One Drop of Non-Determinism in a Random Deterministic Automaton

Arnaud Carayol
Univ Gustave Eiffel, CNRS, LIGM, F-77454 Marne-la-Vallée, France

Philippe Duchon
Univ. Bordeaux, CNRS UMR 5800, LaBRI, F-33400 Talence, France

Florent Koechlin
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Cyril Nicaud
Univ Gustave Eiffel, CNRS, LIGM, F-77454 Marne-la-Vallée, France

Abstract

Every language recognized by a non-deterministic finite automaton can be recognized by a deterministic automaton, at the cost of a potential increase of the number of states, which in the worst case can go from n states to 2^n states. In this article, we investigate this classical result in a probabilistic setting where we take a deterministic automaton with n states uniformly at random and add just one random transition. These automata are almost deterministic in the sense that only one state has a non-deterministic choice when reading an input letter. In our model each state has a fixed probability to be final. We prove that for any $d \geq 1$, with non-negligible probability the minimal (deterministic) automaton of the language recognized by such an automaton has more than n^d states; as a byproduct, the expected size of its minimal automaton grows faster than any polynomial. Our result also holds when each state is final with some probability that depends on n, as long as it is not too close to 0 and 1, at distance at least $\Omega(\frac{1}{\sqrt{n}})$ to be precise, therefore allowing models with a sublinear number of final states in expectation.

2012 ACM Subject Classification Theory of computation → Regular languages; Mathematics of computing → Combinatorics; Mathematics of computing → Probability and statistics

Keywords and phrases non-deterministic automaton, powerset construction, probabilistic analysis

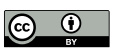
Digital Object Identifier 10.4230/LIPIcs.STACS.2023.19

Acknowledgements The authors would like to thank the reviewers for their helpful comments.

1 Introduction

A fundamental result in automata theory is that deterministic complete finite state automata recognize the same languages as non-deterministic finite state automata. This result can be established using the classical (accessible) subset construction [17, 14]: starting with a non-deterministic automaton with n states, one can build a deterministic automaton with at most 2^n states that recognizes the same language. This upper bound is tight; there are regular languages recognized by an n-state non-deterministic automaton whose minimal automaton (the smallest deterministic and complete automaton that recognizes the language) has 2^n states. The number of states of the minimal automaton of a regular language is called its *state complexity*. Figure 1 shows two n-state non-deterministic automata with somewhat similar shape, and whose languages have very different state complexities. In both automata, there is only one non-deterministic choice, when reading the letter a at the initial state.

In this article, we address the following (informal) question: if we take a random n-state deterministic automaton and add just one random transition, what can be said about the state complexity of the resulting recognized language? Does it hugely increase as for $\mathcal{L}_\ell$, or does it remain small as for $\mathcal{L}_r$?



40th International Symposium on Theoretical Aspects of Computer Science (STACS 2023).
Editors: Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté;
Article No. 19; pp. 19:1–19:14

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

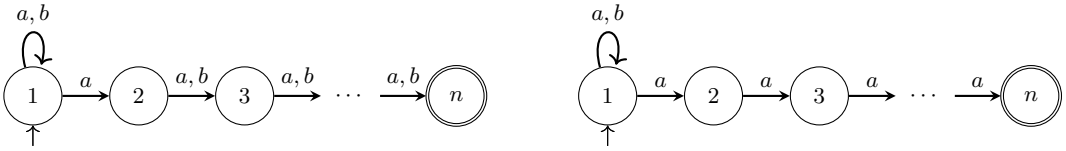


Figure 1 On the left, a non-deterministic automaton with n states recognizing the language $\mathcal{L}_\ell = \Sigma^* a \Sigma^{n-2}$. On the right, a non-deterministic automaton with n states recognizing the language $\mathcal{L}_r = \Sigma^* a^{n-1}$. The minimal automaton of $\mathcal{L}_\ell$ has 2^{n-1} states, whereas the one of $\mathcal{L}_r$ has n states.

From [3], we know that with high probability, the state complexity of the language recognized by a size-n deterministic automaton taken uniformly at random is linear. This is important as it implies that the corresponding distribution on regular languages is not degenerated: this contrasts with the case of random regular expressions where the expected state complexity of the described regular languages is constant [15] which means that the induced distribution on regular languages is concentrated on a finite number of languages.

To be more precise, our formal setting in this article is the following. Let $\Sigma = \{a, b, \ldots\}$ be a finite alphabet with $k \geq 2$ letters. For any $n \geq 1$, we consider the uniform distribution on deterministic and complete automata on Σ, with $\{1, \ldots, n\}$ as their set of states and with no final states (for now); the initial state is picked uniformly at random, and the action of the letters on the stateset are k uniform and independent random mappings. We also pick uniformly at random and independently two states p and q, and add a transition $p \xrightarrow{a} q$, if it is not already there. Finally each state is final with a given fixed probability $f \in (0, 1)$, independently. Hence in this model an almost deterministic automaton has an expected number of final states of $f\,n$. Our results still hold if we allow the probability f of being final to depend on the size n of the automaton, provided that f_n has a distance to 0 and 1 in $\Omega(\frac{1}{\sqrt{n}})$. This allows us to consider a probabilistic model in which random automata have an expected number of final states that is as low as $\Theta(\sqrt{n})$.

Our main result is that for any $d \geq 1$ there exists a constant $c_d > 0$ such that the state complexity of the language of such a random almost deterministic automaton is greater than n^d with probability at least c_d, for n sufficiently large. That is, for any polynomial P, there is a non-negligible probability that the state complexity of the language of a random automaton is greater than $P(n)$: we will say that the state complexity is *super-polynomial* with *visible probability*. As a direct consequence, the expected state complexity is super-polynomial.

It should be noted that with the same random models for deterministic automata, one cannot hope to replace visible probability in our results with a probability that converges to 1 (high probability). Indeed random automata have, with high probability, a constant fraction of states that are not accessible from the initial state; if the source of the added transition is not accessible from the initial state, the added transition does not impact the recognized language, whose state complexity is therefore at most equal to n. Thus, we make no effort in the present paper to optimize our probabilistic lower bounds. See the conclusion for a more advanced discussion on this topic.

Related work. The study of random deterministic automata can be traced back to the work of Grusho on the size of the accessible part [13]: he established that, with high probability, a constant proportion of the states are accessible from the initial state. He also shows that with high probability there is a unique terminal strongly connected component of size approximately $\nu_k n$, for some $\nu_k > \frac{1}{2}$ that only depends on the size k of the alphabet.

