On the Uniform Random Generation of Determinisitic Partially Ordered Automata using Monte Carlo Techniques

Pierre-Cyrille Héam, Jean-Luc Joly

On the Uniform Random Generation of Determinisitic Partially Ordered Automata using Monte Carlo Techniques

Pierre-Cyrille Héam and Jean-Luc Joly

FEMTO-ST, CNRS UMR 6174, Université de Franche-Comté, INRIA
16 route de Gray, 25030 Besançon Cedex, France
pheam@femto-st.fr -- jean-luc.joly@femto-st.fr

Abstract. Partially ordered automata are finite automata admitting no simple loops of length greater than or equal to 2. In this paper we show how to randomly and uniformly generate deterministic accessible partially ordered automata using Monte-Carlo techniques.

1 Introduction

The random generation of data inputs is a general way to test both the (average) efficiency of algorithms and the correctness of the implementations. For non numerical complex data structures, like trees or graphs, the random generation is usually performed using either a combinatorial approach [FS08] or a probabilistic approach [Jer98]. Finite automata are widely used to handle many issues from verification to text processing, requiring the development of dedicated optimized algorithms. The development of algorithms for generating several classes of finite automata is therefore a challenging problem.

A partially ordered automaton is a finite automaton for which there exists a partial ordered relation $\preceq$ on its set of states such that, if there is a transition from p to q, then $p \preceq q$. Equivalently, in a partially ordered automaton all simple loops have length 1. In this paper, we address the problem of randomly generate accessible deterministic partially ordered automata using a Markov chain approach. More precisely, we define in Section 4.1 a Markov chain on the set of deterministic accessible partially ordered automata with n states having the uniform distribution for stationary distribution. Moreover, moving in this Markov chain can be done in time $O(n)$. In Section 5.2, we use a statistical approach to point out a conjecture on the mixing time of this Markov Chain. As it will be incidentally exposed, for this class of automata, the combinatorial approach can lead to an efficient random sampler (the formula are given in Section 3). However, the Markov Chain approach is more flexible and can be easily adapted for several subclasses of automata, for instance by requiring a maximal number of loops.

Related work The enumeration of deterministic finite automata has been first investigated in [Vys59] and was applied to several subclasses of deterministic finite

automata [Kor78,Kor86,Rob85,Lis06],... Several works focus on the random generation of accessible deterministic automata [Nic00,CP05,BN07,BDN09,CN12]; the interested reader is referred to the recent survey [Nic14] for more information. As far as we know, the only work using a Markov chain to randomly generate finite automata is [CF11] for deterministic accessible acyclic automata, extended in [CF12] for minimal acyclic automata.

Partially ordered automata are an interesting class of finite automata used in the classification of regular languages [STV01,Arf87] as well as for model-checking purposes [BMT07,CHM08] and in trace theory [GRS04].

Layout of the paper. Section 2 introduces the necessary formal background. Section 3 is dedicated to the computation of the number of loops in deterministic partially ordered automata, what will be useful for the statistical tests developed in Section 5. The Monte Carlo algorithm proposed to randomly generate deterministic accessible partially ordered automata is exposed in Section 4.1 and a variant ensuring a bound on the maximal number of loops is developed in Section 4.2. Finally some experimental results to statistically evaluate the mixing time of the algorithm are proposed in Section 5.

2 Formal Background

Pearson's Chi-square Test. The χ^2 test is most commonly used to test the nature of a statistical distribution from which some random sample is drawn. Situations typically arise when data can be classified into one of k bins or classes, with probability p_i of falling into the bin b_i. If the classes are exhaustive then $\sum p_i = 1$. For a total of N observations, one has $\sum n_i = N$, where n_i denotes the number of observations falling in b_i. It can be shown that the quantity $\zeta = \sum_{i=1}^{k} \frac{(n_i - N \cdot p_i)^2}{N \cdot p_i}$ has approximately the χ^2 distribution with $k - r$ degrees of freedom, where r is the number of constraints used to estimate the p_i from the data. Since $\sum p_i = 1$, r is at least 1. This particular form of the Chi-square test is called the *Pearson's Chi-square Test.* Following the use in statistics, one will note χ^2 instead of ζ in the rest of this paper. It's important to notice that the mean value and variance of χ^2_d, the χ^2-distribution with d degrees of freedom, are respectively d and $2d$. Furthermore, for large enough values of d, χ^2_d approaches, as predicted by the Central Limit Theorem, a Gaussian distribution. In section 5.2, the fact that $\sum_{i=1}^{k} \frac{(n_i - N \cdot p_i)^2}{d \cdot N \cdot p_i}$ is close to 1 is reassuring, one has nevertheless to keep in mind the relatively large variance of the distribution. It should be either recalled that the probability distribution of χ^2_d is the function $f_d(t) = \frac{t^{(d/2)-1} \cdot \exp\left(-t^2/2\right)}{2^{t/2}\Gamma(d/2)}$. The test consists in choosing a value of the significance level α (0.05 is usually adopted), then to compute or read in an appropriate table the value of $t_{d,\alpha}$ such that $\int_{t_{d,\alpha}}^{\infty} f_d(t)\, \mathrm{d}t = \alpha$. If $\chi^2 > t_{d,\alpha}$, either a statically improbable large value has been obtained, for $\alpha = 0.05$ this case occurs in 5% of the cases involved, or the sampler doesn't fit the expected distribution, which happens in 95% of the cases. The test provides a second informative value, called the p-value which is defined

by p-value $= \int_{\chi^2}^{\infty} f_d(t)\ \mathrm{d}t$. The p-value associated with the χ^2 obtained, is the probability to obtain a value greater than χ^2. Figure 1 highlights the notions set out above.The reader interested in these statistical aspects can refer to [GN96].

