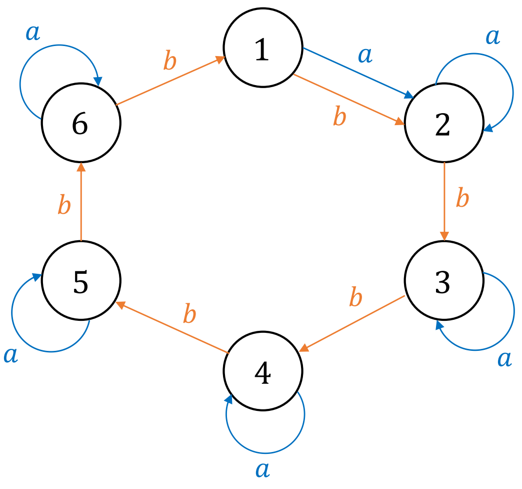


FIG. 2. Černý's automaton for $n = 6$ with SSW of length $(n-1)^2 = 25$. The SSW is $w = abbbbbabbbbbabbbbbabbbbba$. Note, that the occupancy number can be reduced always only by one when states 1 and 2 are occupied and the letter a is applied. Then, the $n-1$-fold application of b results in a configuration where two states can be synchronized again by a. $n-2$-fold application of this process and finally applying a the automaton is synchronized. It is important to note that in the beginning the occupancy number can reduced much faster than this by not applying b $n-1$ times, but only as many times that states 1 and 2 are non empty. However, in this case holes within occupied states emerge that hinder synchronization in the followings. Thus, the occupied states has to be shifted such that they occupy consecutive states. All in all, the length of the synchronizing word cannot be reduced by synchronizing words of different structure than the one described above.

1.3. Synchronization of random DFA. On the other hand, many theoretical and experimental results showed that in many cases DFA have significantly shorter reset words and that with probability tending to 1 all random automata are synchronizable if $n \to \infty$. The latter statement was conjectured by *Cameron*[**9**] and proven by *Berlinkov* [**10**] and quickly after by *Nicaud* [**11**]. *Nicaud* also showed analytically that random DFA admit a synchronizing word of length less than $n^{1+\epsilon}$ with high probability (w.h.p.). *Chapuy* and *Perarnau* have shown analytically that random DFA are synchronizable w.h.p. with some word of length $\mathcal{O}(n^{0.5} \log n)$ [**12**].

Several experimental studies were performed as well. The most natural approach to the problem of finding a synchronizing word is the breadth-first-search method. This method starts from the set of all states of the automaton and forms images based on the letter transformations until a single state is reached. There exists also computational packages utilizing this idea, e.g., COMPAS [**13**] and TESTAS [**14**].

To find short synchronizing words of random automata *Roman* used a genetic algorithm and gave a linear bound for the shortest reset words [**15**]. The paper starts with the reformulation of the problem of finding the SSW as an optimization problem in the word length within all possible words over an alphabet A. In the terms of genetic programming a word is understood as a chromosome and letters represent genes. But since words can have different length, the chromosome length is not fixed, which has important consequences when defining crossover and mutation. Adaptation methods for changing mutation and crossover probability during the algorithm work are used. Also, adaptation is used when modifying the probability distribution $\mathcal{P}(A)$ over an alphabet, which is used in the initiation and also in the mutation phase. The number of chromosomes is N, of which always two pass into P^t from P^{t-1} based on elitist selection. The rest $N-2$ chromosomes are selected based on the roulette wheel method. The "best" chromosomes to be selected are the shortest word with the highest deficiency, i.e., the difference between the size of the DFA and the number of occupied states after the word w has been applied. In Fig. 3 the findings are shown.

Skvortsov described an approach to find minimal reset words using a Boolean satisfiability (SAT) solver [**16**]. The method consists of performing a binary search on sub-problems of finding out whether an automaton of n states can be synchronized in c steps. Then, the smallest value of c means the SSW. The experiment performed multiple times yields a mean value for the minimal length of reset words for automata of size n. The algorithm, not like the previous, genetic one, is exact, thus it not only finds an upper bound on the shortest reset word, but indeed the optimal solution. In return, it is an exponential algorithm, thus with increasing number of states its convergence slows down rapidly. The method was applied to perform an experimental study on automata of maximal 100 states. An estimate of $1.95n^{0.55}$ was given for the shortest reset words.

Kisielewicz et al. introduced an algorithm able to find reset words of minimal length of random automata with up to a few hundred states [**17**]. The main idea of the method is to use bidirectional breadth-first search (BFS) and radix (Patricia) trees to store and compare subsets. *Kisielewicz et al.* also found that the expected length of the shortest resetting word is sublinear with the number of states and gave the

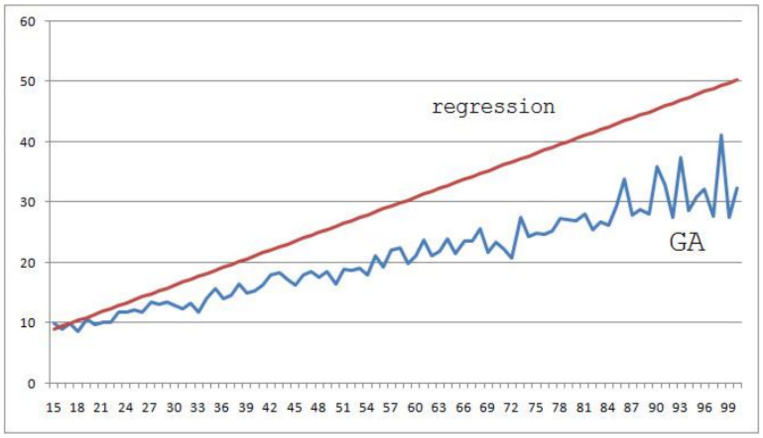


FIG. 3. Figure taken from [**15**]. The mean minimal synchronizing word length depicted against the number of states. A linear estimate $y = 0.486x + 1.654$ obtained for small automata with a sampler size of 1000 (red line) is extrapolated for larger automata and compared to the results found by the genetic algorithm (sample size is 10, blue line).

estimate $\approx 2.5\sqrt{n-5}$ for the shortest resetting word. The algorithm presented in the paper first decides whether the underlying automaton is synchronizing or not, which can be performed in polynomial time [**18**]. Then, they perform the BWS on the power automaton of A starting from the set Q of all states applying letters of the alphabet Σ. It is also possible to search backwards from a single state and apply preimages of the of letters, which is called inverse breadth-first search. The branching factor is k ($|\Sigma|$) in both cases, thus, the cumputational cost is $\mathcal{O}(k^l)$ or $\mathcal{O}(nk^l)$ for IBFS. The bidirectional search uses two searches simultaneously and check if they meet, thus, reducing the computational cost to $\mathcal{O}(nk^{l/2})$. To be able to implement the idea an effective algorithm is necessary to check whether each new subset has already appeared in the search tree of the other half of the search.

Szykuła et al. improved speed of the algorithm significantly by using more efficient data structures, decision mechanisms, and reduction procedures [**19**]. Additionally, the improved algorithm is also capable to be used in multithreading and GPU computing. Using the algorithm the authors were able to find the SSWs of DFA of up to 570 states. Fig. 4 shows the numerical results and the readjusted estimate $2.284n^{0.515}$, which for large automata predicts bigger values than the estimate $2.5\sqrt{n-5}$ found in [**17**] by investigating smaller automata.

The change in the estimate for the expected minimal length of synchronizing words highlights the problem of this kind of numerical experiments: the estimate is only valid in the region, where the fitting was performed. The extrapolation of the fit to automata with much larger number of states guarantees no exact results. Clearly, to give a reliable value for the expected value of the SSW a better understanding of the synchronization process is needed.

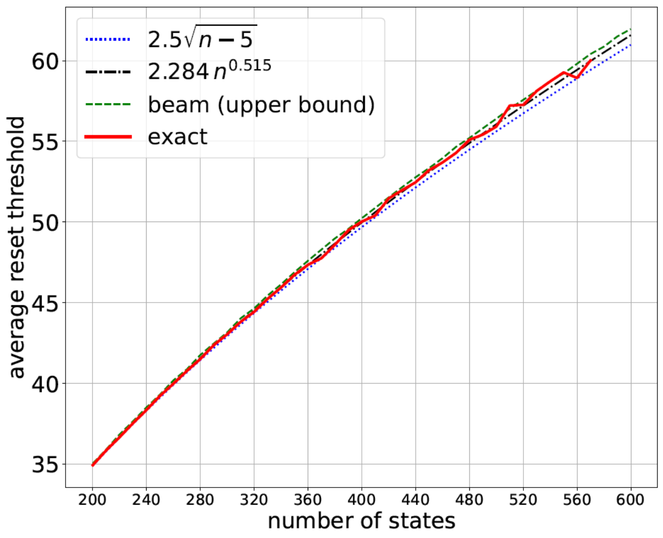


FIG. 4. Figure taken from [**19**]. The mean reset threshold of binary automata with n states. For every $n \in \{5, 10, ..., 300\}$, $n \in \{305, 310, ..., 400, 410, 420, ..., 500\}$, and $n \in \{510, 520, ..., 570\}$ they calculated resp. 10 000, 1 000, and 100 automata.

2. Random w-trees and automaton synchronization

Nicaud investigated random DFA by analytical means and proved that a random DFA with an alphabet at least with two letters can be synchronized w.h.p. by a word of length less than $n^{1+\epsilon}$. By that it is also shown that Černý's conjecture hold w.h.p. for random DFA.

Chapuy and *Perarnau* investigated the synchronization of random, 2-letter DFA with words ω that consists of the repeated concatenation of short word chunks w, i.e., $\omega = w|w|...|w|w$ [**12**]. The application of w-transitions might induce a (loop-rooted) tree. They showed, that if $|w| = (1+\epsilon)\log_2(n)$ then w.h.p. there exists a ω of the above kind for which the automaton synchronizes and the length of the synchronizing word $|\omega|$ is of $\mathcal{O}(n^{0.5}\log n)$.

THEOREM 1 ([**12**]). *A uniformly random automaton with n states on a 2-letter alphabet has a synchronizing word of length at most $C\sqrt{n}\log n$ w.h.p., where C is an absolute constant.*

Given an automaton A and a word $w \in \{a, b\}^*$, we let δ_w be a the function $[n] \to [n]$ induced by w-transitions (a one letter automaton). We say that a function is a loop-rooted tree if it has a unique cyclic point, i.e., if there exists a unique $x \in [n]$ such that $x \in \{f^k(x), k \geq 1\}$. See Fig. 5-Right. If δ_w is a loop-rooted tree, we say that A is a *w-tree*. The main technical result of [**12**] says that almost all automata are w-trees for a very short word w: it is enough if its length is slightly greater than $\log_2 n$.

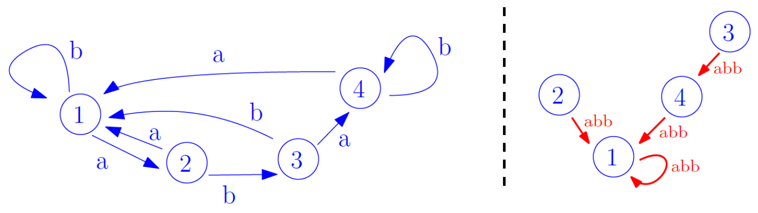


FIG. 5. Figure taken from [**12**]. Left: An automaton A with $n = 4$ states. A is synchronizable, since the word $\omega = abbabb$ is synchronizing: it sends all the states to the state $v_0 = 1$. Right: the one-letter automaton δ_w induced by w-transitions, for $w = abb$. It has a unique cyclic vertex, i.e., it is loop-rooted tree (equivalently, A is w-tree).

THEOREM 2 ([**12**]). *Let A be a uniformly random automaton on n states, and let $\epsilon > 0$. Then, w.h.p., there exists a word w of length at most $(1+\epsilon)\log_2(n)$ such that A is a w-tree.*

The idea of the proof is finding a word w of length $(1+\epsilon)\log_2(n)$ such that δ_w is a loop-rooted tree, then it is clear that the word $\omega = w^H$ is synchronizing, where H is the height of that tree. It is easily shown that the height of the tree $H \leq C\sqrt{n}$ w.h.p.. Thus, the length of the word is of $\mathcal{O}(\sqrt{n}\log(n))$.

The authors of [**12**] suspect that the order of the expected value of the SSWs can be reduced to $\mathcal{O}(\sqrt{n\log(n)})$. The reason for such an expectation is based on the following theorem:

THEOREM 3 ([**12**]). *Assume that there is a random variable X_n such that the following holds. For any non self-conjugated word w of length $(1+\epsilon)\log_2(n)$, the height H_n (maximum distance of a vertex to the root) of a uniform random 0-shifted marked w-tree is stochastically dominated by X_n. Then the SSW of a random automaton is stochastically dominated by $(1+\epsilon)X_n\log(n)$.*

Chapuy and *Perarnau* suspect that there exists such X_n satisfying $X_n = O_p(\sqrt{\frac{n}{k}})$, where O_p denotes a big-O in probability.

2.1. Heuristics: one-letter automata. The idea presented in [**12**] is inspired by the well known problem of the synchronization of random *one-letter* automata, i.e., of random functions $f : [n] \to [n]$.

A function $f : [n] \to [n]$ divides the set $[n]$ into a set of *cyclic points* $S \subset [n]$, restricted to which the function f is a permutation (forming a set if directed cycles), and the set $[n]\backslash S$ of remaining points, which forms a forest of directed trees attached to the cycles of S, see Fig. 6. The number of cyclic points is at least one, and if $|S| = 1$ we say that the function is a *loop-rooted tree.*

Loop-rooted trees are identical to rooted labeled trees (Cayley trees) on n up to the deletion of the loop at the root. The number of these trees is known to be n^{n-1}

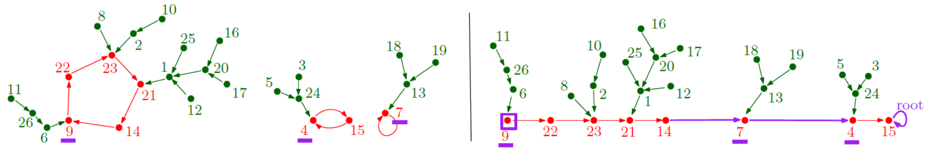


FIG. 6. Figure taken from [**12**]. Left: A function $f : [n] \to [n]$. The set of cyclic points is in red. Right: the function f transformed into a doubly-marked tree via the Joyal bijection. Cycles of f have been cut prior to their minima (underlined), and concatenated by decreasing minima. One obtains a tree with a root (here 15) and a marked vertex (here 9). Lower records on the branch are underlined, they correspond to the cycle minima of the function f (making the construction reversible).

[**20**]. The number of functions $f : [n] \to [n]$ chosen uniformly at random is n^n, thus the probability that it is a loot-rooted tree is

$$\frac{n^{n-1}}{n^n} = \frac{1}{n}, \tag{1}$$

thus, the probability that a unique letter can synchronize the automaton goes to 0 as the size of it goes to infinity. However, conditionally that the automaton is synchronizable, it can be synchronized by the word a^H, where H is the height of the tree. It is well known, that the height of a random tree is of order $O(\sqrt{n})$ w.h.p. ([**21, 22**]).

Before describing the main ideas of the proofs of [**12**] let us prove two simple theorems.

THEOREM 4. *A random DFA synchronizes by the repetition of the word w if and only if the application of w generates a loop-rooted tree.*

PROOF. a) Assume a word w synchronizes. Then, there are no cycles of length greater than 1 in the function f (otherwise contradiction to the assumption). If there are no loops, there can be only one tree or more non-connected trees. In the latter case we have a contradiction to our assumption, since $|S|$ is at least 2. Thus if a word w synchronizes it is a loop-rooted tree.
b) Assume there is a loop-rooted tree. By the definition of a tree, there are no loops in f. Since there is a loop-rooted tree, the directed graph is connected. Then, at most in $n-1$ steps all states are mapped to the root by w. Thus w synchronizes. □

DEFINITION 1. *Two words are said to be conjugate if they differ only by a cyclic permutation.*

THEOREM 5. *If A is synchronized by w then A is synchronized by all conjugates of w as well.*

PROOF. Given that w synchronizes, the root cycle C_0 consists of $|C_0|$, possibly repeated states. Let $\tilde{w}$ be a conjugate of w. The proof is based on contradiction.

Assume $\tilde{w}$ does not synchronize A. Then the application of $\tilde{w}$ results in either a disconnected graph, or in a connected graph with a cycle of length greater than 1.

