On the state complexity of random nondeterministic automata
GT Combinatoire et Intéractions

Arnaud Carayol, Philippe Duchon, Florent Koechlin, Cyril Nicaud

LIGM, LaBRI, LORIA, LIGM

January 23, 2023
A (nondeterministic) automaton can be described as a tuple $(A, V, E, q, F)$, where

- $A$ is a finite set (alphabet)
- $(V, E)$ is a finite directed graph with arcs labeled with letters from $A$
- $q_0 \in V$ is an initial state
- $F \subseteq V$ is a set of terminal states

The automaton is deterministic if each state has at most one outgoing arc labeled with each letter; it is complete deterministic if each state has exactly one such outgoing arc for each letter. Two automata are equivalent if they accept the same sets of words.
Finite state automata

A (nondeterministic) automaton can be described as a tuple \((A, V, E, q, F)\), where

- \(A\) is a finite set (alphabet)
- \((V, E)\) is a finite directed graph with arcs labeled with letters from \(A\)
- \(q_0 \in V\) is an initial state
- \(F \subset V\) is a set of terminal states

A word \(w \in A^*\) is **accepted** by the automaton if there exists a path from \(q_0\) to some terminal state whose arcs spell the word \(w\).
Finite state automata

A (nondeterministic) automaton can be described as a tuple $(A, V, E, q, F)$, where

- $A$ is a finite set (alphabet)
- $(V, E)$ is a finite directed graph with arcs labeled with letters from $A$
- $q_0 \in V$ is an initial state
- $F \subset V$ is a set of terminal states

A word $w \in A^*$ is accepted by the automaton if there exists a path from $q_0$ to some terminal state whose arcs spell the word $w$. The automaton is deterministic if each state has at most one outgoing arc labeled with each letter; it is complete deterministic if each state has exactly one such outgoing arc for each letter.
Finite state automata

A (nondeterministic) automaton can be described as a tuple \((A, V, E, q, F)\), where

- \(A\) is a finite set (alphabet)
- \((V, E)\) is a finite directed graph with arcs labeled with letters from \(A\)
- \(q_0 \in V\) is an initial state
- \(F \subset V\) is a set of terminal states

A word \(w \in A^*\) is **accepted** by the automaton if there exists a path from \(q_0\) to some terminal state whose arcs spell the word \(w\). The automaton is **deterministic** if each state has at most one outgoing arc labeled with each letter; it is **complete deterministic** if each state has exactly one such outgoing arc for each letter. Two automata are **equivalent** if they accept the same sets of words.
Two automata

\[ L = \{a, b\}^* a aa \]

\[ L = \{a, b\}^* a a a a \]
Some well-known facts

- The languages recognized by deterministic and (possibly) nondeterministic automata are the same (regular languages).
- There is a standard algorithm that, given a nondeterministic automaton, builds an equivalent deterministic automaton (“powerset construction”)
- The resulting automaton may have exponentially many more states, and some automata require that many states.
- There is also a well-known minimization algorithm that builds the unique (up to isomorphism) minimum-size deterministic automaton equivalent to a given automaton; the state complexity of an automaton (or regular language) is the number of states of this minimum automaton.
- **Informal question**: what is the typical state complexity of a “random” nondeterministic automaton?
(Non) deterministic automata

Structure of random automata

Sketch of proof

Conclusion

Appendix

The powerset construction

([Rabin, Scott 1959])
Starting from a non-deterministic automaton with state set \( V \), the standard deterministic construction of an equivalent deterministic automaton uses state set \( \mathcal{P}(V) \) as state set.