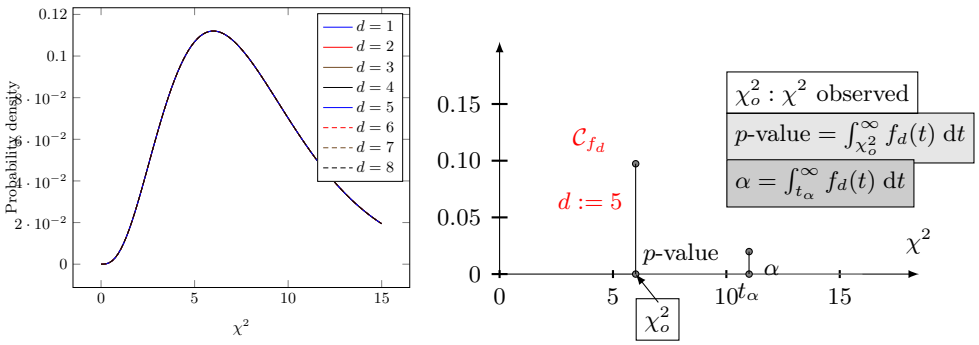


Fig. 1. χ^2_d distributions, α, $t_{d,\alpha}$ and p-value

Deterministic, Accessible, Partially Ordered Automata. A *finite automaton* is a tuple (Q, Σ, E, I, F) where Q is a finite set of states, Σ a finite alphabet containing at least 2 letters, $E \subseteq Q \times \Sigma \times Q$ is the set of transitions, $I \subseteq Q$ is the subset of initial states and $F \subseteq Q$ is the subset of final states. A finite automaton (Q, Σ, E, I, F) is *s-deterministic* if $|I| = s \in \mathbb{N}$ and if for each state $q \in Q$ and each letter $a \in \Sigma$, there exists at most one state p such that $(q, a, p) \in E$. An 1-deterministic automaton is called a *deterministic* automaton. A finite automaton is *accessible* if for each state q there exists a path from an initial state to q. A finite automaton (Q, Σ, E, I, F) is *partially ordered* if there exists an antisymetric, reflexive and transitive relation $\preceq$ on Q such that if $(p, a, q) \in E$, then $p \preceq q$. In an accessible partially ordered automaton, the set of initial states is exactly the set of minimal states for $\preceq$. Two finite automata are *isomorphic* if there are equal up to state's names: there exists a bijection between their sets of states preserving the transitions, initial and final states. The set of deterministic accessible partially ordered automata, DPOAs for short, whose set of states is $[n]$ (the set $\{i,\ 1 \leqslant i \leqslant n\}$), and whose initial state is 1 is denoted $\overline{\mathbb{P}_n}$. The subset of $\overline{\mathbb{P}_n}$ which has no final states is denoted $\mathbb{P}_n$. Each DPOA is isomorphic to exactly $(n-1)!$ other DPOAs. Therefore generating DPOAs or DPOAs up to isomorphism is the same problem.

Markov Chains A *Markov kernel* on a finite set Ω is a a funcion P from Ω^2 into $[0, 1]$ such that, for every $x \in \Omega$, $\sum_{y \in \Omega} P(x, y) = 1$. The graph G of a Markov kernel is the directed graph whose set of vertices is Ω and there is an edge from x to y if $P(x, y) > 0$. A distribution π on Ω is *stationnary* for the Markov kernel if for every x, $\sum_{y \in \Omega} \pi(x)P(x, y) = \pi(x)$. We inductively define P_n as follows:

Algorithm 1 MCMC(t : integer, P : MarkovKernel, x : elementofΩ)

1: $r \leftarrow x$
2: $i \leftarrow 0$
3: **while** $i < t$ **do**
4: $\quad i \leftarrow i + 1$
5: $\quad$ Choose y in Ω with probability $P(r, y)$
6: $\quad r = y$
7: **end while**
8: **return** r

$P_1 = P$ and for every $x, y \in \Omega$, $P_n(x, y) = \sum_{z \in \Omega} P_{n-1}(x, z)P(z, y)$. A Markov kernel is *irreducible* if G is strongly connected. It is *aperiodic* if for every $x \in \Omega$, $\gcd\{\mathrm{t} \mid \mathrm{t} > 0 \text{ and } \mathrm{P_t(x, x)} > 0\} = 1$. An irreductible and aperiodic Markov kernel is called *ergodic*. One can notice that if for every x, $P(x, x) \neq 0$, then P is aperiodic. A *Markov chain* with state space Ω and kernel P, is a sequence of random variables $(X_0, X_1, \cdots)$ such that for all $t \geqslant 1$, for all $(x_0, \ldots, x_t, y) \in \Omega^{t+2}$, $P\{X_{t+1} = y | X_{t_i} = x_i,\ 0 \leq i \leq t\} = P(x_t, y)$.