In the first case, the disconnected graph defined by $\tilde{w}$ has at least two cycles, C_1 and C_2. One of them (C_1) might even be the the same as C_0, however, the other cycle C_2 is distinct. But since $\tilde{w}$ is a cyclic permutation of w, the application of w, starting at the appropriate state of C_2 also leads to the same loop C_2. Which is a contradiction, since w generates a loop-rooted tree having only one cycle C_0.

In the second case $\tilde{w}$ generates a connected graph with a cycle C_3 longer than one. w, being the conjugate of $\tilde{w}$, started at the right state also possesses the same loop C_3. But since w has only one loop C_0, a contradiction is obtained. Thus, having covered all cases, the proof is complete. □

The main ideas of the proofs of [**12**], as formulated in the original paper, are the followings:

(1) one can hope that, for a relatively short (logarithmic) word w, and a random automaton A, the one-letter automaton δ_w shares some qualitative properties with a random function f. For example, one can expect that the probability that it is a tree is of order roughly $1/n$.
(2) if we take two words $w_1 \neq w_2$ that are not conjugate, we can hope that the facts that δ_{w_1} and δ_{w_2} are trees, are somehow independent. This intuition relies on the fact that, from a given vertex v, the w_1-transition and the w_2-transition outgoing from v, although they may share some underlying transitions of A, are unlikely to coincide and thus close to being independent.
(3) consider the set of all words of length $(1+\epsilon)\log_2(n)$, for $\epsilon > 0$. The number of such words is $n^{1+\epsilon} \gg n$. From (1), we can hope that, in average, roughly $n^\epsilon \gg 1$ of these words w are such that A is a w-tree. Moreover, from (2) we can hope that this average value is in fact a typical value. That would imply in particular that the number of such words goes to infinity w.h.p., and in particular, that it is nonzero.

Part 1

Synchronization with w-trees

3. A numerical algorithm for finding w-trees

The main advantage of looking for short synchronizing words in the form of w^H for some short word w is the small amount of possible words, thus, the low computational cost. Since, based on [**12**] w.h.p. a word w of length $l := \lceil (1+\epsilon)\log_2 n \rceil$ exists, such that $\omega = w|w|...w|w$ synchronizes A. The number of all possible w is then roughly n, so even for very large automata the number of possibilities is quite limited and a full search can be performed over the n possibilities. On the other hand, the application of w on all states Q defines a directed graph with each vertex having outdegree one. Thus, the whole structure of the graph might be explored in n steps. However, by Theorem 4 only loop-rooted trees are synchronizing, and to verify whether the underlying structure is a loop-rooted tree also at most n steps are needed. Yet, to see that a w-transition does not result in a loop-rooted tree much less steps might be enough.

In the following, we use the notation $\delta(S_l, w) = S_{l+1}$, i.e., the application of w brings state S_l to S_{l+1}. Starting in a state S_1 and applying the w-transition $m \leq n$ times the states will start to repeat themselves. If the first repetition satisfies $S_{m+1} \neq S_m$ the graph has a cycle longer than one, thus, it cannot be a loop-rooted tree, the word w can be rejected and $n - m$ operations are spared. If, however, $S_{m+1} = S_m$ the search has to continue starting from a state S_{m+2} which is not contained in the set $S_1, S_2, \ldots, S_{m+1}$. For this, the storage of the already investigated states is necessary. Applying the mapping w to S_{m+2} at most $n - m$ times leads finally to the attachment of the new branch to one of the states in $S_1, S_2, ..S_{m+1}$ (in which case the search continues with a state not contained in the set of previously visited states in the same manner until no states are left), or to the attachment of the chain to itself, which means that the graph is not connected, thus not a w-tree, so it is not synchronizing. This way, the search can be interrupted at any moment, and the number of w-transitions reaches n almost only if the directed graph is indeed a w-tree. In which case, as a byproduct of the procedure, the tree-structure along with the height of the H tree is also obtained. The length of the synchronizing word is then given by $H \times l$.

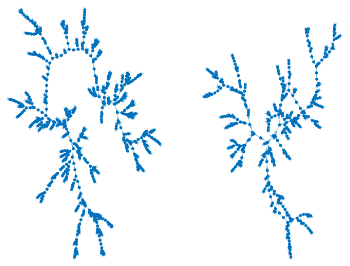
The pseudocode of the algorithm is given in Algorithm 1. Since the algorithm is linear with in n and the number of possible short words w changes also linear with n, the finding of the SSW of the form $w|w|...|w|w$ is proportional to n^2. This algorithm, in comparison to exact ones which are exponential in n, is very fast and can be used effectively even for automata with many hundreds of thousands of states. In contrast, the best exponential algorithm [**19**] known until now can efficiently handle automata of less then 1000 states. The algorithm was implemented in MATLAB, however, implementing it in e.g. C++ would probably increase its speed by several factors. The parallelization of the algorithm is also easy since the processes after a word w gets fixed are independent from each other.

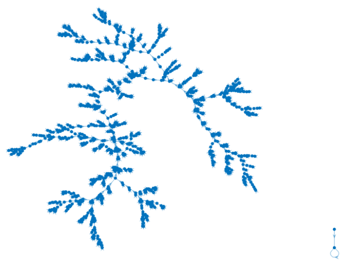
Algorithm 1 Decision algorithm whether $A(Q, \delta, \Sigma)$ is a w-tree

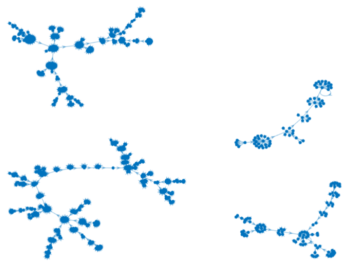
```
Synchable ← 1
$SV \leftarrow \emptyset$                                   ▷ Set of visited states
$SN \leftarrow \bigcup_{i=1}^{n} Q_i$                        ▷ Set of non-visited states
$Q_c \leftarrow Q_1$                                         ▷ Current state
loopFound ← 0
while loopFound = 0 do                                       ▷ Finds the first loop in the graph
    $Q_p \leftarrow Q_c$                                     ▷ Previous state
    $Q_c \leftarrow \delta(Q_c, w)$                          ▷ $w$-transition
    if $Q_c = Q_p$ then
        loopFound ← 1
    else if $Q_c \in SV$ then
        Synchable ← 0
        return Synchable
    end if
    $SV \leftarrow SV \cup Q_c$
    $SN \leftarrow SN \backslash Q_c$
end while
while $|SN| > 0$ do                                          ▷ Checks if the graph is a tree
    Attached ← 0
    $Q_c \leftarrow SN_1$
    $SB \leftarrow Q_c$                                      ▷ Set of states in the current branch
    while Attached = 0 do
        $Q_c \leftarrow \delta(Q_c, w)$
        if $Q_c \in SB$ then                                 ▷ New branch attaches to itself
            Synchable ← 0
            return Synchable
        else if $Q_c \in SV$ then                            ▷ New branch attaches to the main branch
            Attached ← 1
            $SV \leftarrow SV \cup SB$
            $SN \leftarrow SN/SB$
        else                                                 ▷ New branch visits a new cell
            $SB \leftarrow SB \cup Q_c$
        end if
    end while
end while
return Synchable                                             ▷ Is only reached if the graph is a $w$-tree
```

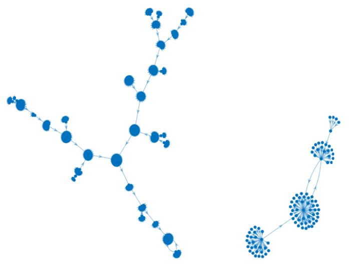
4. Simulation results

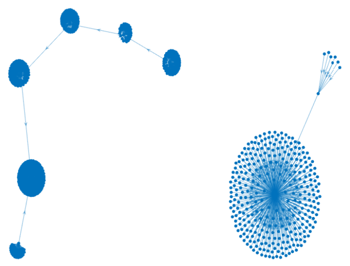
In this section we present several simulation results obtained using Algorithm 1. To obtain a first impression on the structure generated by w-transitions several plots are shown in Fig. 7 for an automaton with $n = 1000$ states under the repeated application of words w with different length.


(A) $|w_k| = 1$


(B) $|w_k| = 2$


(C) $|w_k| = 10$


(D) $|w_k| = 50$


(E) $|w_k| = 200$

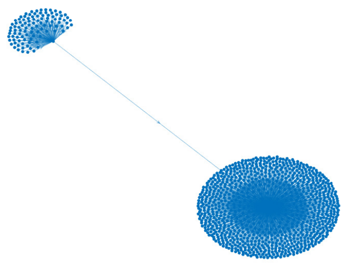

(F) $|w_k| = 1000$

FIG. 7. Graph structure generated by the repeated application of a random word w of length k. In each case the size of the automaton is $n = 1000$.

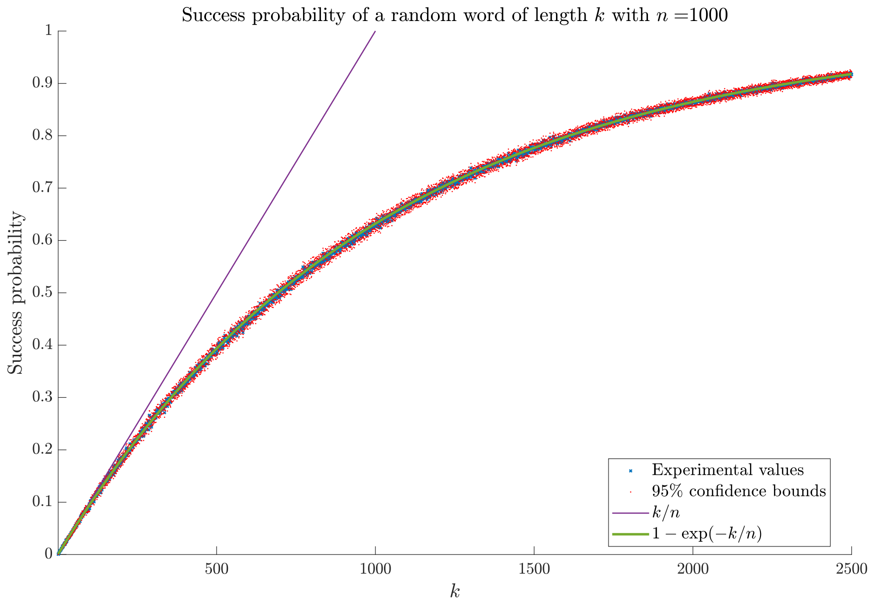


FIG. 8. The probability of synchronization of a randomly selected and repeated word w of length k on 10000 randomly selected automata of size $n = 1000$. The mean probability values for each k are depicted with blue dots. The confidence bounds are given with red; they can be obtained by assuming that the synchronization probability follows a Bernoulli distribution with $p(k)$.

4.1. Synchronizing probability of a repeated random word. A possible experiment to get a better understanding of the structure of automata synchronization with w-trees is the investigation of the probability that a repeated word w of length k synchronizes. For $k = 1$ the result is already known and mentioned in Eq. (1). However, no analytic results are known for $k > 1$. Fig. 8 shows the results of a numerical experiment in which the size of the random automaton n was fixed and a random word w of length k was repeated until it became clear whether the automaton synchronizes or not. The experiment was repeated 10000 times for every value of k from 1 to 2500 and the mean value for synchronization $\hat{p}(k)$ along with its 95% confidence bound was calculated. Surprisingly, the function

$$\tilde{p}(k; n) = 1 - \exp(-k/n) \tag{2}$$

fits $\hat{p}(k)$ with a very high accuracy. Its Taylor expansion around zero k/n shows that for $k = 1$ the result is indeed $p(k = 1) = 1/n$, as the theory also predicts. It would be interesting to obtain a heuristic argument about this problem.

In Fig. 9 the role of k and n are switched. In this case a randomly selected word k of fixed length 50 is applied repeatedly on random automata of varying size n. For each value of n, 10000 random automata are generated and evaluated. The numerical results show, that the previously found function $p(k, n)$ fits the numerically obtained estimate of the probability $\hat{p}(n)$ with a good accuracy once the automaton size n is large enough.

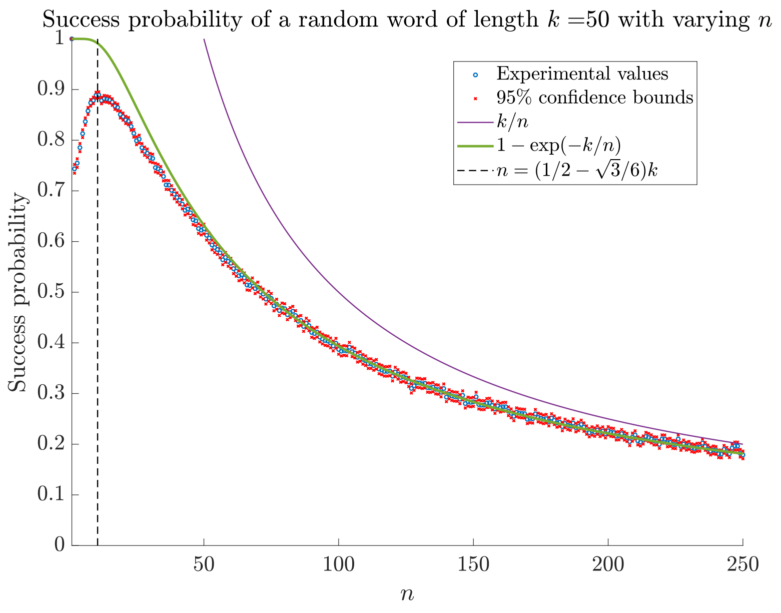


FIG. 9. The probability of synchronization of a randomly selected and repeated word w of length $k = 50$ depicted against the automaton size n. For each value of n, 10000 randomly selected automata were evaluated. The mean probability values $\hat{p}(k)$ are depicted with blue circles. The confidence bounds are given in red. The function $1 - \exp(-k/n)$ fits $\hat{p}(n)$ with a a good accuracy if the size of the automaton gets larger. The hyperbole k/n converges to the experimental results only for very large values of n. The value $(1/2 - \sqrt{3}/6)k$ estimates the n value where the function $1 - \exp(-k/n)$ has its highest curvature. Around this value the numerically obtained probabilities tend to have their maximum as well.

4.2. Probability of the existence of a synchronizing word. In the previous section we investigated the synchronizing probability of a randomly selected repeated word w of length k. However, the probability that any of the possible 2^k word w of length k will synchronize when repeated is significantly larger. Conditioned on the conjecture that the probability of synchronization of a two-letter random automaton by the repeated application of a random word w of length k is given by Eq. (2) the following result can be obtained.

THEOREM 6. *For increasing n and k, assume that w.h.p. a random word w of length k by its repeated application synchronizes a random two-letter automaton A of size n with probability $p(k, n) = 1 - \exp(k/n)$. Further assume that*

$$P(|\delta^{(\infty)}(Q, w_1)| = 1, |\delta^{(\infty)}(Q, w_2)| = 1) = P(|\delta^{(\infty)}(Q, w_1)| = 1)P(|\delta^{(\infty)}(Q, w_2)| = 1)$$

given w_1 and w_2 are not conjugates (otherwise $P(|\delta^{(\infty)}(Q, w_1)| = 1, |\delta^{(\infty)}(Q, w_2)| = 1) = P(|\delta^{(\infty)}(Q, w_1)| = 1)$), i.e., whether w_2 can synchronize A is independent of the fact if w_1 synchronizes A or not. Then, the probability that there exists a random *word w of length k which synchronizes A approaches*

$$p_S(k,n) = 1 - \exp\left(-\frac{2^k}{n}\right). \tag{3}$$

Note, that for sufficiently large k the assumption that two random words $w_{k,1}^{H_1}$ and $w_{k,2}^{H_2}$ will synchronize with the same probability holds w.h.p.

Proof. First, we tackle the problem of self-conjugated words, as these have an inner repetition, thus reducing the value of k to one of its divisors. Knowing that the proportion of the number of self-conjugated words of length k to the number of non-self-conjugated words tends to 0 with increasing k, we can neglect theses cases in an asymptotic analysis. By the assumption that the synchronizing probability of each random word is independent from the rest of the random words (except of its conjugates) and by assuming that for the "average" random word Eq. (2) holds, and making it clear that these words represent the vast majority of all possible words when k increases, we can write that the probability of A not being synchronizable by any of the 2^k words is

$$\begin{aligned} P(\nexists w_k \big| |\delta^{(\infty)}(Q,w_k)| = 1) &= (1-p(k,n))^{\frac{2^k}{k}} \\ &= \left(1-\left(1-\exp\left(-\frac{k}{n}\right)\right)\right)^{\frac{2^k}{k}} \\ &= \exp\left(-\frac{2^k}{n}\right) \end{aligned} \tag{4}$$

where the exponent $\frac{2^k}{k}$ takes into account the fact that if a word synchronizes then all of its conjugates are synchronizing as well (see Theorem 5). Thus,

$$P(\exists w_k \big| |\delta^{(\infty)}(Q,w_k)| = 1) = 1 - \exp\left(-\frac{2^k}{n}\right). \tag{5}$$

□

Fig. 10 shows the comparison of formula (3) with numerically obtained values.