- Initial state is \( \{q_0\} \);
- the \( a \)-transition from some state \( S \subset V \) goes to the state of all destinations of original \( a \)-transitions from all states \( s \in S \);
- a state is terminal if it contains an original terminal state;
- one can keep only the states accessible from \( \{q_0\} \), but all nonempty sets can be accessible.
Random $k$-out graphs

($k$ is the alphabet size, $n$ is the number of states)

- Up to choice of initial and terminal states, a complete deterministic automaton is a $k$-out multigraph: each vertex has exactly $k$ outgoing edges (multiple arcs and loops allowed)
Random $k$-out graphs

($k$ is the alphabet size, $n$ is the number of states)

- Up to choice of initial and terminal states, a complete deterministic automaton is a $k$-out multigraph: each vertex has exactly $k$ outgoing edges (multiple arcs and loops allowed).
- A reasonable, simple model of random automata is to pick the destinations of the $kn$ arcs independently, uniformly at random (if required: pick the initial state uniformly, and the terminal states according to some reasonable distribution).
Random $k$-out graphs

($k$ is the alphabet size, $n$ is the number of states)

- Up to choice of initial and terminal states, a complete deterministic automaton is a $k$-out multigraph: each vertex has exactly $k$ outgoing edges (multiple arcs and loops allowed).
- A reasonable, simple model of random automata is to pick the destinations of the $kn$ arcs independently, uniformly at random (if required: pick the initial state uniformly, and the terminal states according to some reasonable distribution).
- The local and global structure of such random directed graphs is well understood.
Some known results on random DFAs

- With asymptotic probability 1, the graph has a single giant strongly connected component, of size $\nu_k n$ ($0 < \nu_k < 1$)
Some known results on random DFAs

- With asymptotic probability 1, the graph has a single giant strongly connected component, of size $\nu_k n$ ($0 < \nu_k < 1$)
- Typical distances inside this component are of order $\log(n)$
Some known results on random DFAs

- With asymptotic probability 1, the graph has a single giant strongly connected component, of size $\nu_k n$ ($0 < \nu_k < 1$)
- Typical distances inside this component are of order $\log(n)$
- Cycles outside this component are short and there are few of them [Cai-Devroye 2017]
Some known results on random DFAs

- With asymptotic probability 1, the graph has a single giant strongly connected component, of size $\nu_k n$ ($0 < \nu_k < 1$)
- Typical distances inside this component are of order $\log(n)$
- Cycles outside this component are short and there are few of them [Cai-Devroye 2017]
- Very rough “explanation” : the indegree of a vertex is close to a Poisson variable with expectation $k$, and
  - locally, the breadth-first exploration of vertices accessible from $u$ looks like a regular tree of degree $k$;
  - locally, the breadth-first “backwards” exploration of vertices from which $u$ is accessible looks like a Galton-Watson tree with progeny distributed as Poisson variables
  - the constant $\nu_k$ is the survival probability of such a process
The special case $k = 1$

- Each state has a transition to a uniform state.
The special case $k = 1$

- Each state has a transition to a uniform state.
- Starting from a fixed state and following transitions will ultimately lead to completing a cycle.
The special case $k = 1$

- Each state has a transition to a uniform state.
- Starting from a fixed state and following transitions will ultimately lead to completing a cycle.
- Expected number of transitions for this is $\Theta(n^{1/2})$ ("birthday paradox")
The special case $k = 1$

- Each state has a transition to a uniform state.
- Starting from a fixed state and following transitions will ultimately lead to completing a cycle.
- Expected number of transitions for this is $\Theta(n^{1/2})$ (“birthday paradox”)
- Conditioned on this number being $m$, the cycle length is uniform in $[[1, n]]$
The special case $k = 1$

- Each state has a transition to a uniform state.
- Starting from a fixed state and following transitions will ultimately lead to completing a cycle.
- Expected number of transitions for this is $\Theta(n^{1/2})$ ("birthday paradox")
- Conditioned on this number being $m$, the cycle length is uniform in $[1, n]$.
- This is also what one observes for larger $k$ by looking at only transitions for one letter.
A random model for nondeterministic automata

- There is no obviously natural model for random nondeterministic automata.
There is no obviously natural model for random nondeterministic automata.

Simple models, such as having an $a$-transition from $u$ to $v$ with probability $p > 0$ (independently, for each $u$ and $v$) will very likely accept all words (depending on model for terminal states).
A random model for nondeterministic automata

- There is no obviously natural model for random nondeterministic automata.
- Simple models, such as having a transition from \( u \) to \( v \) with probability \( p > 0 \) (independently, for each \( u \) and \( v \)) will very likely accept all words (depending on model for terminal states).
- **Our model**: “minimally nondeterministic”
  - start from a random deterministic (\( k \)-out) transition structure
  - add **one** additional transition: pick two states \( p, q \) at random and add an \( a \)-transition from \( p \) to \( q \)
  - add choice of (uniform) initial state and terminal states; for technical reasons, we consider models where each state is terminal with equal probability \( p \) (\( 0 < p < 1 \)); actually, \( p = \omega(n^{-1/2}) \) and \( 1 - p = \omega(n^{-1/2}) \) suffices.
There is no obviously natural model for random nondeterministic automata.