Proposition 1. *If P is an ergodic Markov kernel on Ω, then there exists a unique stationnary distribution. Moreover, if $P(x, y) = P(y, x)$ for every $x, y \in \Omega$, then the stationnary distribution is the uniform distribution on Ω.*

Theorem 1. *If an ergodic Markov kernel P on Ω has π for stationnary distribution, then*

$$\max \|P_t(x, \cdot) - \pi\|_{\mathrm{TV}} \underset{t \to +\infty}{\to} 0.$$

Where $\|\|_{TV}$ designates the total variation distance between two distributions ([LPW09]). Theorem 1 leads to the design of probabilistic algorithms dedicated to the random generation of elements of Ω according to a distribution π. These algorithms are called *Monte-Carlo Markov Chains* (MCMC for short) algorithms. Their general scheme is detailed in Algorithm 1 : let P be an ergodic Markov kernel on Ω which stationary distribution is π; the algorithm returns an element of Ω following the distribution $P_t(x, \cdot)$, which is close to π by Theorem 1.

Choosing t in Algorithm 1 is a challenging question depending both on how close to π we want to be and on the convergence rate of $P_t(x, \cdot)$ to π. This leads to define the mixing time of a Markov Kernel. For any $\varepsilon > 0$, the *mixing time* $t_{\mathrm{mix}}(\varepsilon)$ of an ergodic Markov kernel P whose stationnary distribution is π is

$$t_{\mathrm{mix}}(\varepsilon) = \min\{t \mid \max_{x \in \Omega} \|P_t(x, \cdot) - \pi\|_{TV} \leq \varepsilon\}.$$

Intuitively, this is the minimal t such that running Alogrithm 1, with this t and with any x, returns an element of Ω with a distribution differing from π by at most ε. Computing mixing time bounds is a central question on Markov Chains [JS96,BD97,BDX04,EMT13,LPW09]. Without the mention of ε, t_{mix}, also called *mixing time* refers to the value $t_{\mathrm{mix}}(1/4)$. The two notions are linked by the inequality $t_{\mathrm{mix}}(\varepsilon) \leqslant \lceil \log_2 \varepsilon^{-1} \rceil t_{\mathrm{mix}}$.

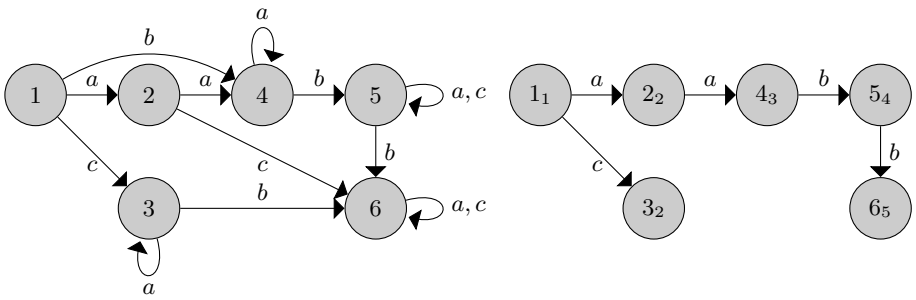


Fig. 2. A 6-states, 3-letters DPOA and its source tree

3 Loops Combinatorics, sources and trees

One key point to design the Pearson's Chi-square test further explored in section 5.2, is to get the loops combinatoric, i.e. the number of n states, k-alphabet DPOAs with a specified number of loops. To achieve this goal, the approach used in [DFN13] to count acyclic deterministic automata can easily be adapted. This combinatorial approach is based on the notion of secondary sources and could allow us in addition to build and efficient DPOAs sampler, which is not our goal in this paper. Let $x = ([n], \Sigma, E, \{1\}) \in \mathbb{P}_n$, $p \in Q$, then p is a *secondary source*, or an *order*-2-*source*, if $p \neq 1$ and if all its incoming transitions, that aren't a loop, come from 1. It's then possible to define the same way the *order*-3-*sources*: $p \in Q$ is an *order*-3-*source* if p is not an order-2-source and if, after the pruning of 1 and all its outgoing transitions, all its incoming transitions, that aren't a loop, come from an order-2-source. This notion can be recursively generalized so that, for each state $p \in Q$, there exists $k \in [n]$ such that p is an order-k-source. These remarks support the foundation of equations 1, 2 and 3. Furthermore, the notion of sources enables us to associate to each $x \in \mathbb{P}_n$ its *source tree*, i.e. the only DPOA $\sigma(x)$ obtained from x by keeping, for each order-$(k+1)$-source q, the smallest (for the lexicographic order) transition (p, a, q) where p is an order-k-source. Figure 2 depicts a DPOA x and its source tree $\sigma(x)$. Notice that $\sigma(x)$ is actually a tree. The source tree is a useful notion in lemma's 4 proof.