4.3. Syncronizing probability of individual words. Whether the assumption of Proof 4.2 with regard to the equal probability of synchronizing of random words is a good approximation can be investigated experimentally as well. In the following experiment the size of the automaton are fixed to $n = 1000$ and the length of w is chosen to be $k = 64$. 8 different words are investigated whether they generate a w-tree. The first two words were generated randomly, however, the rest of the words were chosen based on different strategies.

Table 1 shows the definition of words $w_1 \ldots w_8$. The last two words are only different at the last entry. w_2 is self-conjugate and has an effective length of $k = 2$, whereas w_7 has still effective length $k = 64$. This fact is highlighted evidently also in the proportion of cases w_8 generates a w-tree from the automaton. Fig. 11 shows the numerically found probabilities of each word to generate a w-tree. Also the experimentally obtained fit given by Eq. (2) is compared to the cases showing

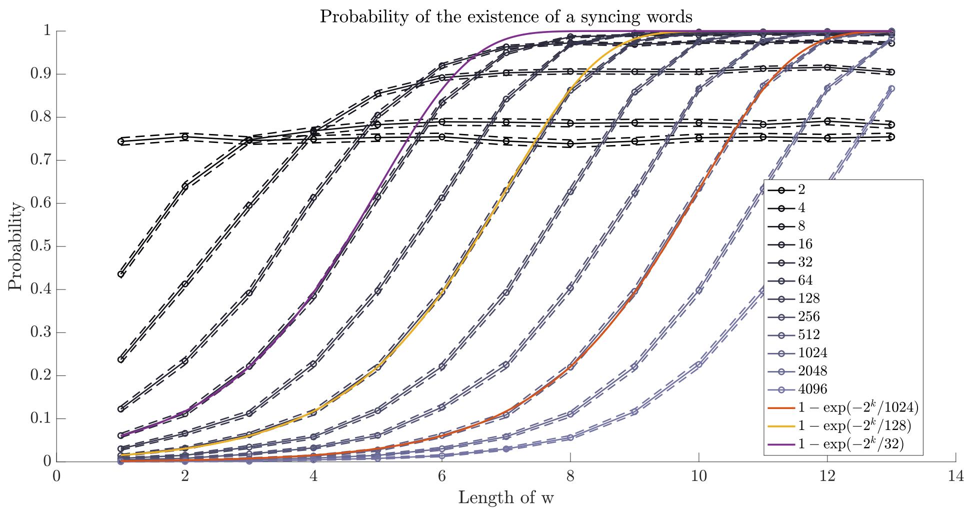


FIG. 10. Probability that a random two-letter automaton of size n is synchronizable by the repeated application of any random word w of length k. Numerically found mean probability values for $n = 2^1, 2^2, ..2^{12}$ along with their 95% confidence intervals are presented in gray-bluish colors. The analytic formula (3) is also evaluated for the values $n = 32, 128$ and 1024. For large values of n and k a good correspondence can be observed.

a good agreement with the first two words, which were generated by MATLAB's pseudo number generator. Somewhat surprisingly, the words w_4-w_7 having low entropy, but yet being non-self-conjugate have significantly higher synchronizing probability than randomly selected words. Word w_3 possesses a pattern as well, however, less trivial than words $w_4 - w_7$ and its synchronizing probability is almost the same as of the random words. This observation, might lead us to the idea to formulate a conjecture about the higher synchronizing probability of low-entropy, non-self-conjugated words. In the following experiment, though, it will be shown, that the conjecture does not hold in general.

$w_1 =$	1221121221121222111222121121222212221112221112112121121112111121
$w_2 =$	2212212121121211121121122111112221211111111112221112221122111121
$w_3 =$	1211211121111211111211111121111111211111111211111111121111111111
$w_4 =$	1111111111111111111111111111111111111111111111111111111111111112
$w_5 =$	1111111111111111111111111111111111111111111111112222222222222222
$w_6 =$	1111111111111111111111111111111122222222222222222222222222222222
$w_7 =$	1212121212121212121212121212121212121212121212121212121212121211
$w_8 =$	1212121212121212121212121212121212121212121212121212121212121212

TABLE 1. Words used in the experiment. 1 and 2 corresponds to the letter a and b, respectively

4.4. Synchronizing with low-entropy words.

To investigate the performance of low-entropy, non-self-conjugated words, we choose w based on the scheme of w_4,

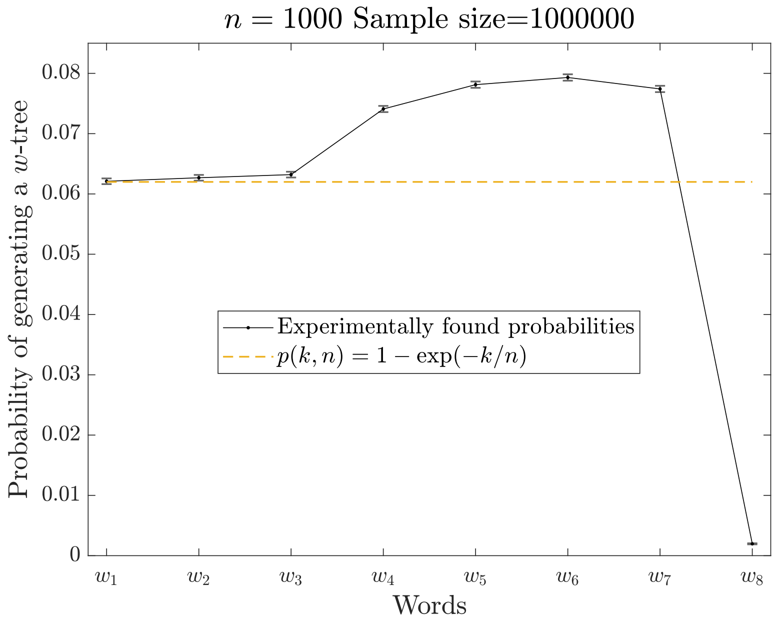


FIG. 11. Synchronization probability of words of length $k = 64$ on 10^6 different random automata.

i.e., we choose every latter to be a, except the last one. In Fig. 12 convergence to a certain probability value way below 1 can be observed. However, the synchronizing probability is greater than the expected in the range of $k = 20 \dots 80$. From that on the probability does not increase further with increasing k. The results of applying the different, but still simple scheme $w = 1212...121(1)$ is shown in Fig. 13. The same effect can be observed as before, however, the interval where the probability exceeds the prediction of Eq. (2) is a bit different, it lies in the range $k = 30..100$.

Remark. The interesting range of k in Eq. 3 is where $\mathcal{O}(\frac{2^k}{n}) = 1$, i.e., $\log_2(k) \approx n$. In this range of values for k the synchronizability of automata is still not affected by the effect of low-entropy words, which generally starts at somewhat larger values, seemingly of order $\mathcal{O}(\sqrt{n})$. Furthermore, the cardinality of such low entropy words becomes negligibly small with respect to the total amount of possible words.

The experiment implies that trying to synchronize by the repeated use of a low-entropy word instead of a random word is only advantageous if the word length is rather short, but not too short. Otherwise, based on the approximation $1 - \exp\left(-\frac{k}{n}\right)$ a sufficiently long random words seems to synchronize the automaton almost surely. This observation will be analyzed further by means of numerical simulations in the next section.

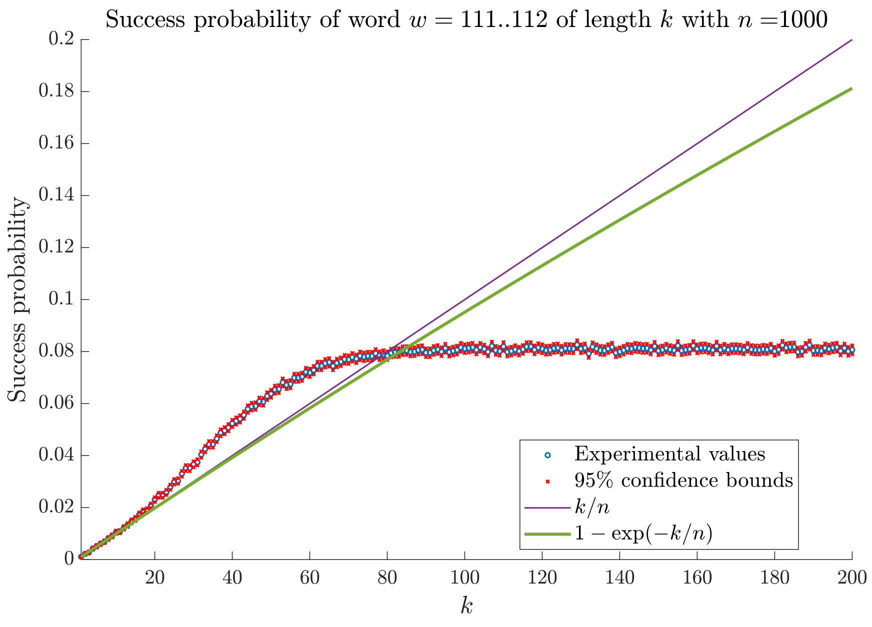


FIG. 12. Synchronization probability of a word in the form $w = 111...112$ of varying length k on 10^5 different random automata for each value of k.

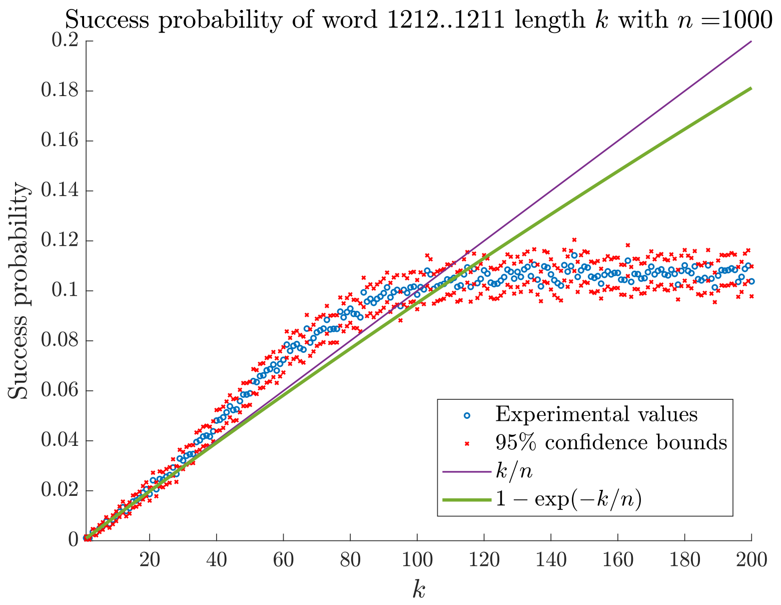


FIG. 13. Synchronization probability of a word in the form $w = 1212...121(1)$ of varying length k on 10^4 different random automata for each value of k.

4.5. Experimental study of the height distribution of w-trees. Knowing the height H of a w-tree immediately implies the knowledge of the length of the reset word $H \times |w|$ associated with the tree. For $k = 1$ the generated trees are random trees whose height distribution is known [**21**]. Let $\mathcal{H}_n$ be the random variable denoting the height of a random loop-rooted tree with n vertices generated by a single letter, i.e., $k = 1$. Then

$$\lim_{n\to\infty} P\left(\frac{\mathcal{H}_n}{\sqrt{2n}} < x\right) = H(x) \tag{6}$$

where

$$\begin{aligned} H(x) &= 4x^{-3}\pi^{\frac{5}{2}} \sum_{p=1}^{\infty} p^2 \mathrm{e}^{-\pi^2 p^2/x^2} \\ &= \sum_{\nu=-\infty}^{\infty} \mathrm{e}^{-\nu^2 x^2}(1 - 2\nu^2 x^2), \end{aligned} \tag{7}$$

whence

$$h(x) = H'(x) = 4x \sum_{\nu=1}^{\infty} \nu^2 (2\nu^2 x^2 - 3)\mathrm{e}^{-\nu^2 x^2}. \tag{8}$$

The moments of the distribution are given by

$$\begin{aligned} M_s &= \int_0^{\infty} x^s h(x) \mathrm{d}x \\ &= 2\Gamma(\frac{1}{2}s + 1)(s-1)\zeta(s), \end{aligned} \tag{9}$$

where $\zeta(s) = \sum_{m=1}^{\infty} 1/m^s$. For the first moment we have

$$M_1 = \sqrt{\pi} \tag{10}$$

and thus

$$E(\mathcal{H}_n) \sim \sqrt{2n\pi} \approx 2.50663\sqrt{n}. \tag{11}$$

The variance is

$$D^2 = M_2 - M_1 = \frac{\pi(\pi - 3)}{3}. \tag{12}$$

However, there are no analytical results on the height of random trees for $k > 1$. Using Algorithm 1 a numerical study was performed to experimentally establish a relationship between the height of a random w-tree generated by k-letter repeated words on random DFA with n states. Figs. 14-15 show the structure of w-trees generated by the repeated application of a word w of length $k = 1$ and $k = 10$ letters respectively. In Fig. 7 the same effect can be observed. Longer words w tend to generate shorter trees.

Since not every repeated word generates a random tree, only the automata which did so were considered . Fixing the value of n and k, random automata of size n and random words of length k were generated until N w-trees were found. In Fig. 16 the height distribution found in this way is depicted in a histogram. The

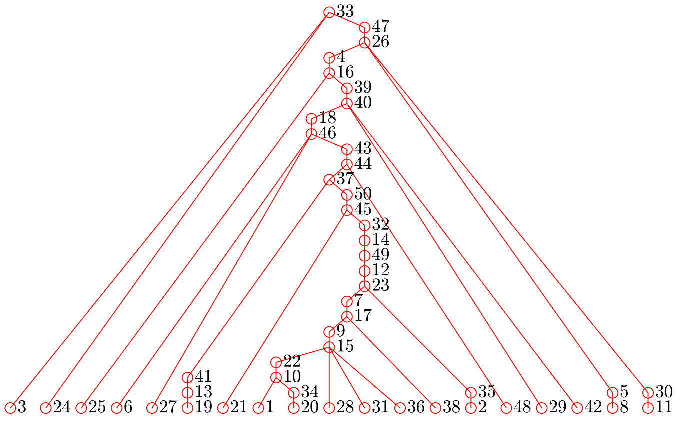


FIG. 14. w-tree generated on an automaton with 50 states by the repetition of one single letter $(k = 1)$. $H = 26$.