Simple models, such as having a transition from \( u \) to \( v \) with probability \( p > 0 \) (independently, for each \( u \) and \( v \)) will very likely accept all words (depending on model for terminal states).

**Our model**: “minimally nondeterministic”

- start from a random deterministic (\( k \)-out) transition structure
- add one additional transition: pick two states \( p, q \) at random and add an \( a \)-transition from \( p \) to \( q \)
- add choice of (uniform) initial state and terminal states; for technical reasons, we consider models where each state is terminal with equal probability \( p \) (\( 0 < p < 1 \)); actually, 
  \[ p = \omega(n^{-1/2}) \text{ and } 1 - p = \omega(n^{-1/2}) \] suffices.

Call this model, random almost-deterministic automata (with alphabet size \( k \) and \( n \) states).
## Main result

**Theorem (CDKN ’23)**

- For fixed alphabet size $k \geq 2$, the **expected state complexity** of random almost-deterministic automata with $n$ states grows faster than any polynomial in $n$.

- More precisely, for any finite $d > 0$, there exist positive constants $A$ and $\varepsilon$ such that there is (asymptotic) probability at least $\varepsilon$ that a random almost-deterministic automaton with $n$ states has state complexity at least $An^d$. 
High-level sketch of proof

- We “look at” transitions as needed, sequentially (so the remaining transitions remain independent of all those already observed).
High-level sketch of proof

- We “look at” transitions as needed, sequentially (so the remaining transitions remain independent of all those already observed).
- **Step 1**: find a short word $w$ such that reading $w$ from $q$ leads back to state $p$. 
High-level sketch of proof

- We “look at” transitions as needed, sequentially (so the remaining transitions remain independent of all those already observed).
- **Step 1**: find a short word $w$ such that reading $w$ from $q$ leads back to state $p$.
- **Step 2**: use $w$ to find a “large” state $\{p_0, \ldots, p_{2d-1}\}$ in the powerset construction.
High-level sketch of proof

- We “look at” transitions as needed, sequentially (so the remaining transitions remain independent of all those already observed).
- **Step 1**: find a short word \( w \) such that reading \( w \) from \( q \) leads back to state \( p \).
- **Step 2**: use \( w \) to find a “large” state \( \{p_0, \ldots, p_{2d-1}\} \) in the powerset construction.
- **Steps 3-4**: hope that \( b \)-transitions from the \( p_i \) lead to independent cycles of length at \( \Theta(n^{1/2}) \), with pairwise coprime lengths; this leads to \( \Omega(n^d) \) states in the powerset construction.
High-level sketch of proof

- We “look at” transitions as needed, sequentially (so the remaining transitions remain independent of all those already observed).

- **Step 1**: find a short word $w$ such that reading $w$ from $q$ leads back to state $p$.

- **Step 2**: use $w$ to find a “large” state $\{p_0, \ldots, p_{2^d-1}\}$ in the powerset construction.

- **Steps 3-4**: hope that $b$-transitions from the $p_i$ lead to independent cycles of length at $\Theta(n^{1/2})$, with pairwise coprime lengths; this leads to $\Omega(n^d)$ states in the powerset construction.