With $b_k(n)$ denoting the number of n-states, k-letters, s-deterministic ($s \in [n]$) partially ordered automata, then

$$b_k(n) = \sum_{t=0}^{n-1} \binom{n}{t} (-1)^{n-t-1} (t+2)^{k(n-t)} b_k(t), \tag{1}$$

$\eta_k(n, s)$ denoting the number of n-states, k-letters, s-deterministic partially ordered automata, that have exactly s order-1-sources, one obtains the following formula

$$\eta_k(n, s) = \binom{n}{s} \sum_{i=0}^{n-1} \binom{n-s}{i} (-1)^i b_k(n-s-i)(n-s-i+2)^{k(s+i)} \tag{2}$$

	Number of loops							
Number of states	0	1	2	3	4	5	6	Total
2	3	8	7	2	0	0	0	20
3	32	100	114	56	10	0	0	312
4	762	2640	3528	2268	702	84	0	9984
5	32712	122064	181848	138240	56640	11904	1008	544416

Table 1. Numbers of loops in 2-letters DPOAs

In equation 2, one only has to mark the loops with letter u to get a polynomial expression of the variable u

$$\eta_k(n,s,u) = \binom{n}{s} \sum_{i=0}^{n-1} \binom{n-s}{i} (-1)^i b_k(n-s-i)(n-s-i+1+u)^{k(s+i)} \quad (3)$$

Table 1 displays the results for n-states, 2-letters DPOAs, $n \in [2,5]$.

4 Random Generation Using Markov Chains

4.1 Random Generation of DPOAs

The random generation will be performed on $\mathbb{P}_n$, final states are then randomly added. Let $x \in \mathbb{P}_n$ and (p,a,q) be a triplet such that $p,q \in [n]$ and $a \in \Sigma$. the following cases may arise:

- The triplet (p,a,q) is a transition of x. Let y be the automaton obtained from x by removing the transition (p,a,q). If y is in $\mathbb{P}_n$, y is denoted $x*(p,a,q)$; otherwise $x*(p,a,q)$ is not defined.
- The triplet (p,a,q) is not a transition of x and there is no transition of the form (p,a,r) in x, with $r \in [n]$. Let y be the automaton obtained from x by adding the transition (p,a,q). If y is in $\mathbb{P}_n$, y is denoted $x*(p,a,q)$; otherwise $x*(p,a,q)$ is not defined.
- The triplet (p,a,q) is not a transition of x and there is a transition of the form (p,a,r) in x, with $r \in [n]$ (of course $r \neq q$). Let y be the automaton obtained from x by removing the transition (p,a,r) and adding the transition (p,a,q). If y is in $\mathbb{P}_n$, then $x*(p,a,q)$; otherwise $x*(p,a,q)$ is not defined.

Note that if x has at least two states, there exists a transition of the form (p,a,q) with $p \neq q$; the automaton $x*(q,a,p)$ doesn't exist since simple loops of length 2 are not allowed. Therefore, there is at least one transition t such that $x*t$ is undefined.

Lemma 1. *Let $x \in \mathbb{P}_n$ and (p,a,q) be a triplet such that $1 \leq p,q \leq n$ and $a \in \Sigma$. If $x*(p,a,q)$ exists, then $(x*(p,a,q))*(p,a,q)$ exists and is equal to x.*

Proof. It's a direct result of the operator's $*$ definition.

We define the Markov kernel K on $\mathbb{P}_n$ as follows: for any $x, y \in \mathbb{P}_n$, if there exists (p, a, q) such that $y = x * (p, a, q)$, then $K(x, y) = \frac{1}{|\Sigma|n^2}$. Otherwise, and if $y \neq x$, then $K(x, y) = 0$. And $K(x, x) = 1 - \sum_{y \neq x} K(x, y)$. Since there are at most $|\Sigma|n^2$ possible triplets, $K(x, x)$ is non-negative. According to the above remark, $x * t$ is undefined for at least one transition, proving that $K(x, x)$ is positive. Lemma 2 is a direct consequence of the kernel K definition and of Lemma 1.

Lemma 2. *For $n \geq 2$ and for each $(x, y) \in \mathbb{P}_n^2$, $K(x, y) = K(y, x)$ and K is aperiodic.*

Definition 1. *Let $x = ([n], \Sigma, E, \{1\}) \in \mathbb{P}_n$, then x is called* serialized *if there's at most one outgoing transition from each state and E contains no loop. In a serialized DPOA x, the underlying order is then total and x can be pictured as $1 = p_0 \overset{a_1}{\to} p_1 \overset{a_2}{\to} \cdots \overset{a_{n-1}}{\to} p_{n-1}$, where $a_i \in \Sigma$ and $\{p_i\} = [n]$.*

Lemma 3. *Let $x, y \in \mathbb{P}_n$ be two serialized DPOAs, then there's a path from x to y in G.*