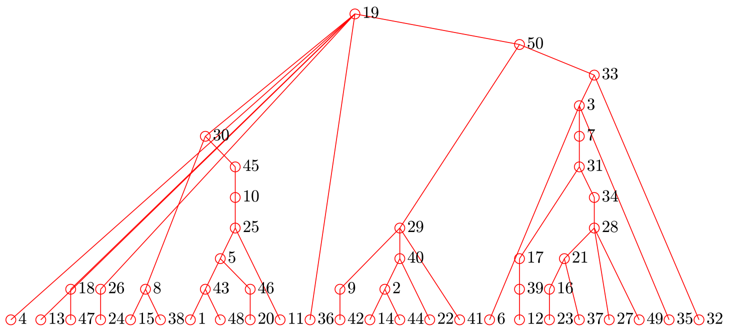


FIG. 15. w-tree generated on an automaton with 50 states by the repetition of a word w of length $k = 10$. $H = 10$.

lognormal distribution seems to fit the data well. The experiment was repeated with different values of n and k and the lognormal fit with PDF

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\log x - \mu)^2}{2\sigma^2}\right) \tag{13}$$

with parameters $\mu(n, k)$ and $\sigma(n, k)$ seems to fit the height distribution well in these cases too. Note, that the expected value of the random variable X following the lognormal distribution is given by

$$E[X] = \exp\left(\mu + \frac{\sigma^2}{2}\right) \tag{14}$$

and its median is given by

$$\tilde{X} := F^{-1}(0.5) = \mathrm{e}^{\mu}. \tag{15}$$

Using this ansatz a parameter study can be performed. First, the value of k was held fixed and the values of n were varied. Random DFA were generated until 10000 w-trees were found. The estimated value of $\exp(\mu)$ is depicted in Fig. 17. A

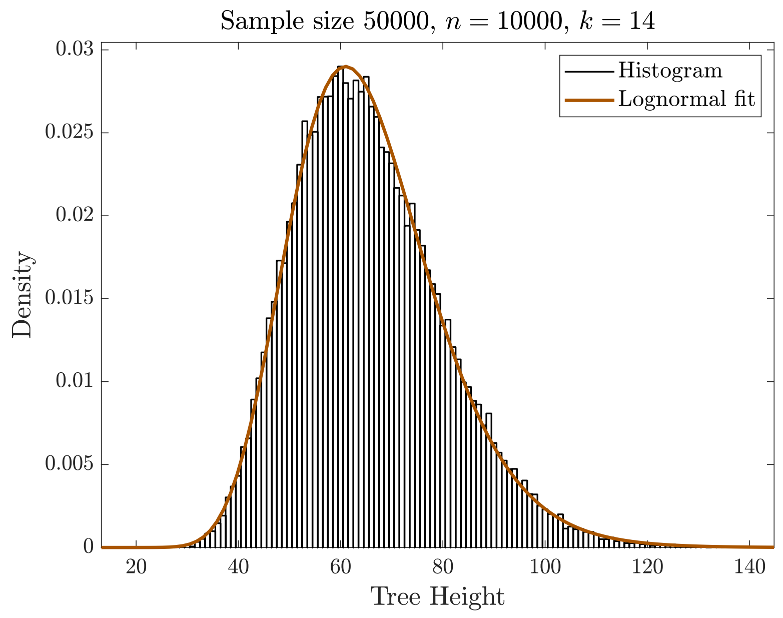


FIG. 16. Histogram of the height of w-trees generated on random DFA of size $n = 10000$ by repeatedly applying a short random word of length $k = 14$. The total number of generated random DFA was 35 610 369 among which 50 000 w-trees were found. The average length of synchronizing words were 917.95 The simulation took 20231 seconds to perform. A lognormal fit with $\mu = 4.15891 \pm 1.96 \cdot 0.0009847$ and $\sigma = 0.220921 \pm 1.96 \cdot 0.0006963$ seems to fit the distribution well.

fit on the obtained results of the form

$$\tilde{H}(n,k) = a(k)\sqrt{n} - c(k)\log(n) \tag{16}$$

seems to have a good correspondence with the simulation results. The presence of the logarithmic correction term will be justified in Sec. 6.3.

Secondly, we can fix the value of n and let k vary. Fig. 18 shows the results of an experiment carried out this way: the estimated value $\exp(\hat{\mu})$. A fit of the form

$$\tilde{H}(n,k) = \frac{A(n)}{\sqrt{k}} \tag{17}$$

shows a good correspondence with the simulation results.

Combining Eq. (16) and Eq. (17) we can write

$$\tilde{H}(n,k) = a\sqrt{\frac{n}{k}} - c\frac{\log n}{\sqrt{k}}. \tag{18}$$

Fig. 19 shows the fitted surface on the numerically obtained data of $\exp(\hat{\mu})$ where at each grid point 2000 w-trees were found. This numerical results also coincides with the conjecture of [**12**] based on analytic results. The median length of synchronizing

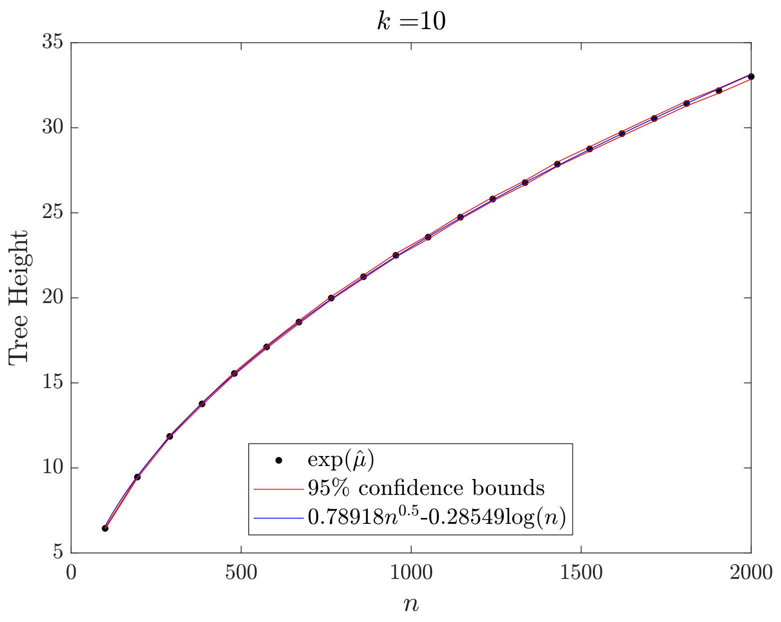


FIG. 17. The estimated value of $\exp \mu$ (black dots) is depicted against the automaton size n. Its 95% confidence bounds are shown in red. A fit of the form $a\sqrt{n} - c\ \log(n)$ is also shown in blue. The estimates are obtained using 10000 w-trees for each value of n. $R^2 = 0.9999$.

words is obtained by multiplying the results by k, i.e.,

$$|\tilde{\omega}| = a\sqrt{nk} - c \log n\sqrt{k}. \tag{19}$$

The values of the parameters based on the fit are

$$a = 2.497 \qquad (2.487, 2.506), \tag{20}$$

$$c = 0.9242 \quad (0.8789, 0.9694). \tag{21}$$

Due to Eq. (3) w.h.p. a word w of length $k \approx \lceil \log n \rceil$ exists such the the automaton is synchronized by $\omega = w|w|..w|w$. Thus, the expected length of the synchronizing words in form of w^H is approximately

$$|\tilde{\omega}| = a\sqrt{n \log n} - c \log^{\frac{3}{2}} n. \tag{22}$$

Using this ansatz to fit a curve on numerically obtained data we find the values

$$a = 2.724(2.704, 2.745), \tag{23}$$

$$c = 0.5284(0.221, 0.8359). \tag{24}$$

The 95% confidence bounds of parameter c almost include 0, thus this parameter is not really significant in the fit. Removing it the fit can be further simplified by

$$|\tilde{\omega}| = a\sqrt{n \log n} \tag{25}$$

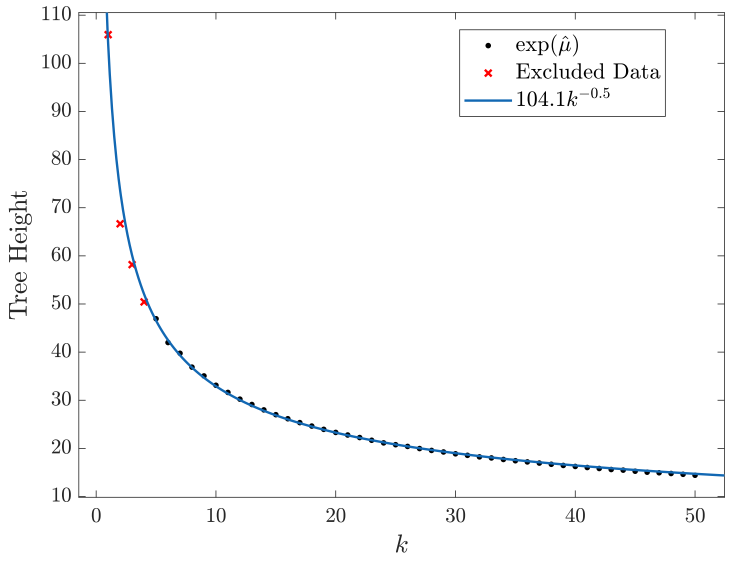


FIG. 18. The estimated value of $\exp \mu$ (black dots) is depicted against the length of the repeated word k. The automaton size is fixed at the value $n = 2000$. A fit of the form $a/\sqrt{k}$ is also shown in blue. The estimates are obtained using 10000 w-trees for each value of k. $R^2 = 0.9991$.

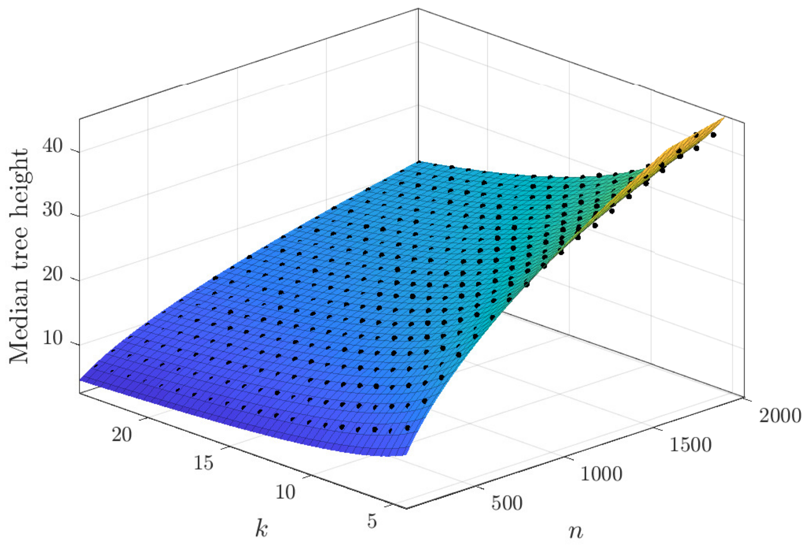


FIG. 19. The estimated value of $\exp \mu$ (black dots) is depicted against the size of the automata n and the length of the repeated word k. The fitted equation is $\tilde{H}(n, k) = 2.497\sqrt{\frac{n}{k}} - 0.9242\frac{\log n}{\sqrt{k}}$. For each grid point 2000 w-trees were found. $R^2 = 0.9995$.

n	$2^1 \ldots 2^8$	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}
N	5 000	6 000	5 000	4 800	2 400	1 200	800	400	200

Table 2. Sample sizes used in the simulation to obtain the mean SSWs using w-trees with $k = \log_2 n$.

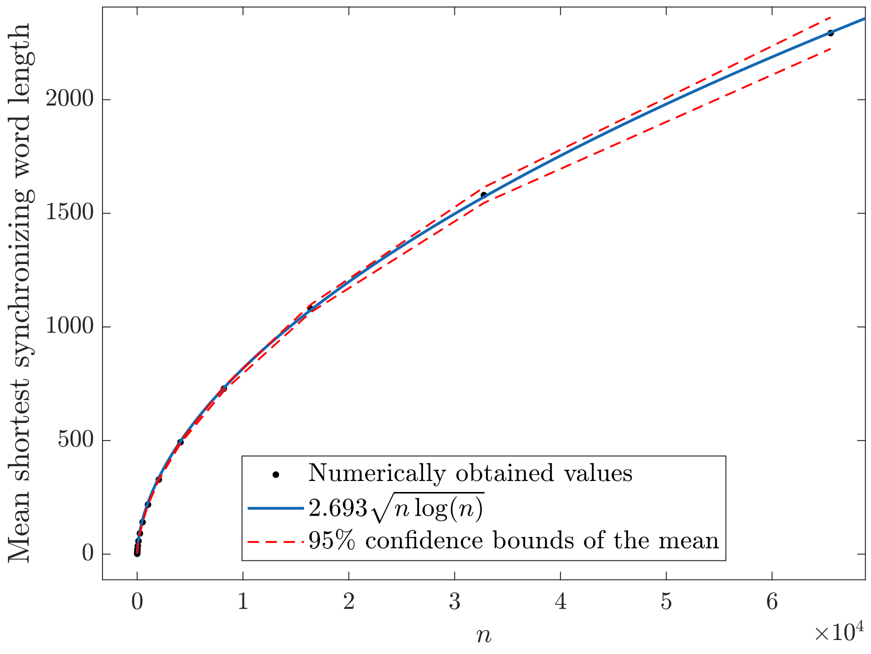


Fig. 20. Mean SSW length depicted against the number of states n using w-trees with $k = \log_2(n)$.

where

$$a = 2.693(2.68, 2.706). \tag{26}$$

In Fig. 20 the fit in Eq. (25) is depicted with automaton size given in Table 2. Since the length of the repeated word $k = |w|$ is very short even for large automata ($k = 1, 2, \ldots, 16$ in this simulation), the increment of k always influences the length of SSWs of form w^H. Some rule has to be made in order to determine a unique k to each value of n: based on Eq. (3) a choice of the form $k = \log_2 n$ is the most reasonable. For automata with size differing from powers of 2 rounding to the next integer is necessary. When the number of states is increased in a linear manner, the increment of k at a certain value of n will take place increasing the number of possible short words w by a factor of 2, which leads to a sudden decrease in the SSWs of this kind. The increment of k can be only made seamless if the automaton size is chosen to be a power of 2.

To get an idea about the standard deviation of the height of w-trees a numerical study was performed first fixing tha value of $k = 15$ and increasing the value of $n = 50, 100, \ldots, 5000$. 5000 w-trees for each grid points were found. The standard

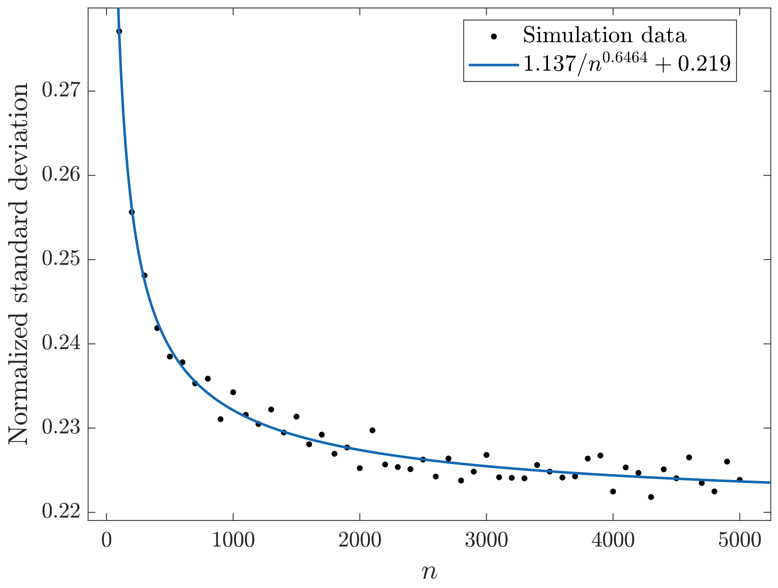


FIG. 21. The standard deviation of the height of w-trees normalized by the mean value of the tree height depicted against the automaton size n for a fixed value of $|w| = 15$. In each grid point 5 000 w-trees were used.

deviation normalized by the mean tree height is plotted in Fig. 21. The normalized standard deviation of H does not seem to tend towards zero, based on the simulation data the asymptotic value seems to be

$$\frac{\sigma(n=\infty, k=15)}{\mu(n=\infty, k=15)} = 0.219 \quad (95\% \text{ CI}: \quad 0.2173, 0.2208). \tag{27}$$

Fig. 22 the case is shown where the value of n is fixed and the length k of the repeated word w is increased where for each value of k 20 000 w-trees were found. In the depicted region the normalized standard deviation shows a slight increase. It is because the standard deviation of the tree height decreases less than the mean value of the tree height. However, for $k \to \infty$ we expect that the height of the random trees tends to 2 and so the standard deviation tends to 0. Thus, the normalized standard deviation will also tend to 0. The turning point, where the normalized standard deviation starts to decrease, however, is certainly not achieved yet for the investigated range of $k = 1 \ldots 50$.

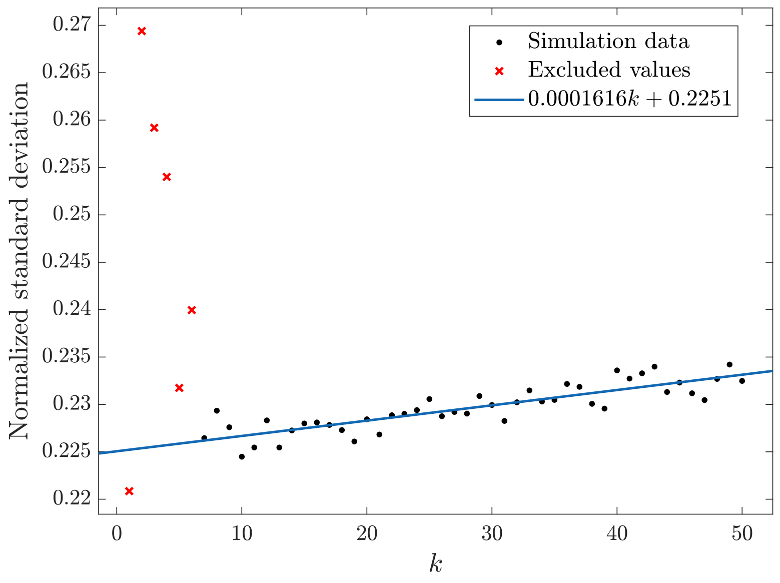


FIG. 22. The standard deviation of the height of w-trees normalized by the mean value of the tree height depicted against the length of the repeated word $|w| = k$ for fixed $n = 2000$. In each grid point 20 000 w-trees were used.