- **Step 5**: hope that the distribution of terminal states in cycles ensures these states remain after minimization.
High-level sketch of proof

- We “look at” transitions as needed, sequentially (so the remaining transitions remain independent of all those already observed).
- **Step 1**: find a short word $w$ such that reading $w$ from $q$ leads back to state $p$.
- **Step 2**: use $w$ to find a “large” state $\{p_0, \ldots, p_{2d-1}\}$ in the powerset construction.
- **Steps 3-4**: hope that $b$-transitions from the $p_i$ lead to independent cycles of length at $\Theta(n^{1/2})$, with pairwise coprime lengths; this leads to $\Omega(n^d)$ states in the powerset construction.
- **Step 5**: hope that the distribution of terminal states in cycles ensures these states remain after minimization.
- We ensure that, conditioned on the previous steps being successful, each new step succeeds with probability bounded away from 0.
Requests on the graph structure

Our exploration of the graph makes two types of requests on the graph structure for the deterministic automaton:

- “out” : given a state $u$ and a letter $x$, find out the destination of the $x$-transition from $u$.
Requests on the graph structure

Our exploration of the graph makes two types of requests on the graph structure for the deterministic automaton:

- **“out”**: given a state $u$ and a letter $x$, find out the destination of the $x$-transition from $u$
- **“in”**: given a state $u$, find out all transitions leading to $u$
Requests on the graph structure

Our exploration of the graph makes two types of requests on the graph structure for the deterministic automaton:

- **“out”**: given a state $u$ and a letter $x$, find out the destination of the $x$-transition from $u$
- **“in”**: given a state $u$, find out all transitions leading to $u$
- The exploration algorithm fails if the results of all requests mention more than $O(n^{1/2})$ different states.
Requests on the graph structure

Our exploration of the graph makes two types of requests on the graph structure for the deterministic automaton:

- “out”: given a state $u$ and a letter $x$, find out the destination of the $x$-transition from $u$
- “in”: given a state $u$, find out all transitions leading to $u$

The exploration algorithm fails if the results of all requests mention more than $O(n^{1/2})$ different states.

Conditioned on the algorithm not having failed yet, all still unknown transitions are independent, with destinations uniform among $n - O(n^{1/2})$ states.
Step 1: find a short cycle

Assume the additional $a$-transition is from state $p$ to state $q$. 
Step 1: find a short cycle

- Assume the additional $a$-transition is from state $p$ to state $q$.
- Alternating between “forward” and “backward”, find the states at distance 1, 2, $\ldots$ from $q$ (“out” requests), and the states at distance 1, 2, $\ldots$ to $p$ (“in” requests).
Step 1: find a short cycle

- Assume the additional $a$-transition is from state $p$ to state $q$.
- Alternating between “forward” and “backward”, find the states at distance 1, 2, … from $q$ (“out” requests), and the states at distance 1, 2, … to $p$ (“in” requests).
- **Fail**: if more than $cn^{1/2}$ states are seen, or if some level in one of the exploration trees fails to have at least $3/2$ as many states as the previous level.
Step 1 : find a short cycle

- Assume the additional $a$-transition is from state $p$ to state $q$.
- Alternating between “forward” and “backward”, find the states at distance $1, 2, \ldots$ from $q$ (“out” requests), and the states at distance $1, 2, \ldots$ to $p$ (“in” requests).
- **Fail** : if more than $cn^{1/2}$ states are seen, or if some level in one of the exploration trees fails to have at least $3/2$ as many states as the previous level.
- **Succeed** : if the new levels of the trees intersect without failure.
Step 1: find a short cycle

- Assume the additional $a$-transition is from state $p$ to state $q$.
- Alternating between “forward” and “backward”, find the states at distance 1, 2, $\ldots$ from $q$ (“out” requests), and the states at distance 1, 2, $\ldots$ to $p$ (“in” requests).
- **Fail**: if more than $cn^{1/2}$ states are seen, or if some level in one of the exploration trees fails to have at least $3/2$ as many states as the previous level.
- **Succeed**: if the new levels of the trees intersect without failure.
- **Result**: the success probability is at least some constant, and in this case, we find a word $w$ of length $O(\log n)$ such that reading $w$ from $q$ leads to $p$. 
Step 2 : find a large state

- Starting from \( p = p_0 \) in the deterministic automaton, read words \( aw \) repeatedly \((2d - 1)\) times
Step 2: find a large state

- Starting from $p = p_0$ in the deterministic automaton, read words $aw$ repeatedly ($2d - 1$ times)

- **Fail**: if this path of length $O(\log n)$ intersects the $O(n^{1/2})$ states previously seen.
Step 2: find a large state