Proof. Let x and y be two different serialized DPOAs respectively pictured by $1 = p_0 \overset{a_1}{\to} p_1 \overset{a_2}{\to} \cdots \overset{a_{n-1}}{\to} p_{n-1}$ and $1 = q_0 \overset{b_1}{\to} q_1 \overset{b_2}{\to} \cdots \overset{b_{n-1}}{\to} q_{n-1}$. Let $i = \delta(x, y)$ be the smallest index such that $(a_i, p_i) \neq (b_i, q_i)$ then, in particular $p_{i-1} = q_{i-1}$. If $p_i = q_i = r$ then $a_i \neq b_i$, $z = x * (p_{i-1}, b_i, r) * (p_{i-1}, a_i, r)$ exists, is serialized and $\delta(z, y) > \delta(x, y)$. If $p_i \neq q_i$, then one can consider $j \in [n] \backslash \{i\}$ such that $q_i = p_j$. Then, if $a_i \neq b_i$, for any $a \in \Sigma$, $z = x * (p_{i-1}, b_i, p_j) * (p_{j-1}, a_j, p_j) * (p_{n-1}, a, p_i) * (p_{i-1}, a_i, p_i)$ exists, is serialized and $\delta(z, y) > \delta(x, y)$. If however $a_i = b_i$, one can consider $c \in \Sigma \backslash \{a_i\}$, and $x' = x * (p_{i-1}, c, p_i) * (p_{i-1}, a_i, p_i)$. It's then possible to apply the former process from x'. From x it's then possible to define a sequence (x_i) of serialized DPOAs ($x_0 = x$), such that the sequence $\delta(x_i, y)$ is strictly increasing. Since $\delta(x, y) \leqslant n - 1$, for each pair (x, y) of serialized DPOAs, the sequence (x_i) stops in less than $n - 1$ steps on y.

Lemma 4. *Let $x \in \mathbb{P}_n$, then there's a path in G from x to a serialized DPOA.*

Proof. Let $x \in \mathbb{P}_n$, then there's a path from x to its sources tree $\sigma(x)$ as far as the pruning of x to obtain $\sigma(x)$ matches with acceptable transitions removals. If $\sigma(x) = ([n], \Sigma, E_\sigma, \{1\})$ is not serialized, let $p \in [n]$ a state having in $\sigma(x)$ at least 2 outgoing transitions. It's then possible to consider q_1, q_2 two different children of the node p and $(p, a, q_2) \in E_\sigma$. Let l be a leaf of $\sigma(x)$ such that q_1 is one of its ancestors, or possibly $l := q_1$. Then $z = \sigma(x) * (l, a, q_2) * (p, a, q_2)$ exists, is a tree and the state p is one child less than in $\sigma(x)$. The same process can be performed repeatedly until a serialized DPOA is obtained.

Proposition 2. *The Markov kernel K is irreducible.*

Proof. Let $(x, y) \in \mathbb{P}_n^2$, then a path from x to a serialized DPOA $s(x)$ and a path from y to a serialized DOPA $s(y)$ exists (Lemma 4). The path from y to $s(y)$ is reversible (Lemma 1). A path from $s(x)$ to $s(y)$ exists (Lemma 3). Finally there's a path from x to y in G and K is irreducible.

Proposition 3. *The diameter of G is in $\Theta(n)$.*

Proof. The distance in G between two different serialized automata is at least $n-1$ if all the transitions are different, which gives the lower bound. The distance between a DPOA x to $\sigma(x)$ is smaller than $|\Sigma|n$. According to the proof of lemma 4, from $\sigma(x)$ to a serialized DPOA the distance is smaller than $2(n-1)$. The proof of lemma 3, likewise, ensures that the distance between two serialized automata is smaller than $6n$. One can conclude that the distance between two DPOAs is smaller than $2(|\Sigma| + 5)n$.

Proposition 4. *The Markov kernel K is ergodic and its stationnary distribution is the uniform distribution on $\mathbb{P}_n$.*

Proof. Proposition 2 and lemma 2 ensure that the Markov chain is ergodic and symmetric, its stationary distribution is therefore the uniform distribution.

4.2 Random Generation of DPOAs with at most m loops

We denote by $\mathbb{P}_n^{[\leq m]}$ the subclass of $\mathbb{P}_n$ which have at most m loops. The random generation will be performed on $\mathbb{P}_n^{[\leq m]}$ and final states are then randomly added. Let $x \in \mathbb{P}_n^{[\leq m]}$ and (p, a, q) be a triplet such that $p, q \in [n]$ and $a \in \Sigma$. The following cases may arise:

- The triplet (p, a, q) is a transition of x. Let y be the automaton obtained from x by removing the transition (p, a, q). If y is in $\mathbb{P}_n^{[\leq m]}$, y is denoted $x \circ (p, a, q)$; otherwise $x \circ (p, a, q)$ is not defined.
- The triplet (p, a, q) is not a transition of x and there is no transition of the form (p, a, r) in x, with $r \in [n]$. Let y be the automaton obtained from x by adding the transition (p, a, q). If y is in $\mathbb{P}_n^{[\leq m]}$, y is denoted $x \circ (p, a, q)$; otherwise $x \circ (p, a, q)$ is not defined.
- The triplet (p, a, q) is not a transition of x and there is a transition of the form (p, a, r) in x, with $r \in [n]$ (of course $r \neq q$). Let y be the automaton obtained from x by removing the transition (p, a, r) and adding the transition (p, a, q). If y is in $\mathbb{P}_n^{[\leq m]}$, then $x \circ (p, a, q)$; otherwise $x \circ (p, a, q)$ is not defined.