Part 2

Synchronization with random words

5. A result from the literature

In this chapter a modeling approach by Markov chains for the random automata synchronization problem will be presented. To justify the approach, an old theorem from the literature is presented first.

Higgins showed that a random automaton with an alphabet of size larger than $2n$ has, w.h.p, a synchronizing word of length $\leq 2n$ [**23**]. His proof is repeated here literally:

Definition 2. *(Full transformation semigroup) The full transformation semigroup of a finite set is a semigroup that consists of all functions from the set to itself. It is called the "full" transformation semigroup because it includes all possible functions from the set to itself, rather than just a subset of them.*

Let T_n denote the full transformation semigroup of a finite set $\{1, 2, \ldots, n\}$. Denote the range of $\alpha \in T_n$ by $\nabla\alpha$. Let X_i be the random variable with the value $\frac{|\nabla\alpha_1\alpha_2\ldots\alpha_i|}{n}$, where $\alpha_1\alpha_2\ldots\alpha_i$ are randomly chosen members of T_n (meaning that each member of T_n is equally likely to be chosen and repetitions are allowed). In [**24**] it was shown that the limiting means of X_1 and X_2 are $1-\mathrm{e}^{-1}$ and $1-\mathrm{e}^{\mathrm{e}^{-1}-1}$ respectively. The natural extension of this result is given below. Denote $\lim_{n\to\infty} E[X_i]$ by M_i, and take $M_0 = 1$.

Theorem 7. *$M_{i+1} = 1 - e^{-M_i}$ for all $i = 0, 1, 2, \ldots$, $Var[X_i] < \frac{1}{n}$ and in addition $M_i \sim 2/i$ as $i \to \infty$.*

Proof. We approach the proof by first supposing that $\alpha_1, \alpha_2, \ldots, \alpha_i$ have been randomly chosen from T_n and that $|\nabla\alpha_1\alpha_2\ldots\alpha_i| = r$. The value of $|\nabla\alpha_1\alpha_2\ldots\alpha_{i+1}|$ then has the same distribution as X, the number of non-empty boxes in a ball tossing experiment in which r balls are tossed independently at random into n boxes. From [**25**, Ch. 2,Sec. 11,exs 8 and 9] we obtain that for $1 \leq m \leq n$,

$$P(X = n-m) =: p_m(r,n) = \binom{n}{m}\sum_{\nu=0}^{n-m}(-1)^{\nu}\binom{n-m}{\nu}\frac{(n-m-\nu)^r}{n^r}. \tag{28}$$

Replacing n and m by $n-1$ and $m-1$ respectively in (28) we obtain

$$p_{m-1}(r, n-1) = \frac{m}{n}\left(\frac{n}{n-1}\right)^r p_m(r,n). \tag{29}$$

Adding equation (29) over m we get

$$1 = \frac{1}{n}\left(\frac{n}{n-1}\right)^r \sum_{m=1}^{n} m p_m(r,n),$$

or in other words

$$E[X] = n - n\left(\frac{n-1}{n}\right)^r. \tag{30}$$

All the factorial moments of $Y = n - X$ can be calculated in this manner. In particular

$$P_{m-2}(r, n-2) = \frac{m(m-1)}{n(n-1)} \left(\frac{n}{n-2}\right)^r p_m(r,n), \text{ whence}$$

$$1 = \frac{1}{n(n-1)} \left(\frac{n}{n-2}\right)^r \sum_{m=1}^{n} m(m-1) p_m(r,n)$$

$$\Rightarrow \quad E[Y(Y-1)] = n(n-1) \left(\frac{n-2}{n}\right)^r .$$

Since $\operatorname{Var}[X] = \operatorname{Var}[Y]$ we get

$$\operatorname{Var}[X] = n(n-1) \left(\frac{n-2}{n}\right)^r + n \left(\frac{n-1}{n}\right)^r - n^2 \left(\frac{n-1}{n}\right)^{2r} \tag{31}$$

or

$$\frac{\operatorname{Var}[X]}{n} = n \left(\left(1 - \frac{2}{n}\right)^r - \left(1 - \frac{1}{n}\right)^{2r} \right) + \left(1 - \frac{1}{n}\right)^r - \left(1 - \frac{2}{n}\right)^r . \tag{32}$$

Note the first and third terms of the right hand side of (32) are negative while the second is bounded above by one. Thus we have $\operatorname{Var}[X] < n$, independently of the rank r. Therefore $\operatorname{Var}[X_i] < 1/n$. The first statement of the theorem can now be proved. For $i = 0$ the result follows from (30) upon putting $r = n$. Assume inductively that the statement holds for $i = k-1, k \geq 1$. Let $\varepsilon, \delta > 0$ be given. By Chebyshev's Inequality and the fact that $\sigma_{X_k} < \frac{1}{\sqrt{n}}$ it follows that for n sufficiently large $P(|X_k - M_k| < \delta) > 1 - \varepsilon$.

The theorem is now obtained by noting that

$$E[X_{k+1} | X_k = \lambda] = 1 - \left(1 - \frac{1}{n}\right)^{n\lambda}$$

where λ almost surely lies in the arbitrarily small interval $(M_k - \delta, M_k + \delta)$ for sufficiently large n. From what has been proved and the fact that $1 - x < \mathrm{e}^{-x}$ for all $x > 0$ we see that the sequence $(M_i)_{i \geq 0}$ is a monotonically decreasing sequence which must approach 0, the unique fixed point of $f(x) = 1 - \mathrm{e}^{-x}$.

Denote $iM(i \geq 0)$ by N_i. We complete the proof by showing that $N_i \to 2$ as $i \to \infty$. Indeed, as will be proved, the sequence $(N_i)_{i \geq 0}$ is strictly increasing and so $2/i$ is an over estimate for M_i. For illustration and later reference we tabulate N_i for $0 < i < 10$. First, we verify that $N_i > 1$ for all $i > 2$. By

i	1	2	3	4	5	6	7	8	9
N_i	.632	.937	1.12	1.25	1.34	1.41	1.47	1.51	1.55

inspection $N_3 > 1$. The claim is now established by showing that if $N_{i+1} \leq 1$ then $N_i < 1$ $(i > 2)$. Since $\mathrm{e} > (1 + 1/i)^i$ it follows that $\mathrm{e}^{-1/i} < i/(i+1)$. Also $N_{i+1} \leq 1 \Rightarrow \mathrm{e}^{-M_i} \geq \frac{1}{i+1} > \mathrm{e}^{-1/i} \Rightarrow N_i < 1$, as asserted.

Next we prove that

$$N_i < 2 - \frac{1}{i} \text{ for all } i > 0. \tag{33}$$

This statement is the same as saying that $M_i < \frac{2i-1}{i^2}$. Certainly (33) is true for $i = 1$. We prove (33) in general by showing that $M_i \geq \frac{2i-1}{i^2}$, $(i > 1)$ implies that $N_{i-1} > 2 - \frac{1}{i-1}$. From our assumption we obtain

$$M_{i-1} = -\log(1 - M_i) \geq -2\log\left(1 - \frac{1}{i}\right) = 2\sum_{k=1}^{\infty} \frac{1}{ki^k}.$$

Hence $M_{i-1} > 2\left(\frac{1}{i} + \frac{1}{2i^2}\right) = \frac{2i+1}{i^2}$. Thus

$$N_{i-1} > \frac{(i-1)(2i+1)}{i^2} = 2 - \frac{i+1}{i^2} > 2 - \frac{1}{i-1},$$

as required. The next step is to prove that $(N_i)_{i\geq 0}$ is strictly increasing. By the table we see this is true for $i < 10$, so take $i \geq 10$. We wish to prove that $N_i > N_{i-1}$, for all $i \geq 10$, and since

$$\frac{N_{i-1}}{i-1} = -\log\left(1 - \frac{N_i}{i}\right)$$

this is equivalent to showing that

$$\frac{N_i}{i-1} > \frac{N_i}{i} + \frac{N_i^2}{2i^2} + \frac{N_i^3}{3i^3} + \ldots. \tag{34}$$

Since

$$\frac{1}{i-1} - \frac{1}{i} = \sum_{k=2}^{\infty} \frac{1}{i^k},$$

statement (34) can be written as

$$S = \left(N_i - \frac{N_i^2}{2}\right) + \frac{1}{i}\left(N_i - \frac{N_i^3}{3}\right) + \frac{1}{i^2}\left(N_i - \frac{N_i^4}{4}\right) + \ldots > 0. \tag{35}$$

The function $y = x - \frac{x^k}{k}$ is strictly decreasing for all $x > 1$, and since $1 < N_i < 2 - \frac{1}{i}$ we have

$$S > \left(2 - \frac{1}{i}\right) - \frac{1}{2}\left(2 - \frac{1}{i}\right)^2 + \frac{1}{i}\left(2 - \frac{2^3}{3}\right) + \frac{1}{i^2}\left(2 - \frac{2^4}{4}\right) + \ldots.$$

Hence

$$\begin{aligned} S &> \frac{1}{i} - \frac{1}{2i^2} + 2\sum_{k=1}^{\infty} \frac{1}{i^k} - \sum_{k=3}^{\infty} \frac{2^k}{ki^{k-2}} \\ &> \frac{1}{i} - \frac{1}{2i^2} + \frac{2}{i-1} - \frac{8}{3i} - \sum_{k=4}^{\infty} \frac{2^{k-2}}{k-2} \\ &= \frac{2}{i-1} - \frac{5}{3i} - \frac{1}{2i^2} - \frac{4}{i(i-2)} \\ &= \frac{2i^3 - 21i^2 + 13i - 6}{6i^2(i-1)(i-2)} > 0 \text{ as } i \geq 10. \end{aligned}$$

Therefore the sequence $(N_i)_{i\geq 0}$ is strictly increasing.

Since $(N_i)_{i\geq 0}$ is increasing and bounded above by 2, $\lim_{i\to\infty} N_i = N$ exists. We have

$$\begin{aligned}
0 < i(N_{i+1} - N_i) &= i\left((i+1)\left(1 - \mathrm{e}^{-M_i}\right) - iM_i\right) \\
&= i\left((i+1)\left(1 - \left(1 - M_i + \frac{M_i^2}{2!} - \dots\right)\right) - iM_i\right) \\
&= i\left(M_i - \frac{iM_i^2}{2} - \frac{M_i^2}{2} + \frac{(i+1)M_i^3}{3!} - \dots\right) \\
&= N_i - \frac{N_i^2}{2} - \frac{N_iM_i}{2} + N_i^2\sum_{k=3}^{\infty}\frac{(-1)^{k-1}M_i^{k-2}}{k!} + N_i\sum_{k=3}^{\infty}\frac{(-1)^{k-1}M_i^{k-1}}{k!}.
\end{aligned}$$

Now, $\lim_{i\to\infty} M_i = 0$, the series have alternating signs, and the absolute value of the terms of the series are decreasing, whereupon it follows that

$$0 \leq \lim_{i\to\infty} i(N_{i+1} - N_i) = N - \frac{1}{2}N^2.$$

If $N - \frac{1}{2}N^2 = \delta > 0$ then $N = \sum_{i=0}^{\infty}(N_{i+1} - N_i)$ would be infinite as the harmonic series diverges. Hence $N - \frac{1}{2}N^2 = 0$, and so $N = 2$ or 0. The latter alternative is impossible whence the proof is complete.

□

CONSEQUENCES. As $n \to \infty$, $\frac{T}{n} \to U$, where U has the same distribution as $\sum_{j=2}^{\infty}\frac{2V_j}{j(j-1)}$, where V_j has the standard exponential distribution: $P(V_j > x) = \mathrm{e}^{-x}$, for $x \geq 0$. Also $\lim_{n\to\infty}\frac{E[T]}{n} = 2$.

Thus the average product length before a constant map results is about $2n$. The slow down in the rate of collapse towards the end of the process is highlighted by the observation that given that $|\nabla\alpha_0| = 2$, the mean value of i such that $|\nabla\alpha_0\alpha_1\dots\alpha_i| = 1$ is n.

This result is in a good correspondence with the results of *Quattropani* and *Sau* on the asymptotic mean value of the synchronization time of random DFA [**26**].

6. Modeling with Markov chains

The excerpt from Higgins' paper ends here. The case of a two-letter automaton is without doubt different from the case when a new, independent random mapping is applied in every step, yet, simulation results do not differ significantly for both cases. In Fig. 23 the mean occupancy number, i.e., the average number of non-empty states is depicted against the number of randomly applied letters. The plot also depicts the 95% percentiles of the distribution. The size of the automaton is $n = 1000$ and $N = 10000$ automata were investigated in both cases. In the mean value, practically no difference can be seen on the plot.

As we can observe, the two results are almost identical, we can give reasonable asymptotics for automata sychronization by assuming a random mapping in every step.

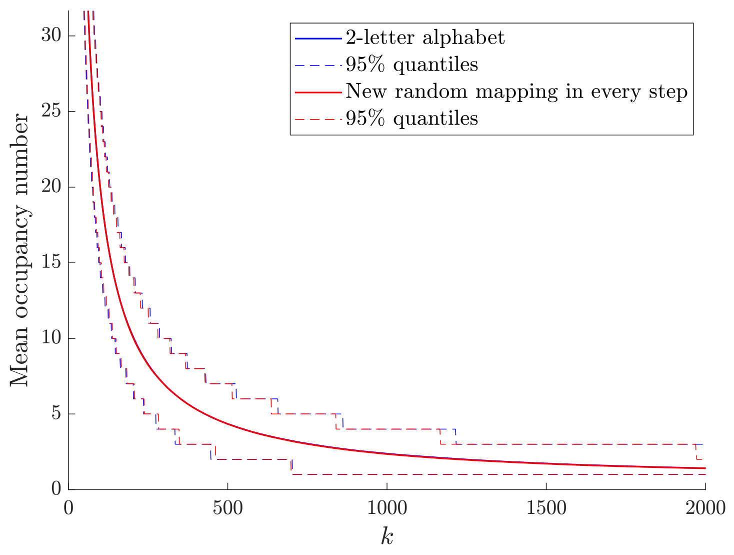


FIG. 23. Mean occupancy number depicted against the number of steps when applying a random letter from a 2-letter alphabet or a new, independent random mapping in each step. With solid blue line the mean occupancy number of the 2-letter alphabet and the 95% percentiles of the distributions are shown. With red the mean and 95% percentiles of the distribution of the occupancy number is shown when in every step a new random mapping is applied. The size of the individual automata is $n = 1000$ and $N = 10000$ automata were generated in each case.

For a better graphical representation the relative error of the simplifying assumption of new, independent random mappings compared to the application of random letters from a 2-letter random automaton is depicted in Fig. 24. We can observe that the relative error is mostly within 1%, but for small values of k it is significantly less. Thus, in the following we replace our automaton with an urn model and we construct a Markov process with n states. The transition probabilities of the Markov chain are given in Eq. (36) by the classical occupancy distribution [**23, 27**]. For further details regarding the statistical properties of this distribution we refer to [**27**]. Each state of the Markov chain corresponds to a given occupancy number. In each step the occupancy number can only get reduced or stay the same, no increase is possible, which simply means, that the already synchronized states cannot get desynchronized. This implies that the structure of the transition probability matrix is lower-diagonal. Every entry above the main diagonal is 0.