More structural results on the underlying graph of a random deterministic automaton were established in the work of Carayol and Nicaud [6], with a local limit law for the size of the accessible part and an application to random generation of accessible deterministic automata, and more recently in the work of Cai and Devroye [5], with, in particular, a fine grained analysis of what is happening outside the large strongly connected component. In [1], Addario-Berry, Balle and Perarnau gave a precise analysis of the diameter of a random deterministic automaton, showing in particular that it is logarithmic. We will use some of these results in this paper, namely one on the size of the largest terminal strongly connected component. We will deal with the restriction to states accessible from the initial state in the powerset construction using the result of [5] that with high probability the cycles outside the accessible part are small: for any $\varepsilon > 0$, with probability at least $1 - \varepsilon$ all the non-accessible cycles have length smaller than some constant C_ε. In particular, for any $\omega(n) \to \infty$, all the cycles outside the accessible part have length at most $\omega(n)$ with high probability.

All these results on random automata focus on the underlying graph of the transition structures, without saying much about the recognized languages, and on the average complexity of textbook algorithms on automata. Some results were established in this direction: the probability that a random accessible automaton is minimal was studied by Bassino, David and Sportiello [3], the analysis of minimization algorithms by Bassino, David and Nicaud [2, 8], etc. More recently, several papers studied the synchronization of random automata [4, 19], until the very recent work of Chapuy and Perarnau [7], establishing that most deterministic automata are synchronizing, with a word of length $O(\sqrt{n}\log n)$. We refer the interested reader to the survey of Nicaud [18] for an overview on random deterministic automata.

To our knowledge, there is no well-established random model for non-deterministic automata (e.g., for the uniform distribution they recognize almost all words with high probability). Applying the powerset construction to the mirror of a random deterministic automaton was studied by De Felice and Nicaud [10, 11], in order to analyze the average case complexity of Brzozowski's state minimization algorithm. As in the present article, they studied the determinization procedure of random automata, but for a model that is very different from ours: they consider the mirror of a uniform random deterministic automaton. In particular, with high probability, there is a linear number of states having a non-deterministic choice in their setting. Another natural model would be to use a critical Erdős-Rényi [9] digraph for each letter, which would also result in a linear number of states having a non-deterministic choice. In this article, we choose a random model with the minimum amount of non-determinism by adding just one transition to a uniform deterministic automaton, and establish that we likely have a combinatorial explosion already in this case.

2 Definitions and notations

For any $n \geq 1$, let $[n] = \{1, \ldots, n\}$. If $x, y \in \mathbb{R}$ with $x \leq y$, let $[\![x, y]\!] = [x, y] \cap \mathbb{Z}$ be the set of integers that are between x and y. Let $\mathcal{E}$ be a set equipped with a size function s from $\mathcal{E}$ to $\mathbb{Z}_{\geq 0}$, and let $\mathcal{E}_n$ denote the elements of $\mathcal{E}$ having size n. A property X on $\mathcal{E}$ (that is, a subset of $\mathcal{E}$ viewed as the set of elements for which the property holds) holds with *visible probability* if there exists some constant $c > 0$ such that, for n sufficiently large, $\mathcal{E}_n$ is non-empty and $\mathbb{P}(X) \geq c$ for the uniform distribution on $\mathcal{E}_n$. By a slight abuse of notation, if X is a random variable $\mathcal{E} \to \mathbb{Z}_{\geq 0}$ we say that for the uniform distribution on $\mathcal{E}$, X is *super-polynomial with visible probability* when for any $d \geq 1$, there exists a constant $c_d > 0$, such that for n sufficiently large, $\mathcal{E}_n \neq \emptyset$ and $\mathbb{P}(X \geq n^d) \geq c_d$ for the uniform distribution on $\mathcal{E}_n$.

Recall that if u and v are two words on an ordered alphabet Σ, u is *smaller than v for the length-lexicographic order* if $|u| < |v|$ or they have same length and $u <_{\text{lex}} v$ for the lexicographic order.

Throughout the article, the stateset of an automaton with n states will always be $[n]$, with the exception of the powerset construction recalled just below. The alphabet will always be $\Sigma = \{a, b\}$, except in the statement of our main theorem, where we allow larger alphabets as it is trivially generalized to this case. Hence, in our setting, a *deterministic (and complete) automaton* is just a tuple (n, δ, F), where $F \subseteq [n]$ is the *set of final states* and δ is the *transition function*, a mapping from $[n] \times \Sigma$ to $[n]$. We will often write $\delta_\alpha(s) = t$ or $s \xrightarrow{\alpha} t$ instead of $\delta(s, \alpha) = t$, for $s, t \in [n]$ and $\alpha \in \Sigma$, and call this an α-transition or a transition. The transition function is classically extended to sets of states by setting $\delta(X, \alpha) = \{\delta(s, \alpha) : s \in X\}$, for $X \subseteq [n]$, and to words by setting inductively $\delta(s, w) = s$ if w is the empty word ε and $\delta(s, w\alpha) = \delta(\delta(s, w), \alpha)$. We will not need to specify the *initial state* until the end of the proof; when we finally do, it will be generated uniformly at random and independently in $[n]$. Final states are only used in the last part of our proof, so to ease the presentation, we define a *deterministic (and complete) transition structure* as being an automaton with neither initial nor final states: they are given by a pair (n, δ) where n is the number of states and δ is the transition function.

An *almost deterministic automaton* $(n, \delta, F, p \xrightarrow{a} q)$ is a deterministic automaton (n, δ, F) in which we add the additional a-transition $p \xrightarrow{a} q$. Similarly, an *almost deterministic transition structure* $(n, \delta, p \xrightarrow{a} q)$ is a deterministic transition structure (n, δ) in which we add the additional a-transition $p \xrightarrow{a} q$. For any $\alpha \in \Sigma$ and any $r \in [n]$, the transition function γ of an almost deterministic automaton $(n, \delta, F, s \xrightarrow{a} t)$ (or almost deterministic transition structure) is therefore defined by $\gamma(r, \alpha) = \{\delta(r, \alpha)\}$ if $(r, \alpha) \neq (p, a)$ and $\gamma(p, a) = \{\delta(p, a), q\}$. These automata or transition structures can be deterministic, when we already have $\delta(p, a) = q$.

The classical *powerset automaton* $\mathcal{B}$ of a possibly non-deterministic automaton $\mathcal{A} = (n, \delta, F, p \xrightarrow{a} q)$, with a transition function γ, is a deterministic automaton $\mathcal{B}$ with states in $2^{[n]}$ and transition function γ extended to sets, as defined above. If we add an initial state i_0 to $\mathcal{A}$, the initial state of $\mathcal{B}$ is $\{i_0\}$ and it recognizes the same language as $\mathcal{A}$ when a state X of $\mathcal{B}$ is final if and only if at least one of its element is final in $\mathcal{A}$, i.e. $X \cap F \neq \emptyset$. We can restrict this construction to the accessible part of $\mathcal{B}$ only (from its initial state $\{i_0\}$, where i_0 is the initial state of $\mathcal{A}$) while still recognizing the same language; we call this automaton the *accessible powerset automaton* of $\mathcal{A}$.