The proof of the following result is left to the reader.

Lemma 5. *Let $x \in \mathbb{P}_n^{[\leq m]}$ and (p, a, q) be a triplet such that $p, q \in [n]$ and $a \in \Sigma$. If $x \circ (p, a, q)$ exists, then $(x \circ (p, a, q)) \circ (p, a, q)$ exists too and $(x \circ (p, a, q)) \circ (p, a, q) = x$.*

We define the Markov kernel $K^{[\leq m]}$ on $\mathbb{P}_n^{[\leq m]}$ as follows: for any $x, y \in \mathbb{P}_n^{[\leq m]}$, if there exists (p, a, q) such that $y = x \circ (p, a, q)$, then $K^{[\leq m]}(x, y) = \frac{1}{|\Sigma| n^2}$. Otherwise, and if $y \neq x$, then $K^{[\leq m]}(x, y) = 0$. The same arguments as those exposed in Section 4.1 ensure that $K^{[\leq m]}$ is aperiodic and reversible.

Proposition 5. *The Markov kernel $K^{[\leq m]}$ is ergodic and its stationary distribution is the uniform distribution on $\mathbb{P}_n^{[\leq m]}$.*

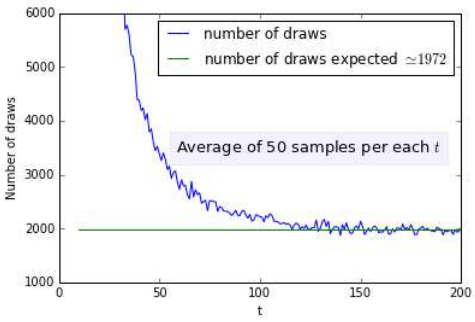
Proof. Irreducibility is the only point which remains to be proven. Let $x, y \in \mathbb{P}_n^{[\leq m]}$, $l(x)$ and $l(y)$ be the numbers of loops in x and y respectively, then $\max(l(x), l(t)) \leq m$. Furthermore, for every vertex z of the path between x and y described in the proof of Proposition 2, one has $l(z) \leq m$.

5 Statistical tests

5.1 Two simple tests for small number of states

For a small number of states, it's possible to generate samples much larger than $|\mathbb{P}_n|$. It's then easy to implement simple tools to test the convergence of the Markov Chain to the stationary distribution (the uniform distribution here).

Coupon collector's test The average number of draws to collect the complete set of different m objects is : $\mu = m \sum_{k=1}^{m} 1/k$, if one assumes that each new object is chosen uniformly and independently from the set of m possible types. The principle of the test is to get the number of necessary draws to obtain all of the m objects of the collection and to compare this number with μ. The following figure displays the results for 3-states, 2-letters DPOAs. Above 100, t appears to provide good mixing. To confirm this initial estimate, it's possible to perform an entropy test.



Entropy test Let s be the size of a sample. If the sampler is perfect, each DPOA appears at about the same frequency $s/|\mathbb{P}_n|$. With $m = |\mathbb{P}_n|$, the sample's entropy is then defined by: $\text{entropy}(s) = -\sum_{i \in [m]} f_i \log_m(f_i)$, where f_i is the frequency of the DPOA numbered by i. For a perfect sampler, since $f_i = \frac{1}{m}$, one has $\text{entropy}(s) = 1$. For $t = 10, 50, 100, 200$, one has respectively $\text{entropy}(2000) \simeq 0.88, 0.97, 0.985, 0.987$. The entropy test confirms that around the theoretical value μ, for an estimated value of the mixing time, the frequencies of each DPOA is correct. Since these tests suppose an exhaustive generation of $\mathbb{P}_n$, they can not be extended to larger scales.

5.2 Testing for larger scales

MCMC or recursive techniques provide samplers for large sets that are impossible to build exhaustively. Even for a relatively small number of states $|\mathbb{P}_n|$ is very large. For instance, there're about 10^{45} 20-state, 2-letter, DPOAs. Then, one solution to taylor a statistical test, lies in partioning the set of objects in different classes to perform a Pearson's Chi-square test.