- Starting from $p = p_0$ in the deterministic automaton, read words $aw$ repeatedly ($2d - 1$ times)
- **Fail**: if this path of length $O(\log n)$ intersects the $O(n^{1/2})$ states previously seen.
- **Succeed** otherwise; find states $p_1, \ldots, p_{2d-1}$ such that reading $aw$ from $p_i$ leads to $p_{i+1}$. 
Step 2: find a large state

- Starting from $p = p_0$ in the deterministic automaton, read words $aw$ repeatedly ($2d - 1$ times)
- **Fail**: if this path of length $O(\log n)$ intersects the $O(n^{1/2})$ states previously seen.
- **Succeed** otherwise; find states $p_1, \ldots, p_{2d - 1}$ such that reading $aw$ from $p_i$ leads to $p_{i+1}$.
- **Result**: the success probability is at least some constant (actually the failure probability is $o(1)$). Note that $b$-transitions from $p_i$ are still unknown.
Step 2 : find a large state

• Starting from $p = p_0$ in the deterministic automaton, read words $aw$ repeatedly ($2d - 1$ times)

• **Fail** : if this path of length $O(\log n)$ intersects the $O(n^{1/2})$ states previously seen.

• **Succeed** otherwise; find states $p_1, \ldots, p_{2d-1}$ such that reading $aw$ from $p_i$ leads to $p_{i+1}$.

• **Result** : the success probability is at least some constant (actually the failure probability is $o(1)$). Note that $b$-transitions from $p_i$ are still unknown.

• Also, note that in the powerset construction, reading word $(aw)^{2d}$ from state $\{p\}$ leads to state $P = \{p_0, p_1, \ldots, p_{2d-1}\}$. 
Step 3 : read $b$-transitions

- Starting from each state $p_i$, repeatedly read $b$-transitions until a $b$-cycle is found.
Step 3: read $b$-transitions

- Starting from each state $p_i$, repeatedly read $b$-transitions until a $b$-cycle is found.
- **Fail**: if the $2d$ cycles are not all distinct, or if one requires less than $n^{1/2}$ or more than $2n^{1/2}$ transitions, or if a cycle length is shorter than $1/2n^{1/2}$ or longer than $n^{1/2}$. 
Step 3: read $b$-transitions

- Starting from each state $p_i$, repeatedly read $b$-transitions until a $b$-cycle is found.

- **Fail**: if the $2d$ cycles are not all distinct, or if one requires less than $n^{1/2}$ or more than $2n^{1/2}$ transitions, or if a cycle length is shorter than $1/2n^{1/2}$ or longer than $n^{1/2}$.

- In case of success, we get $2d$ separate $b$-cycles, of lengths $\ell_0, \ldots, \ell_{2d-1}$; the $\ell_i$ are independent, and uniform in $[c'n^{1/2}, cn^{1/2}]$. 

Step 3: read $b$-transitions

- Starting from each state $p_i$, repeatedly read $b$-transitions until a $b$-cycle is found.
- **Fail**: if the $2d$ cycles are not all distinct, or if one requires less than $n^{1/2}$ or more than $2n^{1/2}$ transitions, or if a cycle length is shorter than $1/2n^{1/2}$ or longer than $n^{1/2}$.
- In case of success, we get $2d$ separate $b$-cycles, of lengths $\ell_0, \ldots, \ell_{2d-1}$; the $\ell_i$ are independent, and uniform in $[c'n^{1/2}, cn^{1/2}]$.
- **Result**: the success probability is at least some constant.
Step 3: read $b$-transitions

- Starting from each state $p_i$, repeatedly read $b$-transitions until a $b$-cycle is found.
- **Fail**: if the $2d$ cycles are not all distinct, or if one requires less than $n^{1/2}$ or more than $2n^{1/2}$ transitions, or if a cycle length is shorter than $1/2n^{1/2}$ or longer than $n^{1/2}$.
- In case of success, we get $2d$ separate $b$-cycles, of lengths $\ell_0, \ldots, \ell_{2d-1}$; the $\ell_i$ are independent, and uniform in $[c'n^{1/2}, cn^{1/2}]$
- **Result**: the success probability is at least some constant.
- Note that in the powerset construction, repeatedly reading $b$-transitions from state $P$ leads to states of size $2d$ that are made of one state out of each cycle.
Step 4 : cross fingers about cycle lengths

- **Succeed**: if the cycle lengths $\ell_0, \ldots, \ell_{2d-1}$ are pairwise coprime.
Step 4: cross fingers about cycle lengths

- **Succeed**: if the cycle lengths $\ell_0, \ldots, \ell_{2d-1}$ are pairwise coprime.