Recall that two states r and s in a deterministic automaton $\mathcal{A}$ are *equivalent* if the languages recognized by moving the initial state to r or to s are equal. The *minimal automaton* of a regular language $\mathcal{L}$ is the deterministic complete automaton with the smallest number of states that recognizes $\mathcal{L}$. The number of states of the minimal automaton of $\mathcal{L}$ is called the *state complexity* of $\mathcal{L}$. We will use the following classical property [14]:

▶ **Proposition 1.** *If there is a set of accessible states X in a deterministic automaton $\mathcal{A}$ such that the states of X are pairwise non-equivalent, then $\mathcal{A}$ has state complexity at least $|X|$.*

The following remark allows us to focus on the case of a two-letter alphabet:

▶ Remark 2. Let $\Gamma \subseteq \Sigma$ be two non-empty alphabets. If $\mathcal{L}$ is a regular language on Σ, the state complexity of $\mathcal{L}$ is at least the state complexity of $\mathcal{L} \cap \Gamma^*$.

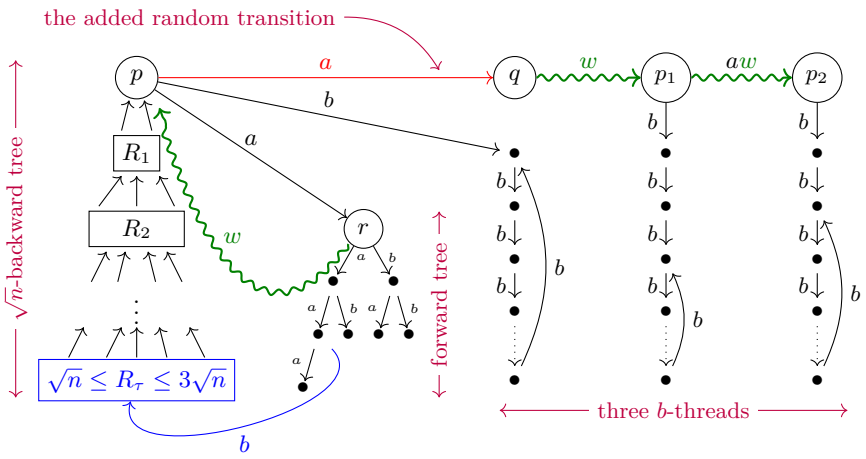


Figure 2 Illustration of the proof sketch. On the left, the backward tree from p that is detailed in Section 4.1, it has size $O(\sqrt{n})$ and contains between $\sqrt{n}$ and $3\sqrt{n}$ extremal leaves (i.e. leaves in its last level τ) to be valid. On its right, the forward tree from r, described in Section 4.2; it is a breadth-first traversal that is valid if it hits an extremal leaf of the backward tree before $O(\sqrt{n})$ states are examined. On the right the b-threads introduced in Section 4.3, obtained by reading b's from the p_i's; they are valid if they are made of previously unseen states and do not intersect.

3 Main statement and proof outline

Our main result is that the state complexity of the language recognized by a random almost deterministic automaton is super-polynomial with visible probability, when for each n, each state is final, independently, with some probability f_n that is not too close to either 0 or 1, as precised in the statement:

▶ **Theorem 3.** *Let Σ be an alphabet with at least two letters. Let f_n be a map from $\mathbb{Z}_{\geq 1}$ to $(0,1)$ such that there exists a constant $\alpha > 0$ such that $f_n \geq \frac{\alpha}{\sqrt{n}}$ and $1 - f_n \geq \frac{\alpha}{\sqrt{n}}$ for n sufficiently large. Consider an almost deterministic n-state transition structure $\mathcal{A}$ on Σ taken uniformly at random. Each state of $\mathcal{A}$ is then taken to be final with probability f_n, independently of everything else. Then, with visible probability, the language recognized by $\mathcal{A}$ has super-polynomial state complexity.*

▶ **Corollary 4.** *Under the conditions of Theorem 3, the expected state complexity of the language recognized by $\mathcal{A}$ growths faster than any polynomial in n.*

The proof of Theorem 3 consists in identifying a structure and several constraints (see Figure 2) that guarantee that when performing the accessible powerset construction and adding a random set of final states, we have sufficiently many pairwise non-equivalent states. At each step, we add a new constraint on top of those we already have, and we have to ensure that these constraints are still satisfied by sufficiently many almost deterministic transition structures. A convenient way to sketch the proof is to consider that we start with n states and no transitions, and add random transitions when needed, on the fly. More precisely, our proofs can be seen as the description of an algorithm that tries to expose the required structure by performing two types of queries on the set of still unknown transitions: either we ask what the destination of a given transition is, or we ask for all the transitions that have a given state as their destination. Thus, at any point in the algorithm, conditioned on the results of all previous queries, the destinations of all still unexposed transitions are

independent and uniform among the set of states for which we have not performed the second type of query. We use this to prove that our algorithm has a non-negligible probability of success. We also have two random states p and q and will add the transition $p \xrightarrow{a} q$ at some point. We fix $d \geq 1$ and describe the main steps of the proof below. In this high level description, recall that δ refers to the transition function of the deterministic base of the almost deterministic automaton being generated, whereas the γ refers to its transition function when adding the transition $p \xrightarrow{a} q$ during step 2. Except during this step, all transitions are added to both δ and γ simultaneously.

1. Generate $r = \delta_a(p)$, the target of the a-transition starting from p in the deterministic transition structure. With visible probability, $r \neq q$ and there is a word w of length $\Theta(\log n)$ such that $\delta_w(r) = p$, which can be found by generating $O(\sqrt{n})$ random transitions. We also assume that the b-transition starting at p is still unset. This step is the most technical, we explore backward from p and forward from r until we reach a common state.
2. Assuming such a w is found, we add the transition $p \xrightarrow{a} q$ to γ, which makes the automaton non-deterministic. We then iteratively generate the transitions starting from q and following the word $w(aw)^{d-1}$, and ask that the target of each such transition be a state that was not previously seen in the whole process. This happens with visible probability.
3. Let $p_0 = p$ and $p_i = \delta_{w(aw)^{i-1}}(q)$ for $i \in [d]$. If the two previous steps are successful, then $\gamma_{(aw)^d}(\{p\}) = \{p_0, p_1, \ldots, p_d\}$, and the outgoing b-transition of each p_i is still unset. Then, for each p_i, we iteratively generate the b-transitions $\delta_b(p_i)$, $\delta_{bb}(p_i)$, ... until we cycle after λ_i steps. This process is considered successful if we do not use an already set b-transition and if the $d+1$ cycles are pairwise disjoint. We furthermore ask that the λ_i are all in $\Theta(\sqrt{n})$. All these properties happen with visible probability.
4. At this stage, we have $\gamma_{(aw)^d}(\{p\}) = \{p_0, \ldots, p_d\}$; this set is composed of $d+1$ different states, and reading b's from each p_i eventually ends in a b-cycle of length ℓ_i. Given the λ_i's, each ℓ_i is a uniform element of $[\lambda_i]$, and they are independent. We now ask that the ℓ_i's are pairwise coprime, and that each of them is in $\Omega(\sqrt{n})$. This also happens with visible probability [20]. Our precise requirements ensure that once met, the ℓ_i are uniform and independent elements of $[\![\frac{1}{2}\sqrt{n}, \sqrt{n}]\!]$.
5. If everything worked so far, in the powerset construction applied to the almost deterministic transition structure there is a b-cycle of length $\prod_{i=0}^{d} \ell_i = \Omega(n^{\frac{d+1}{2}})$. We now randomly determine which states are final. If we consider a b-cycle alone in the automaton, of length $\Omega(\sqrt{n})$, its states are pairwise non-equivalent with visible probability as soon as the probability f_n that a state is final is not too close to either 0 or 1, which we assumed in our model. This property happens to be preserved when building the product automaton for the union of two one-letter cycles, provided their lengths are coprime. Consequently, the large b-cycles in the powerset construction are made of pairwise non-equivalent states with visible probability.
6. It just remains to guarantee that $\{p\}$ is accessible in the subset construction. We use the fact that with high probability, all cycles with length in $\Omega(\log(n))$ are accessible in a random deterministic automaton [5]. By construction the cycle around p labelled aw built at step 1 has length $\Theta(\log n)$, hence p is accessible with high probability.