To achieve a result about the accuracy of the MCMC sampler, the sample obtained is sliced in a partition according to the loops' number in each automaton. Nevertheless, the partition has to be suited to the Cochran rule, meaning

that each bin of the partition has to contain, according to the theoretical frequency, at least 5 automata. Reaching this goal supposes to gather the too small sized bins. For instance, a sample of $10,000$ 14-state, 2-letter DPOAs, gives the theoretical weights of the 9 to 15 loop bins displayed in the table 2. To comply

loop(s)	9	10	11	12	13	14	15
weight	6.68	0.87	0.08	$6 \cdot 10^{-3}$	$2 \cdot 10^{-4}$	$7 \cdot 10^{-6}$	$9 \cdot 10^{-8}$

Table 2. weights of the looping-bins, 14-states, 2-letters DPOA, $10,000$ samples

with the Cochran rule, the bins 9 to 15 shall be grouped in a single bin. Our purpose is to check for which values of t the sampler is correct. The outcomes in table 3 result from the computation of χ^2 and p-values for $10,000$ automata per drawing, the number of objects in each looping-bin for each value of t, is a calculated average on the base of 20 random drawings. According to Table 2, for a sample of $10,000$ 14-state, 2-letter DPOAS, there're 10 bins, it follows that $d = 9$. Furthermore, with $\alpha = 0.05$ one has $t_\alpha = 16.9190$ and $t_\alpha/d \simeq 1.88$. Table

t	50	100	150	200	250	300	350	400	450	500
χ^2/d	3121.49	632.46	194.1	66.85	25.22	10.12	3.57	1.68	0.6	0.26
p-value	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$	0.0002	0.09	0.79	0.98

Table 3. χ^2 goodness of fit test for 14-states, 2-letters DPOAs

3 suggests that $t \in [400, 450]$ provides an accurate picture of the mixing time for this particular kind of DPOAs.

The processing[1] of the results in Table 3 leads to a general empirical strategy to approach the mixing time value for DPOAs when numbers of states and letters are fixed. In the Pearson's Chi-square test, it suffices to compute χ^2 for increasing values of t. Furthermore, the results for 14-state, 2-letter DPOAs sketches out a mixing time lying somewhere between $O(n^2)$ and $O(n^3)$. These observations provide a guideline for the next experiments.Three different values of t, n^2, n^3 and $10 \cdot n^3$ have been tested for different DPOAs. Table 4 confirms the intuition about the mixing time rough order of magnitude.

6 Conclusion

In this paper we proposed a Markov chains based approach to randomly generate accessible deterministic partially ordered automata. The main advantage of Markov chains techniques, compared to combinatorial techniques, is the great flexibility of the tool. We showed for instance how to generate DPOAs with a fixed maximal of loops; the approach can be adapted to other subclasses of DPOAs, it suffices to check the ergodicity of the chain. However, the main problem of Markov chains techniques is to evaluate the mixing time representing the number of iterations required to ensure that the result is close to the stationary

[1] Computations have been performed on the supercomputer facilities of the Mésocentre de calcul de Franche-Comté.

n	9				11				13				15			
$\lvert\Sigma\rvert = 2$	d	t_α/d	χ^2/d	p-val	d	t_α/d	χ^2/d	p-val	d	t_α/d	χ^2/d	p-val	d	t_α/d	χ^2/d	p-val
$T = n^2$	9	1.88	2700	0	9	1.88	2073	0	10	1.83	1706	0	10	1.83	1187	0
$T = n^3$	9	1.88	0.27	0.98	9	1.88	1.85	0.05	10	1.83	1.84	0.047	10	1.83	0.53	0.86
n	17				19				21				23			
$\lvert\Sigma\rvert = 2$	d	t_α/d	χ^2/d	p-val	d	t_α/d	χ^2/d	p-val	d	t_α/d	χ^2/d	p-val	d	t_α/d	χ^2/d	p-val
$T = n^2$	11	1.79	914	0	11	1.79	765	0	11	1.79	685	0	11	1.79	580	0
$T = n^3$	11	1.79	0.89	0.5	11	1.79	0.79	0.64	11	1.79	0.47	0.91	11	1.79	0.83	0.61
n	9				11				13				15			
$\lvert\Sigma\rvert = 3$	d	t_α/d	χ^2/d	p-val	d	t_α/d	χ^2/d	p-val	d	t_α/d	χ^2/d	p-val	d	t_α/d	χ^2/d	p-val
$T = n^3$	13	1.72	0.97	0.48	14	1.69	0.46	0.95	15	1.66	0.92	0.54	15	1.66	0.85	0.61
n	9				11				13				15			
$\lvert\Sigma\rvert = 5$	d	t_α/d	χ^2/d	p-val	d	t_α/d	χ^2/d	p-val	d	t_α/d	χ^2/d	p-val	d	t_α/d	χ^2/d	p-val
$T = n^2$	20	1.57	10^5	0	21	1.55	88315	0	22	1.54	72035	0	23	1.53	52843	0
$T = n^3$	20	1.57	3.925	$\simeq 0$	21	1.55	1.46	0.05	22	1.54	0.9	0.5	23	1.53	1.46	0.054
$T = 10 \cdot n^3$	20	1.57	1.024	0.36	21	1.55	0.75	0.73	22	1.54	0.79	0.6	23	1.53		

Table 4. Chi-square results for different DPOAs, $\alpha = 0.05$, $2 \cdot 10^5$ draws

distribution. Statistical tests seem to show that the proposed Markov Chain has a mixing time bounded by $O(n^3)$, where n is the number of states of the DPOA. We conjecture the mixing time is actually in $O(n^2 \log(n))$ and we plan to prove it in a future work.