- **Lemma (from Tóth, 2015)**: the success probability is asymptotically at least some positive constant.
Step 4: cross fingers about cycle lengths

- **Succeed**: if the cycle lengths $\ell_0, \ldots, \ell_{2d-1}$ are pairwise coprime.
- **Lemma (from Tóth, 2015)**: the success probability is asymptotically at least some positive constant.
- **Consequence**: reading words of the form $b^m$ from state $P$ in the powerset automaton, results in at least $\prod_i \ell_i \geq (n/4)^d$ different states.
Step 4 : cross fingers about cycle lengths

- **Succeed**: if the cycle lengths $\ell_0, \ldots, \ell_{2d-1}$ are pairwise coprime.

- **Lemma (from Tóth, 2015)**: the success probability is asymptotically at least some positive constant.

- **Consequence**: reading words of the form $b^m$ from state $P$ in the powerset automaton, results in at least $\prod_i \ell_i \geq (n/4)^d$ different states.

- These states make up one big cycle in the powerset automaton.
Step 5 : look at terminal states

- We now have a large number of states making up one big cycle \( C \) in the powerset automaton.
Step 5: look at terminal states

- We now have a large number of states making up one big cycle $C$ in the powerset automaton.

- **Fact**: provided each original cycle of the deterministic automaton contains at least one terminal state and one non-terminal state, the cyclic word on $C$ recording which states are terminal or not is primitive (not a power of a shorter word).
Step 5 : look at terminal states

- We now have a large number of states making up one big cycle $C$ in the powerset automaton.
- **Fact**: provided each original cycle of the deterministic automaton contains at least one terminal state and one non-terminal state, the cyclic word on $C$ recording which states are terminal or not is primitive (not a power of a shorter word).
- This occurs with asymptotic probability 1 under our hypotheses, and implies that the states in $C$ are pairwise non-equivalent (they remain distinct under minimization).
Wrapping it up : accessibility

- All we still need is that, in the powerset automaton, the state \( \{ p \} \) is accessible from the initial state \( \{ q_0 \} \).
Wrapping it up: accessibility

- All we still need is that, in the powerset automaton, the state $\{p\}$ is accessible from the initial state $\{q_0\}$.
- This is equivalent to state $p$ being accessible from $q_0$ in the deterministic automaton (only paths going through $p$ "see" the nondeterministic transition).
Wrapping it up: accessibility

- All we still need is that, in the powerset automaton, the state \( \{p\} \) is accessible from the initial state \( \{q_0\} \).
- This is equivalent to state \( p \) being accessible from \( q_0 \) in the deterministic automaton (only paths going through \( p \) “see” the nondeterministic transition).
- For this it is sufficient that \( p \) belongs to the giant strongly connected component, which is unique and large with asymptotic probability 1 (thus unaffected by our conditioning by events of probability far from 0 or 1).
Wrapping it up: accessibility

- All we still need is that, in the powerset automaton, the state \{p\} is accessible from the initial state \{q_0\}.
- This is equivalent to state \(p\) being accessible from \(q_0\) in the deterministic automaton (only paths going through \(p\) “see” the nondeterministic transition).
- For this it is sufficient that \(p\) belongs to the giant strongly connected component, which is unique and large with asymptotic probability 1 (thus unaffected by our conditioning by events of probability far from 0 or 1).
- This concludes the proof!
Our result is not that surprising, but requires a detailed analysis...

Most of the proof sketch feels like the “explosion” probability should actually be pretty high, possibly as high as $\nu_k^2$ (probability that both states in the additional transition lie in the giant component); this would require a more detailed analysis though.

The most unsatisfying part of our proof is the need for a model that means the original automaton has many terminal states; it would be nice if a single terminal state was sufficient.
Dernière remarque

Merci à Nicolas pour ces années à la tête du GT!