The first steps of the proof sketch are depicted in Figure 2, with more details and notations that will be introduced in the next section.

4 Random almost deterministic transition structures

As indicated in the presentation of the proof in Section 3, a convenient way to see a uniform random transition structure is to start with no known transition at all, and generate them on the fly, when needed: we use the fact that the targets of the $2n$ transitions in a size-n uniform transition structure are independent uniform random elements of $[n]$.

Consider for instance that we take a random state s and iteratively follow b-transitions starting from s: we generate the path $s \xrightarrow{b} \delta(s,b) \xrightarrow{b} \delta(s,bb) \xrightarrow{b} \dots$ until we cycle back on a previously seen state. In this process, we keep picking uniformly at random and independently integers in $[n]$ until we have a collision: this is exactly the setting of the classical Birthday Problem (see for instance [12, p.114]). Straightforward computations show that the expected length ℓ_s of this b-path $\mathcal{P}_s$ is in $\Theta(\sqrt{n})$, and that it is between $\sqrt{n}$ and $2\sqrt{n}$ with visible probability.

Now suppose that we want to add the condition that the target of every a-transition outgoing from a state of $\mathcal{P}_s$ is not in $\mathcal{P}_s$. We can proceed as follows: for a given fixed path $\mathcal{P}_s$ of length ℓ_s, the Birthday Problem analysis tells us that with visible probability the outgoing a-transitions do not reach $\mathcal{P}_s$. As long as $\sqrt{n} \leq \ell_s \leq 2\sqrt{n}$, we can lower bound this probability by a constant that does not depend on ℓ_s. Moreover, a given transition structure can have only one b-path from s, so we can partition the set of size-n transition structures according to their b-path, for a given s. Hence a simple computation using the law of total probabilities (or direct counting) shows that we can combine the two "with visible probability" and that, with visible probability there is a b-path $\mathcal{P}_s$ from s of length between $\sqrt{n}$ and $2\sqrt{n}$ such that every outgoing a-transition ends outside $\mathcal{P}_s$.

We detailed this reasoning because it is the main technique we will use in the sequel to build on the previous results and add new constraints, until we exhibit a shape that ensures that applying the accessible powerset construction will produce a large (super-polynomial) number of states. Also, we will rely much on properties derived from the Birthday Problem, such as:

- If we generate $O(\sqrt{n})$ elements of $[n]$, there is no collision with visible probability, even if there is a set of forbidden states of size $O(\sqrt{n})$ which make the process fail.
- If we generate $\Omega(\sqrt{n})$ elements of $[n]$, there is a collision with visible probability, even if there is a set of forbidden states of size $O(\sqrt{n})$ which make the process fail.
- If we generate random elements of $[n]$, with visible probability we hit a fixed set of states of size $\Omega(\sqrt{n})$ before a collision occurs.

4.1 Backward tree

We first look at the shape of a typical backward tree[1] from a state p in a random transition structure $\mathcal{T} = (n, \delta)$. We define $d(x,y)$ as the smallest length of a word w such that $\delta_w(x) = y$ (and ∞ if y is not accessible from x). For a given state p, we consider the backward exploration of $\mathcal{T}$ starting from p: we iteratively build the sets of states $R_i(p) = \{x : d(x,p) = i\}$. For $\tau \geq 1$, the nodes of the *backward tree* of depth τ from p are $B_\tau(p) = \cup_{i=0}^{\tau} R_i(p)$ and the edges are the transitions $x \xrightarrow{\alpha} y$ that go from a state $x \in R_i(p)$ to a state $y \in R_{i-1}(p)$, for $i \in [\tau]$.

We keep building the backward tree until the first time τ where $|R_\tau(p)| \geq \sqrt{n}$. If such a τ exists, the tree is called the $\sqrt{n}$-backward tree. If the transition structure is taken uniformly at random, there is a visible probability that $R_\tau(p)$ exists and has size at most $3\sqrt{n}$, that $\tau = \Omega(\log n)$ and that the whole $\sqrt{n}$-backward tree contains at most $O(\sqrt{n})$ nodes.

[1] The backward tree is not a tree in the graph theoretical sense as a node at depth ℓ can have two out-going edges to two different nodes at depth $\ell - 1$.

To see that, first consider $R_1(p)$. Each state $x \neq p$ can be in $R_1(p)$, if there is a transition $x \xrightarrow{a} p$ or $x \xrightarrow{b} p$ (or both) in $\mathcal{T}$. This happens with probability $\pi_n^{(1)} = \frac{2}{n} - \frac{1}{n^2} \approx \frac{2}{n}$. The cardinality of $R_1(p)$ thus follows a binomial law of parameters $n-1$ and $\pi_n^{(1)}$. In particular, in expectation it contains around 2 states.