The m^{th} column and r^{th} row of the transition probability matrix is then given by

$$P_{r,m} = \frac{\binom{n}{m} m! \left\{ {r \atop m} \right\}}{n^r} = \frac{n!}{(n-m)!n^r} \sum_{i=0}^{m} (-1)^i \frac{(m-i)^r}{i!(m-i)!}. \tag{36}$$

The meaning of the formula is the following: in total, there are n^r possible ordered placements of r labeled balls into n urns. If m boxes are occupied, i.e., the occupancy number is $m \leq r$, then there are $\binom{n}{m}$ possible choices for the boxes into

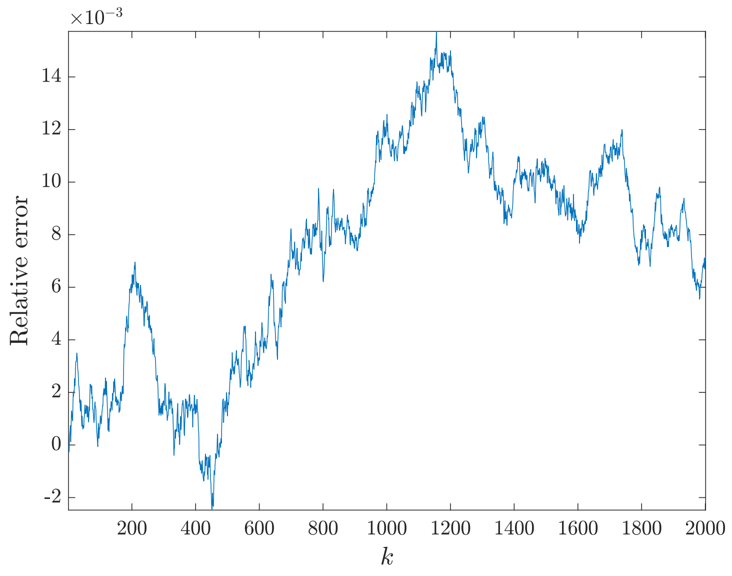


FIG. 24. Relative error of the mean occupancy number depicted against the number of steps when applying a random letter from a 2-letter alphabet or a new, independent random mapping in every step, the underlying data is also presented in Fig. 23. The size of the individual automata is $n = 1000$ and $N = 10000$ automata were generated in each case.

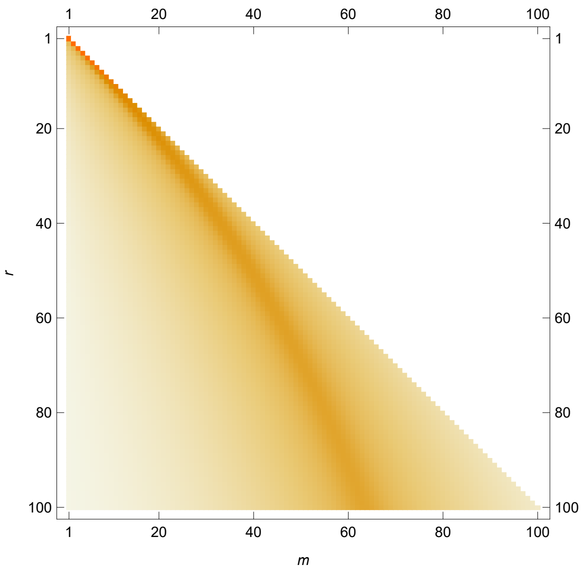


FIG. 25. The structure of the probability transition matrix of the Markov chain generated by the classical occupancy distribution for $n = 100$.

which balls are placed. By the inclusion-exclusion principle, the number of ways r balls can be placed into m boxes by taking into account their order is

$$\sum_{i=0}^{m}(-1)^i \frac{(m-i)^r}{i!(m-i)!} = m!\left\{ {r \atop m} \right\}. \tag{37}$$

Combining these observations Eq. (36) is obtained. The fraction with curly brackets denotes the Stirling numbers of the second kind. The significant probability values lie around a backbone curve, defined by the mean value of the distribution for a fixed r which is given by Eq. (30). The variance for fixed r is given by Eq. (31). In Fig. 25 the matrix is visualized for the case $n = 100$.

Evaluation of the probability values of the classical occupancy distribution is somewhat difficult numerically, since the terms to sum become huge, yet the result is a number between 0 and 1. To maintain precision, a symbolic script was used in the software Mathematica. In the range $n = 1, 2, \dots, 1000$ the matrix P was evaluated and stored to be able to perform calculations with it in MATLAB.

The Markov chain is absorbing with the only absorbing state $m = 1$ which corresponds to the synchronization of the automaton. It can be rewritten in the form

$$\mathbf{P} = \begin{pmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}, \tag{38}$$

where the rows and columns of the matrix are reorganized such, that the first row and column is placed as last row and column. $\mathbf{Q}$ is the rest of the matrix, i.e., all the transient states. $\mathbf{I}$ is the identity matrix. The fundamental matrix is given by

$$\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1} \tag{39}$$

and the expected absorption time can be calculated by

$$\tau = \mathbf{N}\mathbf{c} \quad \text{with } \mathbf{c} = [11 \dots 1]^\top. \tag{40}$$

Since the process always starts from the last state n, we are interested in the last entry of the vector of the expected absorption time τ_{n-1}. Fig. 26 shows the relative error of the estimate $2n$ compared to the numerically calculated expected absorption times up to $n = 1000$. In the beginning the estimate $2n$ performs quite badly and it is easy to understand why: for an automaton of size 1 it predicts the absorption time 2, whereas the automaton is already synchronized. For $n = 2$ it is easy to see that the expected time is 2 and not 4, which is based on the prediction. For increasing n the prediction becomes more and more accurate and converges to $2n$ as it was shown by [**23**].

Furthermore, we can get the probability of the automaton being synchronized in k steps under the application of random letters by computing

$$P(|\delta^k(Q)| = 1) = (\mathbf{P}^k)_{n,1}. \tag{41}$$

Since applying k random letters in a row offers us 2^k possibilities, the solution of

$$(\mathbf{P}^k)_{n,1} 2^k = 1 \tag{42}$$

gives us a prediction for the expected value of the SSW. Fig.27 represents the results compared to simulation results of exactly determined SSWs from [**19**] received as a courtesy from the paper's authors. The Markov chain based prediction gives a

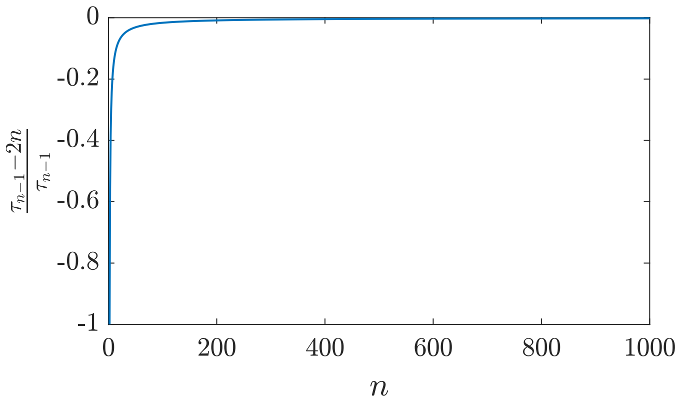


FIG. 26. Relative error of the absorption time compared to the asymptotic estimate $2n$.

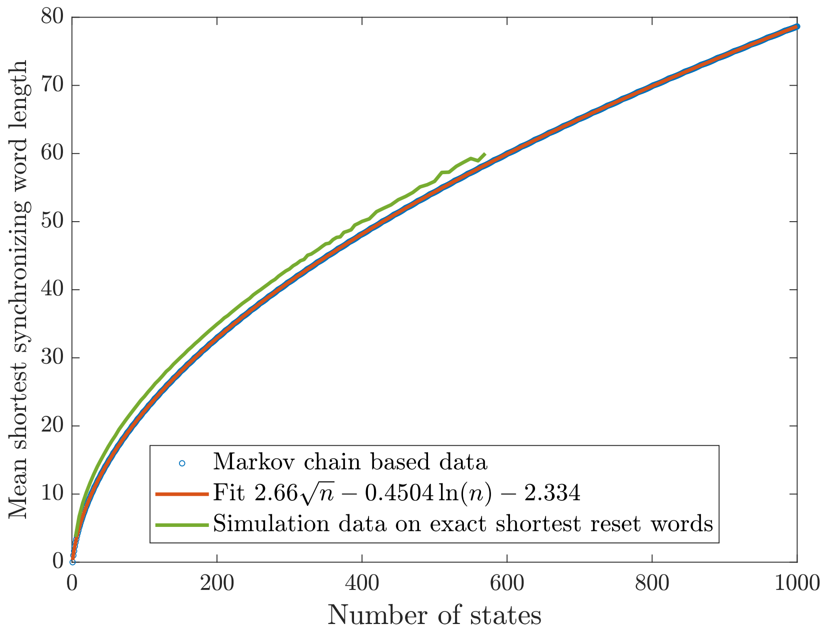


FIG. 27. Predicted length of the mean SSWs obtained by the Markov chain approach depicted against the automaton size n (blue circles). Fit on the data of the form $a\sqrt{n} - c\log n - d$ with $R^2 > 0.9999$ (orange line). The green line represents the simulation results of [**19**] showing the mean value of exactly determined SSWs.

result that is approximately 2 letters shorter on the whole interval than the ones found by [**19**].

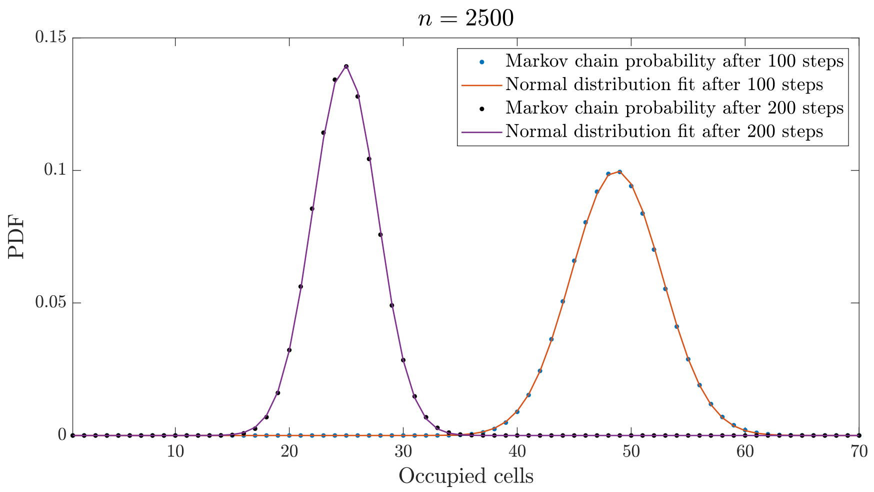


FIG. 28. Distribution of the occupancy number after the 100 and 200-fold application of a random mapping with $n = 2500$.

6.1. Estimation of the mean SSW length using the probability transition matrix. Let X_k be a random variable following the probability distribution

$$f(m; n, k) = (\mathbf{P}^k)_{n,m} \text{ for } 1 \leq m \leq n. \tag{43}$$

It is the distribution of the occupancy number after the application of k random mappings. Fig. 28 shows $f(m; 2500, 100)$ and $f(m; 2500, 200)$ respectively. The resulting distributions resemble normal ones with $\mu(n, k)$ and $\sigma(n, k)$. Thus, by fitting curves on the parameters, the SSW can be estimated by solving equation

$$f(m = 1)2^k = \frac{2^k}{\sigma(n, k)\sqrt{2\pi}} \exp\left(-\frac{(1 - \mu(n, k)^2)}{2\sigma(n, k)^2}\right) = 1 \tag{44}$$

for k. To do so, we first have to find $\mu(n, k)$ and $\sigma(n, k)$. We start with $\mu(n, k)$. Considering that the expected value of non-empty boxes after tossing μ_k balls into n containers is given by

$$\mu_{k+1} = n\left(1 - \left(1 - \frac{1}{n}\right)^{\mu_k}\right) \tag{45}$$

with $\mu_0 = n$, we have defined a non-linear difference equation which also approximates the mean value of the occupancy number due to Theorem 7 . This equation does not possess any analytic solution, therefore we perform some simplifications by rearranging it into

$$\log\left(1 - \frac{\mu_{k+1}}{n}\right) = \mu_k \log\left(1 - \frac{1}{n}\right) \tag{46}$$

and expanding it into Taylor series around $\mu = 0$:

$$-\frac{\mu_{k+1}}{n} - \frac{\mu_{k+1}^2}{2n^2} - O(\frac{1}{n^3}) = -\frac{\mu_k}{n} - \frac{\mu_k}{2n^2} - \mathcal{O}(\frac{1}{n^3}). \tag{47}$$

Now, we neglect all the terms of order less than $\mathcal{O}(\frac{1}{n^3})$, rearrange and divide by $\Delta k = 1$.

$$\mu_{k+1} - \mu_k = \frac{\mu_k - \mu_{k+1}^2}{2n} \qquad : \Delta k (= 1). \tag{48}$$

After neglecting the indices of the right hand side and writing d instead of Δ we have the differential equation

$$\frac{\Delta\mu}{\Delta k} = \frac{\mu - \mu^2}{2n} \qquad \Delta \to \mathrm{d} \tag{49}$$

$$\frac{\mathrm{d}\mu}{\mathrm{d}k} = \frac{\mu - \mu^2}{2n} \tag{50}$$

with the initial condition $\mu(0) = n$. It is a Bernoulli differential equation which has the solution

$$\mu_k = \frac{n\mathrm{e}^{\frac{k}{2n}}}{n(\mathrm{e}^{\frac{k}{2n}} - 1) + 1}. \tag{51}$$

If we are only interested in the solution for small values of k we can also neglect the term μ in 49 yielding the solution

$$\mu_k = \frac{2n}{2 + k}. \tag{52}$$

This estimate is meaningful, since we are looking for the SSWs, thus the value of k compared to n will always stay rather small. For large values of k this solution tends to 0, thus it is unrealistic, since the smallest value the occupancy number can take is 1. However, Eq. (51) fulfills this requirement. Surprisingly, Eqs. (51-52) fit the simulation data quite well. Fig. 29 depicts the comparison of the analytic estimated with simulation data. Thus for the mean of the normal distribution observed in Fig. 29 we have

$$\mu(n, k) \approx \frac{2n}{2 + k}. \tag{53}$$

It remains to find $\sigma(n, k)$, which will be done through curve fitting. In Fig. 30 standard deviation of the distribution defined in (43) is depicted against the value of k for $n = 2500$. A fit of the form

$$\sigma^2(k; n) = \frac{2n}{3k + B} \tag{54}$$

seems appropriate. Finding a plausible value for B is undertaken below. The variance of a random variable with the PDF $f(m; n, 1)$ is given by Eq. (31). One can observe that the hyperbolic shape is followed only for $k > 2$, there is no reduction in the variance from $\sigma^2(m; n, 1)$ to $\sigma^2(m; n, 2)$ since in the first step the distribution of the occupancy number is degenerate with $P(m = n | n, k = 0) = 1$. Thus in the first step, the values first spread out having a variance

$$\mathrm{Var}[X_1] = n(n-1)\left(\frac{n-2}{n}\right)^n + n\left(\frac{n-1}{n}\right)^n - n^2\left(\frac{n-1}{n}\right)^{2n}. \tag{55}$$

In the next step the variance $\mathrm{Var}[X_2]$ stays more or less constant and only for $\mathrm{Var}[X_k]$ with $k > 2$ a hyperbolic decrease might be observed. By assuming $\mathrm{Var}[X_1] \approx \mathrm{Var}[X_2]$ the value of B can be chosen such that the hyperbola passes through the

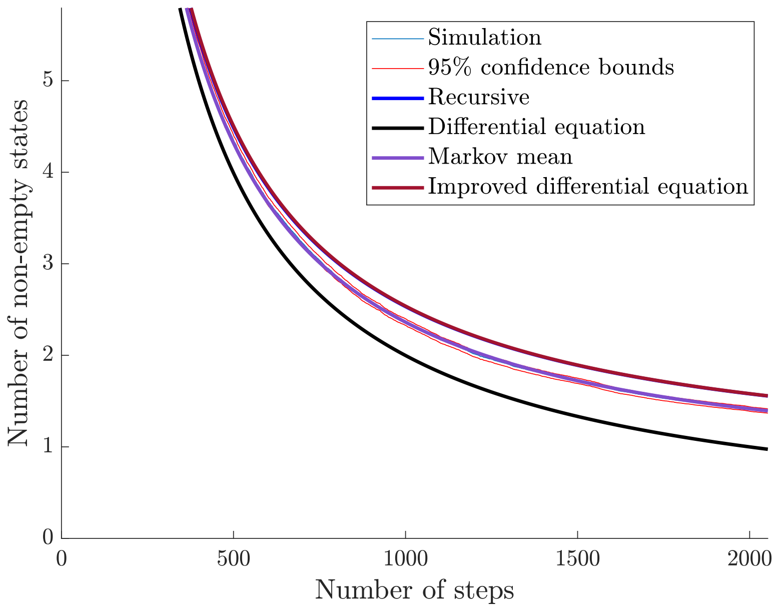


FIG. 29. Mean occupancy number depicted against the number of steps. Simulation data on random 2-letter automata is shown in light blue along with its 95% confidence bound estimate. The numerical solution of the recursive equation is shown in deep blue which for large k overlaps almost perfectly with the solution of Eq. (49) in red. The solution of the simplified differential equation given in Eq. (52) is given in black. In purple the expected value of a random variable is depicted which follows the distribution given in Eq. (43). A very good correspondence with the simulation data can be observed.