References

[Arf87] Mustapha Arfi. Polynomial operations on rational languages. In *STACS 87, 4th Annual Symposium on Theoretical Aspects of Computer Science*, volume 247 of *Lecture Notes in Computer Science*, pages 198–206. Springer, 1987.

[BD97] Russ Bubley and Martin Dyer. Path coupling: A technique for proving rapid mixing in markov chains. In *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*, pages 223–231. IEEE, 1997.

[BDN09] F. Bassino, J. David, and C. Nicaud. Enumeration and random generation of possibly incomplete deterministic automata. *Pure Mathematics and Applications*, 19:1–16, 2009.

[BDX04] S. Boyd, P. Diaconis, and L. Xiao. Fastest mixing markov chain on a graph. *SIAM Review*, 46(4):667–689, 2004.

[BMT07] Ahmed Bouajjani, Anca Muscholl, and Tayssir Touili. Permutation rewriting and algorithmic verification. *Inf. Comput.*, 205(2):199–224, 2007.

[BN07] F. Bassino and C. Nicaud. Enumeration and random generation of accessible automata. *Theor. Comput. Sci.*, 381(1-3):86–104, 2007.

[CF11] V. Carnino and S. De Felice. Random generation of deterministic acyclic automata using markov chains. In *CIAA 2011*, volume 6807 of *Lecture Notes in Computer Science*, pages 65–75, 2011.

[CF12] Vincent Carnino and Sven De Felice. Sampling different kinds of acyclic automata using markov chains. *Theor. Comput. Sci.*, 450:31–42, 2012.

[CHM08] Gérard Cécé, Pierre-Cyrille Héam, and Yann Mainier. Efficiency of automata in semi-commutation verification techniques. *ITA*, 42(2):197–215, 2008.

[CN12] A. Carayol and C. Nicaud. Distribution of the number of accessible states in a random deterministic automaton. In *STACS 2012*, volume 14 of *LIPIcs*, pages 194–205, 2012.

[CP05] J.-M. Champarnaud and Th. Paranthoën. Random generation of dfas. *Theor. Comput. Sci.*, 330(2):221–235, 2005.

[DFN13] Sven De Felice and Cyril Nicaud. Random generation of deterministic acyclic automata using the recursive method. In *Computer Science Theory and Applications*, volume 7913 of *Lecture Notes in Computer Science*, pages 88–99. 2013.

[EMT13] P. L. Erdős, I. Miklós, and Z. Toroczkai. A decomposition based proof for fast mixing of a Markov chain over balanced realizations of a joint degree matrix. *ArXiv e-prints*, July 2013.

[FS08] Ph. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2008.

[GN96] P. E. Greenwood and M. S. Nikulin. *A Guide to Chi-Squared Testing*. Wiley, New York, NY, 1996.

[GRS04] Giovanna Guaiana, Antonio Restivo, and Sergio Salemi. On the trace product and some families of languages closed under partial commutations. *Journal of Automata, Languages and Combinatorics*, 9(1):61–79, 2004.

[Jer98] Mark Jerrum. Mathematical foundations of the markov chain monte carlo method. In *Probabilistic Methods for Algorithmic Discrete Mathematics*, volume 16 of *Algorithms and Combinatorics*, pages 116–165. 1998.

[JS96] Mark Jerrum and Alistair Sinclair. The markov chain monte carlo method: an approach to approximate counting and integration. *Approximation algorithms for NP-hard problems*, pages 482–520, 1996.

[Kor78] D. Korshunov. Enumeration of finite automata. *Problemy Kibernetiki*, 34:5–82, 1978.

[Kor86] A. D. Korshunov. On the number of non-isomorphic strongly connected finite automata. *Elektronische Informationsverarbeitung und Kybernetik*, 22(9):459–462, 1986.

[Lis06] V.A. Liskovets. Exact enumeration of acyclic deterministic automata. *Discrete Applied Mathematics*, 154(3):537–551, 2006.

[LPW09] David Asher Levin, Yuval Peres, and Elizabeth Lee Wilmer. *Markov chains and mixing times*. American Mathematical Soc., 2009.

[Nic00] C. Nicaud. *Etude du comportement en moyenne des automate finis et des langages rationnels*. PhD thesis, 2000.

[Nic14] Cyril Nicaud. Random deterministic automata. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014 Symposium, MFCS 2014*, volume 8634 of *Lecture Notes in Computer Science*, pages 5–23. Springer, 2014.

[Rob85] R. Robinson. *Counting Strongly Connected finite Automata*, pages 671–685. Graph theory with Applications to Algorithms and Computer Science. Wiley, 1985.

[STV01] Thomas Schwentick, Denis Thérien, and Heribert Vollmer. Partially-ordered two-way automata: A new characterization of DA. In *Developments in Language Theory*, volume 2295 of *Lecture Notes in Computer Science*, pages 239–250. Springer, 2001.

[Vys59] V. Vyssotsky. A counting problem for finite automata. Technical report, Bell Telephon Laboratories, 1959.