Assume now that we know all the $R_j(p)$ for $j \leq i$ and want to compute $R_{i+1}(p)$; we suppose that $R_i(p) \neq \emptyset$. Recall that $B_i(p) = \cup_{j=0}^{i} R_j(p)$ and let $k_i = |B_i(p)|$. By definition of d, none of the states of $B_i(p)$ can be in $R_{i+1}(p)$. On the other hand, any state x of $[n] \setminus B_i(p)$ can be in $R_{i+1}(p)$, and the condition that a state is not in $B_i(p)$ is exactly that its outgoing transitions are not in $B_{i-1}(p)$. All other target states are equally likely under this conditioning, for both transitions. Hence there are $n - k_{i-1}$ possible targets for $\delta(x,a)$ and $\delta(x,b)$: the probability that at least one of them is in $R_i(p)$ is $\pi_n^{(i)} = \frac{2|R_i(p)|}{(n-k_{i-1})} - \frac{|R_i(p)|^2}{(n-k_{i-1})^2} \approx \frac{2|R_i(p)|}{n}$ if $|R_i(p)|$ and k_{i-1} are both $o(n)$. Hence the number of elements in $R_{i+1}(p)$ follows a binomial law of parameters $n - k_i$ and $\pi_n^{(i)}$. In particular, in expectation, $R_{i+1}(p)$ is roughly twice as large as $R_i(p)$, as long as they are not too big. Since binomial laws are concentrated around their means, the presentation above can be turned into a formal proof, establishing the following result.

▶ **Lemma 5.** *Let p be a random state of a random n-state deterministic transition structure. With visible probability, the $\sqrt{n}$-backward tree from p exists, has depth $\tau \in \Theta(\log n)$, contains between $\sqrt{n}$ and $3\sqrt{n}$ extremal leaves, i.e. states in $R_\tau(p)$, and has a total number of nodes in $\Theta(\sqrt{n})$.*

In [5], Cai and Devroye also consider backward trees, with a precise analysis for fixed depth (that does not depend on n) conditionally on p being in the large strongly connected component; they use approximation by a Galton-Watson branching process. This allows them to give a more precise analysis on the existence of the circuit we are building in this paper: they prove that conditioned on the fact that p is accessible, there is such a circuit with high probability. However we cannot reuse their result directly, since we need to quantify the amount of randomness used to discover the circuit: we need unset transitions to continue our construction. It is not obvious to describe the distribution of the transitions if we condition on the existence of the circuit (in particular, there can be several such circuits).

In our setting, we have a direct access to the distribution of most unseen transitions. Indeed, if we fix the $\sqrt{n}$-backward tree T_p from p and consider a state x that is not in the tree, its outgoing transitions can end either in $[n] \setminus T_p$ or at an *extremal leaf*, a leaf of maximal depth, of T_p (otherwise x would be in T_p); and every possible state has the same probability. It is a bit more complicated for transitions outgoing from a state of T_p that are not already part of the tree, but we will not use them in our construction; except for p itself, but if we condition on having T_p, its outgoing transitions ends in uniform elements of $[n]$. So as long as we do not consider a transition outgoing from a node of T_p, except p, we can easily perform our probabilistic computations given the $\sqrt{n}$-backward tree of p being T_p. Since the $\sqrt{n}$-backward tree of p of a transition structure is unique if it exists, we can use the law of total probabilities at the end to complete the proof.

Also observe that we cannot hope for a result with high probability in our setting: the probability that p has no incoming transition is $(1 - \frac{1}{n})^{2(n-1)} \approx e^{-2}$ and is therefore visible.

4.2 Forward tree and existence of a small circuit

We fix the $\sqrt{n}$-backward tree T_p of p that satisfies the conditions of Lemma 5. Then, we generate the a-transition $p \xrightarrow{a} r$ outgoing from p: as explained in the previous section, this is a uniform random element of $[n]$. We then begin a process consisting in doing a breadth-first

traversal of the transition structure starting from $r_0 := r$. We discover the states $r_0 = \delta(r, \varepsilon)$, $r_1 = \delta(r, a)$, $r_2 = \delta(r, b)$, $r_3 = \delta(r, aa)$, $r_4 = \delta(r, ab)$, ..., where the words are taken in length-lexicographic order. We continue this process until we reach either some r_i that belongs to T_p, or an already seen r_i ($r_i = r_j$ for some $j < i$). The process is successful if we halt because we hit an extremal leaf of T_p after at most $\sqrt{n}$ steps, otherwise it fails.

Let L_p be the set of extremal leaves of T_p. As mentioned above, since we only discover new states before the last step of the process, the transition considered at time $i \geq 1$ ends in a uniform random state of $[n] \setminus (T_p \setminus L_p)$: the fact that T_p is the $\sqrt{n}$-backward tree from p prevents transitions from ending at a node of $T_p \setminus L_p$ (the case of time 0 is easily handled separately). Hence we are in a variant of the Birthday Problem: we have a target set L_p of size $\Theta(\sqrt{n})$ and we iteratively draw random numbers of $[n] \setminus (T_p \setminus L_p)$ until we hit L_p (success) or we see an element twice (failure). All the computations are classical even if we ask that the process halts before $\sqrt{n}$ steps. In particular $|[n] \setminus (T_p \setminus L_p)| = n - O(\sqrt{n})$ so we do not differ much from the standard case with parameter n. This yields:

▶ **Lemma 6.** *For the uniform distribution on size-n transition structures having T_p as $\sqrt{n}$-backward tree from p, with visible probability the breadth-first traversal starting at $r := \delta_a(p)$ hits an extremal leaf of T_p before it discovers the same state twice, and it does this in at most $\sqrt{n}$ steps.*

If the conclusions of Lemma 6 hold then there is a word w of length $\Theta(\log n)$ such that $\delta_w(r) = p$, and aw labels a circuit around p: starting from p, we read a to reach r, then we follow the path that hits an extremal leaf of T_p, discovered during the breadth-first traversal; then finally go back to p using the transitions of T_p. Observe that there can be several paths that work in the last part: it is possible that both transitions outgoing from a state at distance $i+1$ from p end in states at distance i. To uniquely determine w, we choose, in this last part, the smallest for the lexicographic order. Doing this still preserves uniqueness in the following sense: for a given transition structure, there is at most one triplet (T_p, r, F_r) such that T_p is the $\sqrt{n}$-backward tree from p, $r = \delta_a(p)$, and F_r is the forward tree from r, and all the properties of Lemma 5 and Lemma 6 are satisfied. The choice of w is then fixed by (T_p, r, F_r), and the uniqueness of the triplet, which exists when all the requirement are fulfilled, allows the use of the law of total probabilities.

Let $p \in [n]$. An n-state transition structure is *p-compatible* if its $\sqrt{n}$-backward tree from p exists and satisfies the conclusions of Lemma 5, and if the breadth-first traversal from r discovers different states that are not in T_p for all labels smaller than z, and $\delta(r, z) \in L_p$, with $|z| \leq \frac{1}{2}\log_2 n$. When the transition structure $\mathcal{T}$ is p-compatible, we define its *p-substructure* as being the incomplete automaton whose states are the states in T_p together with r and all the other states discovered during the breadth-first traversal until label z. Its transitions are the transitions of T_p, and all the transitions of the breadth-first search until label z (included). We have:

▶ **Proposition 7.** *With visible probability, an n-state transition structure taken uniformly at random is p-compatible, where p is also taken uniformly at random and independently in $[n]$. In this case, the p-substructure is uniquely determined, has $O(\sqrt{n})$ states, and contains a circuit around p labelled aw, where w is uniquely determined using the transitions of the p-structure only and we have $|w| \in \Theta(\log n)$.*

4.3 Discovering the b-threads

Fix a p-substructure X_p and consider the uniform distribution over n-state transition structures that are p-compatible with X_p. For this distribution, if we take a state $s \notin X_p$, its outgoing transitions end in an element of $[n] \setminus (T_p \setminus L_p)$, uniformly at random and independently from the others transitions. Otherwise, the state s would be in the $\sqrt{n}$-backward-tree of p.