point $(k = 1, \mathrm{Var}X_1)$. Taking into account Eq. (55) and approximating the power terms by the exponential function the value

$$B := 2\left(\frac{1}{\mathrm{e}^{-1}(1-2\mathrm{e}^{-1})} - 3\right) = 14.574 \tag{56}$$

can be derived. Combining Eq. (44) with Eqs. (53-54) we obtain

$$\frac{1}{\sqrt{2\pi\frac{2n}{3k+B}}}\exp\left(-\frac{(1-\frac{2n}{k+2})^2}{2\frac{2n}{3k+B}}\right)2^k = 1 \tag{57}$$

$$-\frac{1}{2}\left(\log 4\pi + \log n - \log(3k+B)\right) - \left(1-\frac{2n}{k+2}\right)^2\frac{3k+B}{4n} + k\log 2 = 0 \tag{58}$$

$$\begin{aligned}&-2\log(4\pi)n(k+2)^2 - 2\log(n)n(k+2)^2 + 2\log(3k+B)n(k+2)^2 - (3k+B)(k+2)^2\\ &\qquad +4n(3k+B)(k+2) - 4n^2(3k+B) + 4n(k+2)^2\log(2)k = 0.\end{aligned} \tag{59}$$

Since we expect that $k = \mathcal{O}(n^{\frac{1}{2}})$ we can neglect all the terms but the two last ones for large values of n, as they are all of order less than $\mathcal{O}(n^{\frac{5}{2}})$. Thus, neglecting the

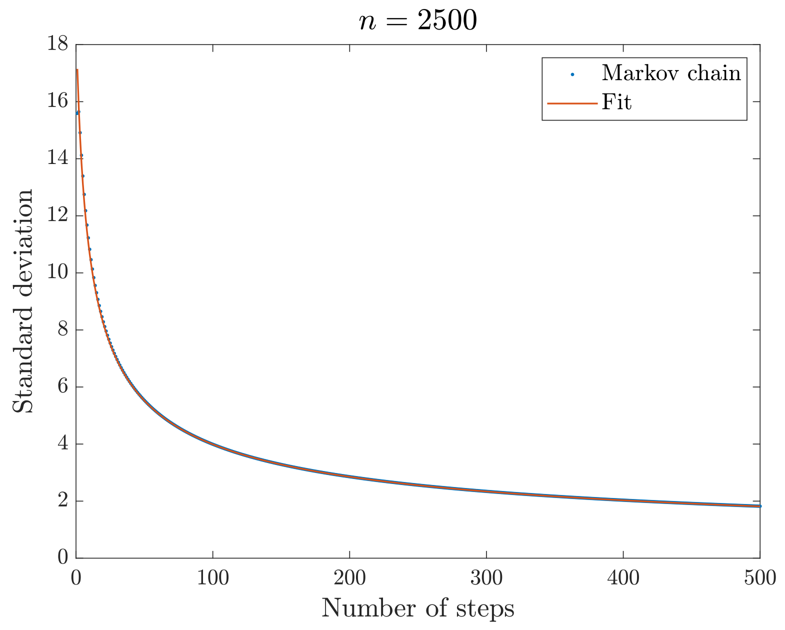


FIG. 30. Standard deviation of the occupancy number depicted against the number of steps k. Fit in the form $\sigma^2(k; n) = \frac{2n}{3k+B}$ yields a good correspondence with the numeric data. The number of states was chosen to be $n = 2500$. $B \approx 14.57$

lower order terms of the remaining expression we find

$$k \approx \sqrt{\frac{3}{\log 2} n} \approx 2.08\sqrt{n}. \tag{60}$$

The comparison of the asymptotic simplification given by Eq. (60) with the numerical solution of Eq. (57) is shown in Fig. 31. The mean length of the SSWs grows with the square root of the number of states. The obtained formula predicts a somewhat shorter length than the ones found by simulations in [**17, 19**]. It is due to the fact that the tails of the normal distribution are much heavier than the tails of the distribution defined in Eq. (43).

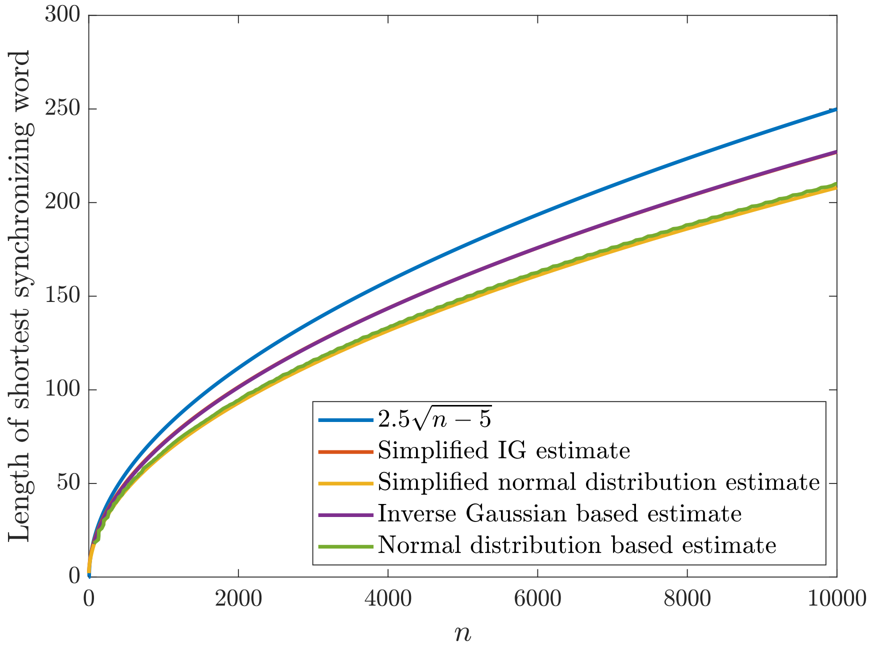


FIG. 31. Comparison of different estimates of the mean SSW of a 2-letter random DFA depicted against the number of states n. With green the numerically obtained solution of Eq. (57) is shown, whereas yellow shows the asymptotic estimate obtained by Eq. (60). With purple the numerically obtained solution of Eq. (67) is shown, which coincides with its asymptotic estimate Eq. (68) almost perfectly (shown in orange). The blue line represents the estimate given by [**17**].

6.2. Hitting time based estimation of the length of SSWs. Without using the simplified Markov chain model, we still can make an observation regarding the distribution of the hitting time under application of random random letters, i.e., when the occupancy number of a 2-letter random DFA reaches the value 1.

Based on several simulations, the absorption time (or hitting time) of the random variable corresponding to the occupancy number is approximated well by the inverse Gaussian distribution (see Fig. 32 and Fig. 33). Classically, this distribution emerges as the first passage time through a certain coordinate of a particle with drift superposed with a Brownian motion. The PDF of the distribution is given by

$$f(x;\mu,\lambda) = \sqrt{\frac{\lambda}{2\pi x^3}} \exp\left(-\frac{\lambda(x-\mu)^2}{2\mu^2 x}\right) \tag{61}$$

where parameter μ denotes the mean of the distribution and λ is the shape parameter. If X follows an IG distribution then we have

$$E[X] = \mu, \tag{62}$$

$$E\left[\frac{1}{X}\right] = \frac{1}{\mu} + \frac{1}{\lambda}. \tag{63}$$

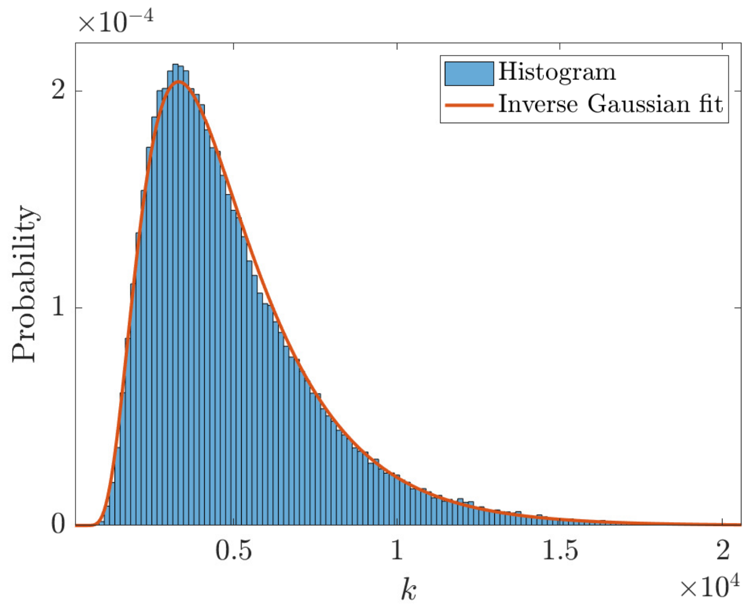


FIG. 32. Histogram and Inverse Gaussian fit of the synchronization time under the subsequent application of random letters. The results were obtained on 100 000 random DFA of size $n = 2500$. 13 DFA did not synchronize even after 50000 random letters, thus they were labeled as non-synchronizing and removed from the sample.

These facts are useful when estimating the parameters of the distribution. The results of a numerical analysis performed with automata of different size are presented in Table 3. It seems that

$$\lim_{n\to\infty} \frac{\mu}{n} = 2, \tag{64}$$

$$\lim_{n\to\infty} \frac{\lambda}{\mu} = \text{const.} \approx 3.5 - 3.6. \tag{65}$$

Thus, the estimation of the SSWs becomes possible by solving

$$\sqrt{\frac{\lambda}{2\pi x^3}} \exp\left(-\frac{\lambda(k-\mu)^2}{2\mu^2 k}\right) 2^k = 1 \tag{66}$$

for k. Taking the logarithm and rearranging the terms we obtain

$$\left(\log 2 - \frac{\lambda}{2\mu^2}\right) k^2 - \frac{3}{2} k \log k + \left(\frac{\lambda}{\mu} + \frac{1}{2}\log\frac{\lambda}{2\pi}\right) k - \frac{\lambda}{2} = 0. \tag{67}$$

Note that $\lim_{n\to\infty} \frac{\lambda}{\mu^2} = 0$. As $\lim_{n\to\infty} \lambda = \lim_{n\to\infty} \mu = \infty$ it is clear that the terms of order less than $\mathcal{O}(k^2)$ will become negligible, thus asymptotically the solution tends to

$$k = \sqrt{\frac{\lambda}{2\log 2}} \approx \sqrt{\frac{3.5}{\log 2} n} \approx 2.247\sqrt{n}, \tag{68}$$

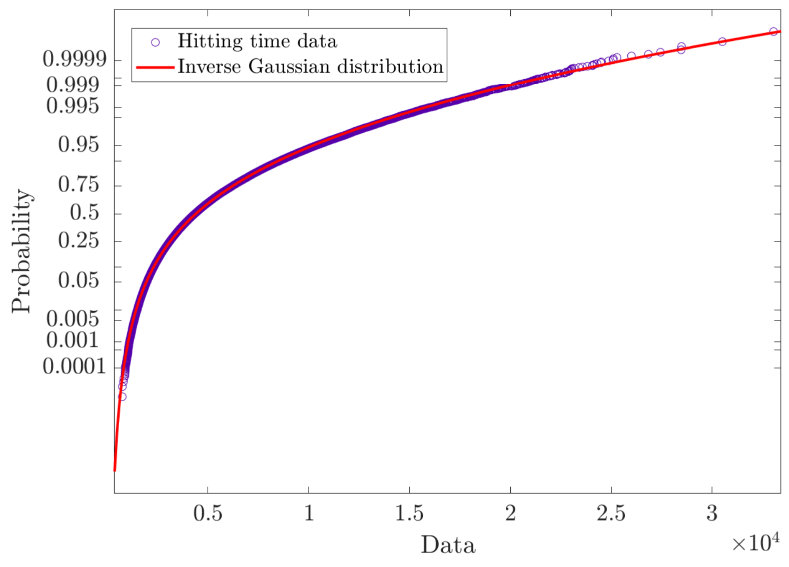


Fig. 33. Probability plot comparing the distribution of data with the inverse Gaussian distribution. The results were obtained on 100 000 random DFA of size $n = 2500$. 13 DFA did not synchronize even after 50000 random letters, thus they were labeled as non-synchronizing and removed from the sample.

n	$\hat{\mu}$	$\hat{\lambda}$	$\hat{\lambda}/\hat{\mu}$
5	11.46	10.84	0.9458
10	25.72	30.06	1.1686
25	56.00	105.6	1.8717
50	105.6	246.7	2.3365
100	204.7	568.9	2.7785
200	403.7	1261	3.1222
300	603.9	1976	3.2711
500	1004	3411	3.3956
750	1503	5116	3.4020
1000	2009	6941	3.4551
1500	2996	10485	3.4986
2000	4003	14117	3.5269
2500	5002	17723	3.5435
4000	7992	28703	3.5917
6000	11961	42825	3.5804
10000	20003	71537	3.5763

Table 3. Parameter estimates depending on the automaton size.

which slightly underestimates the numerically obtained estimate $2.5\sqrt{n-5}$ found by [**17**]. The reason why the asymptotic estimate is below the observed length of

SSWs comes from the fact that the inverse Gaussian distribution has heavier left tail than the true, but unknown distribution of the data. The estimate is plotted in Fig. 31.

6.3. Estimation of the upper bound of the mean SSW length–The "lucky" word. In this section the estimation of the upper bound of the mean SSWs is undertaken. Based on the observation that for small values of k a "typical" word follows the curve given in Eq. (52), that is,

$$m(k) \approx \frac{2n}{2+k}, \tag{69}$$

where m denotes the occupancy number. m reaches the value $\sqrt{n}$ in approximately $2\sqrt{n}$ steps. We assume that a realization m_k of the random variable X_k, being the occupancy number of the Markov process after k steps can lie in the lowest 2^{-k} quantile of the distribution of X_k even if the values of m_{k-l} with $l > k/2$ were around $E[X_{k-l}]$. In other words, a certain word performs quite mediocre with its starting letters, i.e., not reducing the occupancy number more than any other "typical" random words, but after a certain number of steps, the accumulated luck of the words leads it into the 2^{-k} quantile of the distribution after the k^{th} step, making it the SSW. Assuming that the word gets "lucky" when the occupancy number reaches the magnitude $\mathcal{O}(\sqrt{n})$, more specifically the closest integer number to $\sqrt{n}$, that we denote by $M := \lfloor\sqrt{n}\rfloor$, we can estimate the number of steps the synchronization takes in the following way: the probability of decreasing the occupancy number at least by one in a step is given by Eq. (36) as

$$P(m_{k+1} < m_k) = 1 - P(m_{k+1} = m_k) = 1 - \frac{\binom{n}{m_k} m_k! \left\{ {m_k \atop m_k} \right\}}{n^{m_k}} = 1 - \frac{(n)_{m_k}}{n^{m_k}}, \tag{70}$$

where $(x)_m$ denotes the falling factorial, i.e., $(x)_m = x(x-1)(x-2)\dots(x-m+1)$. After expansion of the falling factorial the coefficient of x^l is given by the Stirling numbers of the first kind $s(m,l)$ with $1 \le l \le m$. The following identities regarding Stirling numbers of the first kind are known

$$s(m,m) = 1, \tag{71}$$

$$s(m,m-1) = -\binom{m}{2} = -\frac{m(m-1)}{2} = -\frac{m^2}{2} + \frac{m}{2}, \tag{72}$$

$$s(m,m-2) = \frac{3m-1}{4}\binom{m}{3} = \frac{1}{8}m^4 - \frac{5}{12}m^3 + \frac{3}{8}m^2 - \frac{1}{12}m, \tag{73}$$

$$s(m,m-3) = -\binom{m}{2}\binom{m}{4} = -\frac{1}{48}m^6 + \frac{7}{48}m^5 - \frac{17}{48}m^4 + \frac{17}{48}m^3 - \frac{1}{8}m^2. \tag{74}$$

In case $m \le M$ we can rewrite Eq. (70) as

$$
\begin{aligned}
P(m_{k+1} < m_k) &= 1 - \frac{\sum_{i=0}^{m_k-1} s(m_k, m_k-i) n^{m_k-i}}{n^{m_k}} \\
&= 1 - s(m_k,m_k) - \frac{s(m_k,m_k-1)}{n} - \frac{s(m_k,m_k-2)}{n^2} - \frac{s(m_k,m_k-3)}{n^3} - \mathcal{O}(n^{-4}) \\
&= \frac{m_k(m_k-1)}{2n} - \frac{\frac{1}{8}m_k^4 - \frac{5}{12}m_k^3 + \frac{3}{8}m_k^2 - \frac{1}{12}m_k}{n^2} \\
&\quad \underbrace{+ \frac{\frac{1}{48}m_k^6 - \frac{7}{48}m_k^5 + \frac{17}{48}m_k^4 - \frac{17}{48}m_k^3 + \frac{1}{8}m_k^2}{n^3} - \mathcal{O}(n^{-4}))}_{>0 \text{ for } m_k \approx \sqrt{n} \text{ otherwise } o(1)} \\
&> \frac{m_k(m_k-1)}{2n} - \frac{\frac{1}{8}m_k^4 - \frac{5}{12}m_k^3 + \frac{3}{8}m_k^2 - \frac{1}{12}m_k}{n^2} \\
&> \frac{3m_k(m_k-1)}{8n}.
\end{aligned} \tag{75}
$$

Now, the probability that from occupancy number $m = M$ on the word w is such that it decreases the value of m_k in each step by one until it reaches 1 in $m-1$ steps, is bounded below by

$$P(m_{k+i} = m_{k+i-1} - 1 \text{ for all } i \in (1, m-1)) = \prod_{i=2}^{M} \frac{3i(i-1)}{8n} \tag{76}$$

$$= \left(\frac{3}{8}\right)^{M-1} \frac{(M!)^2}{Mn^{M-1}}. \tag{77}$$

The probability is bounded below, since it assumes decrements always only by one, however decrements by more than one are also possible.