We now add a random a-transition $p \xrightarrow{a} q$ to form a random almost deterministic transition structure that has X_p as p-substructure, by picking uniformly at random $q \in [n]$. Since $|X_p| \in O(\sqrt{n})$, with high probability $q \notin X_p$. We fix some $d \geq 1$ from now on, and read, letter by letter, the word $w(aw)^{d-1}$ starting from q, where aw labels the circuit around p in X_p given in Proposition 7. Since w has length $\Theta(\log n)$, the word $w(aw)^{d-1}$ has logarithmic length, and, using the Birthday Problem once again, with high probability we only discover new states that are not in X_p while reading the whole word. In this case, we name $p_0 = p$ and $p_i = \delta(q, w(aw)^{i-1})$ for $i \in [d]$. Observe that in the whole process, we never considered b-transitions starting from one of the p_i, with $0 \leq i \leq d$. Since $p_0 = p$ is the root of X_p, there are no constraints on its out-going transitions, thus $\delta(p_0, b)$ is a uniform random element of $[n]$. Moreover, for $i \geq 1$ each $p_i \notin X_p$ and, under our conditioning, its outgoing transitions are uniform random elements of $[n] \setminus (T_p \setminus L_p)$.

Let us define the *b-thread* of p_i as the set of all states reached from p_i using words of the form b^j. Discovering state by state such a b-thread consists in iteratively generating the outgoing b-transition of the previous state, which is done by taking a uniform element of $[n] \setminus (T_p \setminus L_p)$. Let us start with the b-thread of p_0. By the Birthday Problem again, with visible probability it cycles back after discovering between $\sqrt{n}$ and $2\sqrt{n}$ states while never discovering a state of X_p, since $|X_p| \in O(\sqrt{n})$. If this happens, we consider the b-thread from p_1. With visible probability, it also cycles back after discovering between $\sqrt{n}$ and $2\sqrt{n}$ states while never discovering a state of X_p or of the b-thread from p_0, as they both have size in $O(\sqrt{n})$. Since d is fixed, doing this for the b-thread starting at each p_i we obtain:

▶ **Lemma 8.** *Let $d \geq 1$. Let X_p be a p-substructure of size-n transition structures. For the uniform distribution on size-n transition structures that are p-compatible and that have X_p as p-substructure, if we add a random transition $p \xrightarrow{a} q$ by choosing q uniformly at random and independently in $[n]$, then with visible probability (i) the states discovered while following the path labeled by $w(aw)^{d-1}$ are all different and do not belong to X_p (ii) the b-threads starting at the p_i's, where $p_0 = p$ and $p_i = \delta(q, w(aw)^{i-1})$, have length between $\sqrt{n}$ and $2\sqrt{n}$, are pairwise disjoint and do not intersect X_p.*

4.4 Cycle lengths and accessibility

An almost deterministic transition structure that satisfies the conditions of Lemma 8 is called *(p, b)-compatible*, and we say that it has b-thread lengths $\vec{\lambda} = (\lambda_0, \ldots, \lambda_d)$ if the b-thread from each p_i has length λ_i. We also define its *(p, b)-substructure* as its p-substructure where we add the states along the path labeled by $w(aw)^{d-1}$ from q and the b-threads from each p_i.

Consider an almost deterministic transition structure $\mathcal{T}$ of given (p, b)-substructure $X_{p,b}$ with b-thread lengths $\vec{\lambda} = (\lambda_0, \ldots, \lambda_d)$ and cycle lengths $\vec{\ell} = (\ell_0, \ldots, \ell_d)$. If $\vec{\ell'} = (\ell'_0, \ldots, \ell'_d)$ is another vector where each $\ell'_i \in [\lambda_i]$, we can re-target the last b-transition of each b-thread so that the cycle lengths are now $\vec{\ell'}$. Thus, conditioned on $\vec{\lambda}$, each cycle length ℓ_i is a uniform random element of $[\lambda_i]$. Since $\sqrt{n} \leq \lambda_i \leq 2\sqrt{n}$, and since each $\ell_i \in [\![\frac{1}{2}\sqrt{n}, \sqrt{n}]\!]$, with visible probability the ℓ_i's are uniform and independent random elements of $[\![\frac{1}{2}\sqrt{n}, \sqrt{n}]\!]$.

To conclude this part, we generate the initial state i_0 uniformly at random. All our constraints so far hold with visible probability, and one of them implies the existence of a circuit of length $\Omega(\log n)$ around p. Cai and Devroye [5] established that with high probability such a cycle is accessible; the conjunction of a high-probability event with a visible event is still visible. This yields:

▶ **Theorem 9.** *Let $d \geq 1$. There exists a set $\mathfrak{T}_n$ of almost deterministic transition structures with n states and one initial state such that with visible probability for the uniform distribution over size-n almost deterministic transition structure with an initial state, the state p (source*

of the additional a-transition) is accessible from the initial state and there exists a word w of length $\Theta(\log n)$ such that $\gamma(p, w(aw)^{d-1}) = \{p_0, \dots, p_d\}$ is a set of $d+1$ states, and the b-threads starting from the p_i's have lengths λ_i in $[\![\sqrt{n}, 2\sqrt{n}\,]\!]$ and their cycle length is in $[\![\frac{1}{2}\sqrt{n}, \sqrt{n}\,]\!]$. Moreover, this set $\mathfrak{T}_n$ can be built so that for the uniform distribution on $\mathfrak{T}_n$, the cycle lengths are uniform and independent random elements of $[\![\frac{1}{2}\sqrt{n}, \sqrt{n}\,]\!]$.

If $\mathcal{T}$ is in the set $\mathfrak{T}_n$ and we read b's from $P = \{p_0, \dots, p_d\}$, we eventually reach the b-cycle of P in the accessible powerset transition structure of $\mathcal{T}$, and its length is $\mathrm{lcm}(\ell_0, \dots, \ell_d)$. As the ℓ_i's are uniform and independent random elements of $[\![\frac{1}{2}\sqrt{n}, \sqrt{n}]\!]$, their lcm is $\Omega(n^{\frac{d+1}{2}})$ with visible probability [11], yielding our first main consequence (before adding final states):

▶ **Corollary 10.** *For the uniform distribution on size-n almost deterministic transition structures, the accessible powerset transition structure has a super-polynomial number of states with visible probability.*

5 Adding final states

We are now ready to randomly select which states are final. In our model, for every n, each state is final with fixed probability f_n, which may depend on n as long as it is not too close to either 0 or 1: we require that a set of $\Theta(\sqrt{n})$ states contains both final and non-final states with visible probability. This holds under our condition that f_n and $1 - f_n$ are in $\Omega(\frac{1}{\sqrt{n}})$, as a variant of the Birthday Problem again.

Previously, we exhibited the existence with visible probability of $d+1$ occurrences of b-cycles in a random almost deterministic transition structure, yielding a large b-cycle when applying the powerset construction. We will focus on b-cycles in the sequel, as it turns out to be sufficient to prove our main result. It relies on the notion of primitive words, which we now recall.

Let Γ be a nonempty finite alphabet. If $w \in \Gamma^\ell$ is a word of length ℓ, we write $w = w_0 \cdots w_{\ell-1}$ and use the convention that all indices are taken modulo ℓ: for instance w_ℓ is the letter w_0. A nonempty word w is *primitive* if it is not a non-trivial power of another word: it cannot be written $w = z^k$ for some word z and some $k \geq 2$. If w is primitive, it is easily seen that every circular permutation of w is also primitive. See [16] for a more detailed account on primitive words.

Primitive words appear in our proof with the following observation. If $\mathcal{C} = (c_0, \dots, c_{\ell-1})$ is a b-cycle of states starting at c_0, its *associated word* is the size-ℓ word $v = v_0 \dots v_{\ell-1}$ of $\{0,1\}^\ell$ where $v_i = 1$ if and only if c_i is a final state. Recall that if we start the same cycle elsewhere, at c_i, the associated word $v' = v_i \cdots v_\ell v_0 \cdots v_{i-1}$ is primitive if and only if v is primitive: reading the associated word from any starting state preserves primitivity. A b-cycle is said to be *primitive* if one (equivalently, all) of its associated words is (are) primitive. Our study is based on the following statement.

▶ **Lemma 11.** *Let $\mathcal{A}$ be a deterministic automaton on Σ and $\alpha \in \Sigma$. If $\mathcal{C}$ is a primitive α-cycle of $\mathcal{A}$, then the states of $\mathcal{C}$ are pairwise non-equivalent: the state complexity of the language recognized by $\mathcal{A}$ is at least $|\mathcal{C}|$.*

So we reduced our problem to studying the primitivity of the b-cycles we built in Section 4, and to how it exports to the associated b-cycle in the powerset construction.

5.1 Some properties of primitive words

If $w^{(1)}$ and $w^{(2)}$ are two non-empty words of respective lengths ℓ_1 and ℓ_2 on the binary alphabet $\{0,1\}$, we denote by $w^{(1)} \odot w^{(2)}$ the word w of length $\ell = \mathrm{lcm}(\ell_1, \ell_2)$ given by $w_i = 1$ if and only if $w_i^{(1)} = 1$ or $w_i^{(2)} = 1$ (recall that the indices are taken modulo the

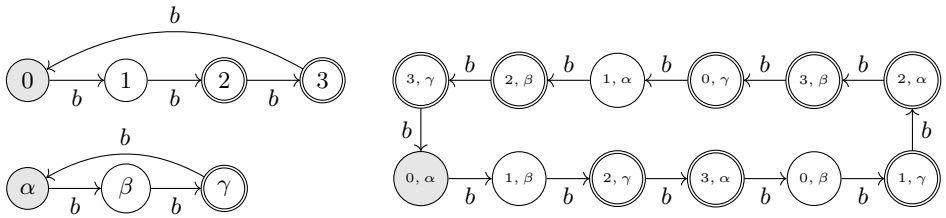


Figure 3 On the left, two primitive b-cycles (accepting states are denoted by double circles) whose associated words are 0011 (top) and 001 (bottom), starting at 0 and α, respectively. On the right, the b-cycle of $\{0, \alpha\}$ of associated word $0011 \odot 001 = 001101111011$, which is primitive by Lemma 12.

length of the word). We will see in the sequel that this operation naturally happens when extending the notion of state equivalence from each b-cycle to the corresponding b-cycle in the powerset construction.

▶ **Lemma 12.** *Let $w^{(1)}$ and $w^{(2)}$ be two primitive words on $\{0,1\}$ of lengths at least 2 that are coprime. Then, the word $w^{(1)} \odot w^{(2)}$ is primitive.*

▶ Remark 13. Lemma 12 does not hold if the lengths are not coprime. For instance, if $w^{(1)} = 011111$ and $w^{(2)} = 1011$, then $w^{(1)} \odot w^{(2)} = \underbrace{1 \dots 1}_{12 \text{ times}}$, which is not primitive.

From a probabilistic point of view, it is well known [16] that a uniform random word is primitive with very high probability. We rely on the following finer result.

▶ **Lemma 14** (De Felice, Nicaud [11])**.** *Let μ be a probability measure on $\{0,1\}^n$ such that $\mu(0^n) = \mu(1^n) = 0$ and such that two words with the same number of 0's have same probability. Then, the probability that a word is not primitive under μ is at most $\frac{2}{n}$.*

We adapt it to our needs as follows:

▶ **Corollary 15.** *Let f_n be a sequence of real numbers in $(0,1)$ such that $f_n = \Omega(\frac{1}{\sqrt{n}})$ and $1 - f_n = \Omega(\frac{1}{\sqrt{n}})$. Let ℓ be an integer greater than $\alpha\sqrt{n}$, for a fixed α, and let w be a random binary word of length ℓ whose letters are 1's with probability f_n and 0 with probability $1 - f_n$, independently. Then, w is primitive with visible probability.*

5.2 Finalizing the proof of Theorem 3

By Lemma 12, primitivity is preserved by the product $\odot$ when the lengths are coprime, so we restrict the cycle lengths built in Section 4 so that they are pairwise coprime. By Theorem 9, these lengths are uniform random elements of $[\![\frac{1}{2}\sqrt{n}, \sqrt{n}]\!]$, we therefore adapt a known result of probabilistic number theory to prove that it still happens with visible probability.

More precisely, Tóth established [20] that the probability that $d+1$ integer taken uniformly at random and independently in $[n]$ are pairwise coprime tends to some positive constant A_{d+1}, generalizing the folklore result that two independent random numbers in $[n]$ are coprime with probability that tends to $\frac{6}{\pi^2}$. This can be used to derive the following variant:

▶ **Corollary 16.** *Let $\ell_0, \ell_1, \dots, \ell_d$ be $d+1$ integers taken uniformly at random and independently in $[\![\frac{1}{2}\sqrt{n}, \sqrt{n}]\!]$. With visible probability, the ℓ_i's are pairwise coprime.*

Combining Corollary 15 and Corollary 16, we can extend Theorem 9 to also require that the b-cycles are primitive and their lengths are pairwise coprime. And this still happens with visible probability.