After applying k letters w.h.p. one synchronizing word is such that the resulting occupancy number lies in the lowest 2^{-k} quantile of the distribution. Thus, using a word of k letters the automaton is synchronized w.h.p. if

$$\left(\frac{3}{8}\right)^{M-1} \frac{(M!)^2}{Mn^{M-1}} 2^k > 1. \tag{78}$$

Finding the value of k where the above inequality is tight we can give an upper bound on the mean value of the shortest syhncronizing words. Taking the logarithm we can write

$$(M-1)\log\frac{3}{8} + 2\log(M!) - \log M - (M-1)\log n + k\log 2 = 0 \tag{79}$$

Making use of Stirling's formula, i.e.,

$$M! \approx \sqrt{2\pi M}\left(\frac{M}{\mathrm{e}}\right)^M, \tag{80}$$

we can write Eq. (79) as

$$M\log\frac{3}{8} - \log\frac{3}{8} + \log(2\pi) + 2M\log M - 2M - M\log n + \log n + k\log 2 = 0. \tag{81}$$

Using that $M \approx \sqrt{n}$ we can solve the equation finally for k

$$k = \frac{2 + \log \frac{8}{3}}{\log 2}\sqrt{n} - \frac{\log n}{\log 2} - \frac{\log \frac{16\pi}{3}}{\log 2} \tag{82}$$

$$k \approx 4.3\sqrt{n} - 1.44 \log n - 4.07. \tag{83}$$

Thus, the mean value of the SSW is limited from above by $Cn^{0.5}$. This analytic results also provides a useful further hint about the form of the curve of the mean SSWs might take:

$$E[|SSW|] \approx a\sqrt{n} - c \log n - d. \tag{84}$$

In [**19**] the fitting ansatz $|SSW| = 2.5\sqrt{n-5}$ suggested by [**17**] was found to be predict to low values when extrapolated to larger n values and a fit of the form $|SSW| = an^b$ with $b = 0.515$ was suggested. However, there is no theoretical justification of this exponent, and extending the investigation to larger automata this number certainly will have to be readjusted again.

Previously, we have already used the fit given in Eq. (84), in Fig. (27) the estimate obtained of the SSW based on Markov chains were presented together with a fit of the same form, resulting in

$$E[|SSW|_{\text{Markov}}] \approx 2.66\sqrt{n} - 0.4504 \log n - 2.334 \tag{85}$$

with $R^2 > 0.9999$.

This result also gives an insight into the behavior of random DFA based on an l-letter alphabet. Eq. (78) shows, that the influence of the alphabet size comes solely through the base of 2^k. Thus, we conjecture that the mean SSWs of an n-state random DFA with an l-letter alphabet is bounded from above by

$$E[|SSW|] < \frac{2 + \log \frac{8}{3}}{\log l}\sqrt{n} - \frac{\log n}{\log l} - \frac{\log \frac{16\pi}{3}}{\log l}. \tag{86}$$

To validate the approach of the "lucky word", a fit on simulation data of explicitly found SSW was also performed. The data was obtained from *Szykuła et al.* [**19**]. The results are plotted in Figs. 34-35. It turns out that the amount of data is too small to perform a fit in the form of Eq. (84), as the 95% confidence interval of value of c includes 0, implying that the $\log n$ term is not significative. However, the author of this work believes that the logarithmic correction term will be necessary when improved algorithms in the future become capable of find exact shortest synchronizing words for even larger automata.

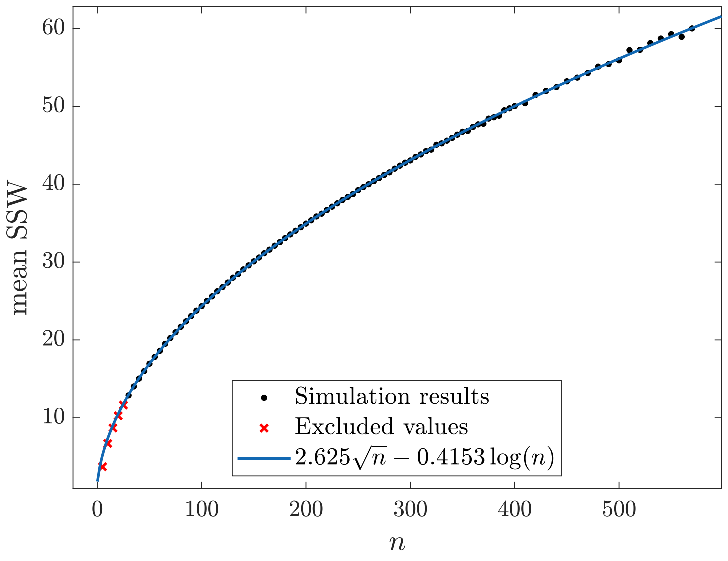


FIG. 34. $|SSW| = a\sqrt{n} - c\log n$ type of fit on the length of SSW on experimentally obtained data of [**19**]. The first few value of small automata are excluded from the fit, since they still might exhibit other effects due to the small number of states. $R^2 > 0.9999$. $RMSE = 0.1248$. The 95% confidence interval of the parameters of a and c are $(2.616, 2.635)$ and $(0.3875, 0.4431)$, respectively.

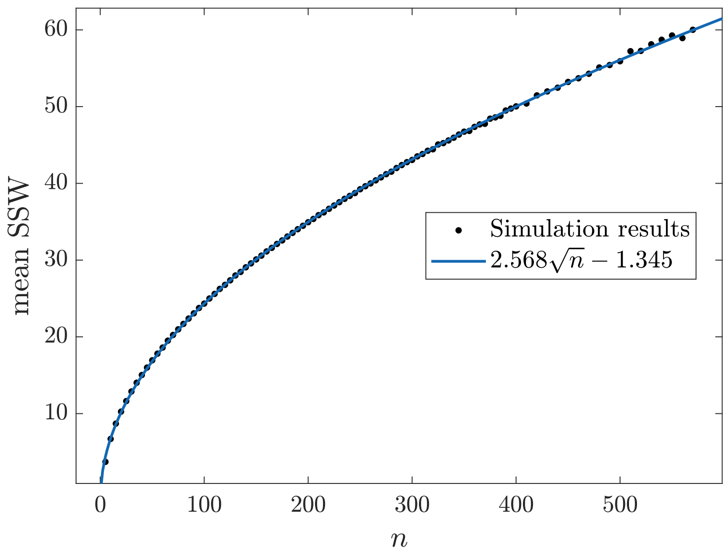


FIG. 35. $|SSW| = a\sqrt{n} - d$ type of fit on the length of SSW on experimentally obtained data of [**19**]. $R^2 > 0.9999$. $RMSE = 0.1533$. The 95% confidence interval of the parameters of a and d are $(2.562, 2.574)$ and $(1.254, 1.437)$, respectively.

7. Summary and conclusions

In this study we investigated the synchronization of random deterministic finite automata using repeated short words and random words as well. The main goal of the study was to get a better estimate for the mean SSWs of automata with n states. When aiming to synchronize by the repetition of a short word w the problem is identical to find the height of w-trees depending on the automaton size and the length of w. In Sec. 3 an algorithm was introduced to check whether the repeated application of a word w synchronizes. Using this algorithm several studies were performed. In Eq. (2) the probability that by the repetition of a random word of length k will synchronize an automaton of size n was obtained empirically. An experimental investigation on the synchronizing probability of some selected words was also performed in Sec. 4.4. The results show that when the repeated word has low entropy, thus, it is not random, but also not self-conjugated, the probability of synchronization differs from the probability random words have.

In Eq. (3) the probability that an automaton of size n is synchronizable by the repetition of any of the words w with length k was derived. This equation supports the analytic results of [**12**], i.e., w.h.p. an automaton of size n is synchronizable by the repetition of some word w of length $(1+\epsilon)\log_2 n$.

In Sec. 4.5 the height of random w-trees was investigated experimentally. The results show that the height of w-trees with $|w| = k$ generated on an automaton of size n can be well approximated by an equation of the form $\tilde{H}(n,k) = a\sqrt{\frac{n}{k}} - c\frac{\log n}{\sqrt{k}}$. Combining this with Theorem 6 implies the existence w.h.p. of a synching word of length $|\tilde{\omega}| = a\sqrt{n\log n} - c\log^{\frac{3}{2}} n$.

From Sec. 5 on, the general problem of synchronization of random DFA was investigated not limiting it to w-trees. In Sec. 5 Higgins' results on repeated random mappings on finite sets are presented which can be interpreted as the state transitions in a random DFA when in every step a new letter is applied. Sec. 6 shows by numerical means that the process described in Sec. 5 does not differ significantly from the synchronization process of a two-letter automaton. Utilizing this observation the synchronization process of a random DFA was modelled by means of Markov chains, where the transition probability matrix is defined by the use of the classical occupancy distribution. Well-known properties of Markov chains, such as the calculation of the absorption time could be made use of showing a good correspondence with experimental data.

Three estimates for the mean SSW length were given in Sec. 6 as well, two of which are based on the Markov chain model and one relies on the observation that the distribution of the synchronization time by random words of random DFA resembles much to an inverse Gaussian distribution.

All these asymptotic results suggest that the mean synchronizing word length is of $\mathcal{O}(\sqrt{n})$ with no logarithmic correction. One of these results also suggests that the mean SSW length can be represented asymptotically in a form $a\sqrt{n} - c\log n - d$. Comparison with numerical results show a good correspondence to this fit, however, the data available currently is limited due to rather small automaton size that could be investigated by exponential algorithms, thus, the significance of the terms c and d cannot by determined statistically based on the available data.

References

[1] M. S. Mahoney, "The structures of computation and the mathematical structure of nature," *The Rutherford Journal*, 2010.

[2] T. L. Booth, *Sequential machines and automata theory.* Wiley, 1967.

[3] u. unknown. (2022, Aug) Automata theory. [Online]. Available: https://en.wikipedia.org/wiki/Automata_theory

[4] J. Černý, "Poznámka k homogénnym eksperimentom s konečnými automatami," *Matematickofyzikálny Časopis Slovenskej Akadémie Vied*, vol. 14(3), pp. 208–216, 1964.

[5] J. Pin, "On two combinatorial problems arising from automata theory," in *Combinatorial Mathematics*, ser. North-Holland Mathematics Studies, C. Berge, D. Bresson, P. Camion, J. Maurras, and F. Sterboul, Eds. North-Holland, 1983, vol. 75, pp. 535–548. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304020808734327

[6] M. V. Volkov, "Synchronizing automata and the černý conjecture," in *Language and Automata Theory and Applications*, C. Martín-Vide, F. Otto, and H. Fernau, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 11–27.

[7] A. N. Trahtman, "Modifying the upper bound on the length of minimal synchronizing word," in *Fundamentals of Computation Theory*, O. Owe, M. Steffen, and J. A. Telle, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 173–180.

[8] Y. Shitov, "An improvement to a recent upper bound for synchronizing words of finite automata," 2019. [Online]. Available: https://arxiv.org/abs/1901.06542

[9] P. J. Cameron, "Dixon's theorem and random synchronization," 2011. [Online]. Available: https://arxiv.org/abs/1108.3958

[10] M. V. Berlinkov, "On the probability of being synchronizable," 2013. [Online]. Available: https://arxiv.org/abs/1304.5774

[11] C. Nicaud, "Fast synchronization of random automata," 2014. [Online]. Available: https://arxiv.org/abs/1404.6962

[12] G. Chapuy and G. Perarnau, "Short synchronizing words for random automata," in *Proceedings of the 34th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2023)*, Florence, Italy, 2023.

[13] K. Chmiel and A. Roman, "Compas - a computing package for synchronization," in *Implementation and Application of Automata*, M. Domaratzki and K. Salomaa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 79–86.

[14] A. N. Trahtman, "A package testas for checking some kinds of testability," in *Implementation and Application of Automata*, J.-M. Champarnaud and D. Maurel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 228–232.

[15] A. Roman, "Genetic algorithm for synchronization," in *Language and Automata Theory and Applications*, A. H. Dediu, A. M. Ionescu, and C. Martín-Vide, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 684–695.

[16] E. Skvortsov and E. Tipikin, "Experimental study of the shortest reset word of random automata," 2011. [Online]. Available: https://arxiv.org/abs/1105.1704

[17] A. Kisielewicz, J. Kowalski, and M. Szykuła, "A fast algorithm finding the shortest reset words," in *Lecture Notes in Computer Science.* Springer Berlin Heidelberg, 2013, pp. 182–196. [Online]. Available: https://doi.org/10.1007%2F978-3-642-38768-5_18

[18] D. Eppstein, "Reset sequences for monotonic automata," *SIAM Journal on Computing*, vol. 19, no. 3, pp. 500–510, 1990. [Online]. Available: https://doi.org/10.1137/0219033

[19] M. Szykuła and A. Zyzik, "An improved algorithm for finding the shortest synchronizing words," 2022. [Online]. Available: https://arxiv.org/abs/2207.05495

[20] J. Bondy and U. Murty, *Graph Theory with Applications*. Elsevier Science Publishing Co., Inc., 1976.

[21] A. Renyi and G. Szekeres, "On the height of trees," *Journal of the Australian Mathematical Society*, vol. 7, no. 4, pp. 497–507, 1967.

[22] L. Addario-Berry and S. Donderwinkel, "Random trees have height $o(\sqrt{n})$," 2022. [Online]. Available: https://arxiv.org/abs/2201.11773

[23] P. M. Higgins, "The range order of a product of i transformations from a finite full transformation semigroup," *Semigroup Forum*, vol. 37, no. 1, pp. 31–36, Dec 1988. [Online]. Available: https://doi.org/10.1007/BF02573120

[24] K. H. Kim and F. W. Roush, "The average rank of a product of transformations," *Semigroup Forum*, vol. 19, no. 1, pp. 79–85, Dec 1980. [Online]. Available: https://doi.org/10.1007/BF02572503

[25] W. Feller, *An Introduction to Probability Theory and Its Applications*. Wiley, January 1968, vol. 1. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20{&}path=ASIN/0471257087

[26] M. Quattropani and F. Sau, "On the meeting of random walks on random dfa," 2022. [Online]. Available: https://arxiv.org/abs/2204.02827

[27] B. O'Neill, "The classical occupancy distribution: Computation and approximation," *The American Statistician*, vol. 75, no. 4, pp. 364–375, 2021. [Online]. Available: https://doi.org/10.1080/00031305.2019.1699445

Acknowledgment

I would first like to thank my thesis advisor *Guillem Perarnau* for his guidance during the writing of this master thesis. Thanks for his hints and suggestions in our weekly meetings the research could progress in a well scheduled manner. I would also like to thank him for the thorough read through and correction of the final version of the manuscript.

I would also like to thank *Dr. Marek Szykuła* and *Adam Zyzik* for sharing with me their original data on shortest synchronizing words. Without this data I would not have been able to validate many of the ideas presented in this master thesis.

Finally, I must express my very profound gratitude to my family and friends for providing me with unfailing support and continuous encouragement throughout the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Author

Attila Genda

Master thesis statement of originality

I hereby confirm that I have written the accompanying thesis myself, without contributions from any sources other than those cited in the text and acknowledgments. This applies also to all graphics, drawings, maps and images included in the thesis.

[illegible]