We can then conclude as follows: if all these requirements are met, the state p is accessible and there is a word z such that $\delta(p, z) = \{p_0, \dots, p_d\}$, the b-threads of the p_i's are pairwise disjoint and eventually form cycles of respective pairwise coprime lengths ℓ_i, and each such cycle is primitive. Moreover, all the ℓ_i are in $\Theta(\sqrt{n})$. By a direct induction on Lemma 12, this yields that the b-cycle of $\{p_0, \dots, p_d\}$ in the powerset automaton is primitive and has length $\Theta(\sqrt{n}^{d+1})$. By Lemma 11, the language recognized by this almost deterministic automaton has state complexity at least $\Theta(n^{\frac{d+1}{2}})$. This concludes the proof, as it holds for every fixed d.

6 Conclusion and discussion

Our main theorem states that the state complexity of a random almost deterministic automaton is greater than n^d with probability at least $c_d > 0$ for n sufficiently large. One can wonder how small the constant c_d is and for which sizes the lower-bound holds. As we said in the introduction, we did not try to estimate c_d nor did we try to optimize its value in this article. Since the powerset construction quickly generates very large automata which would need to be minimized, a proper experimental study does not seem feasible. However, we did generate 1000 almost deterministic transition structures with $n = 100$ states and apply the accessible powerset construction: in 78.6% of the 1000 cases the output had more than n^3 states. This would lead us to guess that even if the constant c_3 that can be derived from our proof is very small, combinatorial explosion does occur frequently in practice.

Also, as noticed above, in our setting it is certain that the property does not hold with high probability, as there is an asymptotically constant probability that the source of the added transition is not accessible. However, this probability is roughly 20.4%, not too far from what we obtained in our experiment on size-100 structures: it is very possible that if we condition the source of the added transition to be accessible, then our result holds with high probability. However, our proof techniques, based on an intensive use of the Birthday Problem cannot prove this: completely new ideas are necessary to establish such a result.

Another natural direction is to consider the case when there are *few* final states, as $\Theta(\sqrt{n})$ final states may be considered too large for a random deterministic automaton. The extreme case is to allow exactly one final state by choosing it uniformly at random. If we do so, our analysis using primitive words fails: with high probability the b-cycles we built have no final state at all, and neither has the associated b-cycle $\mathcal{C}$ in the powerset construction. However, we are confident that our techniques can be used to capture this distribution: by studying the paths ending in this final state, we should be able to find for each b-cycle $\mathcal{C}_i$ a word w_i that maps exactly one state to the final state, and such that the w_i are all different. This would be enough to establish that the states of $\mathcal{C}$ are pairwise non-equivalent and prove the conjecture. Completely formalizing and proving this idea is an ongoing work.

References

1 Louigi Addario-Berry, Borja Balle, and Guillem Perarnau Llobet. Diameter and stationary distribution of random r-out digraphs. *Electronic journal of combinatorics*, 27(P3. 28):1–41, 2020.

2 Frédérique Bassino, Julien David, and Cyril Nicaud. Average case analysis of Moore's state minimization algorithm. *Algorithmica*, 63(1-2):509–531, 2012. doi:10.1007/s00453-011-9557-7.

3 Frédérique Bassino, Julien David, and Andrea Sportiello. Asymptotic enumeration of minimal automata. In Christoph Dürr and Thomas Wilke, editors, *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th – March 3rd, 2012, Paris, France*, volume 14 of *LIPIcs*, pages 88–99. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012. doi:10.4230/LIPIcs.STACS.2012.88.

4 Mikhail V. Berlinkov. On the probability of being synchronizable. In Sathish Govindarajan and Anil Maheshwari, editors, *Algorithms and Discrete Applied Mathematics – Second International Conference, CALDAM 2016, Thiruvananthapuram, India, February 18-20, 2016, Proceedings*, volume 9602 of *Lecture Notes in Computer Science*, pages 73–84. Springer, 2016. doi:10.1007/978-3-319-29221-2_7.

5 Xing Shi Cai and Luc Devroye. The graph structure of a deterministic automaton chosen at random. *Random Structures & Algorithms*, 51(3):428–458, 2017.

6 Arnaud Carayol and Cyril Nicaud. Distribution of the number of accessible states in a random deterministic automaton. In Christoph Dürr and Thomas Wilke, editors, *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th – March 3rd, 2012, Paris, France*, volume 14 of *LIPIcs*, pages 194–205. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012. doi:10.4230/LIPIcs.STACS.2012.194.

7 Guillaume Chapuy and Guillem Perarnau. Short synchronizing words for random automata. *CoRR*, abs/2207.14108, 2022. doi:10.48550/arXiv.2207.14108.

8 Julien David. Average complexity of Moore's and Hopcroft's algorithms. *Theor. Comput. Sci.*, 417:50–65, 2012. doi:10.1016/j.tcs.2011.10.011.

9 Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

10 Sven De Felice and Cyril Nicaud. Brzozowski algorithm is generically super-polynomial for deterministic automata. In Marie-Pierre Béal and Olivier Carton, editors, *Developments in Language Theory – 17th International Conference, DLT 2013, Marne-la-Vallée, France, June 18-21, 2013. Proceedings*, volume 7907 of *Lecture Notes in Computer Science*, pages 179–190. Springer, 2013. doi:10.1007/978-3-642-38771-5_17.

11 Sven De Felice and Cyril Nicaud. Average case analysis of Brzozowski's algorithm. *Int. J. Found. Comput. Sci.*, 27(2):109–126, 2016. doi:10.1142/S0129054116400025.

12 Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

13 Aleksandr Aleksandrovich Grusho. Limit distributions of certain characteristics of random automaton graphs. *Mathematical Notes of the Academy of Sciences of the USSR*, 14(1):633–637, 1973.

14 J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

15 Florent Koechlin, Cyril Nicaud, and Pablo Rotondo. Simplifications of uniform expressions specified by systems. *Int. J. Found. Comput. Sci.*, 32(6):733–760, 2021.

16 Lothaire. *Combinatorics on Words*. Cambridge Mathematical Library. Cambridge University Press, 2 edition, 1997. doi:10.1017/CBO9780511566097.

17 Albert R Meyer and Michael J Fischer. Economy of description by automata, grammars, and formal systems. In *12th Annual Symposium on Switching and Automata Theory (SWAT 1971)*, pages 188–191. IEEE Computer Society, 1971.

18 Cyril Nicaud. Random deterministic automata. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014 – 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I*, volume 8634 of *Lecture Notes in Computer Science*, pages 5–23. Springer, 2014. doi:10.1007/978-3-662-44522-8_2.

19 Cyril Nicaud. The Černý conjecture holds with high probability. *J. Autom. Lang. Comb.*, 24(2-4):343–365, 2019. doi:10.25596/jalc-2019-343.

20 László Tóth. The probability that k positive integers are pairwise relatively prime. *Fibonacci Quart*, 40:13–18, 2002.