Enumeration and random generation of possibly incomplete deterministic automata.

Frédérique Bassino, Julien David, Cyril Nicaud

Enumeration and random generation of possibly incomplete deterministic automata

Frédérique Bassino *
LIPN UMR 7030, Université Paris 13 - CNRS
99, avenue J.-B. Clément. 93430 Villetaneuse – France
email: Frederique.Bassino@lipn.univ-paris13.fr

Julien David * and Cyril Nicaud *
LIGM UMR 8049, Université Paris Est,
77454 Marne-la-Vallée Cedex 2 – France
email: {Julien.David,nicaud}@univ-mlv.fr

Abstract

This paper presents an efficient random generator, based on a Boltzmann sampler, for accessible, deterministic and possibly not complete automata. An interesting intermediate result is that for any finite alphabet, the proportion of complete automata with n states amongst deterministic and accessible ones is greater than a positive constant.

1 Introduction

The enumeration of finite automata according to various criteria (non-isomorphic [11], up to permutation of the labels of the edges [11], with a strongly connected underlying graph [13, 14, 15, 19], accessible [13, 15, 19], acyclic [16],...) is a problem that was studied since 1959 [21].

In [1] the first and third authors exhibit a bijection between the set $\mathcal{A}_n$ of deterministic, complete and accessible automata with n states over a k-letter alphabet and some diagrams, which can themselves be represented as partitions of the set $\{1, \cdots, kn\}$ into n nonempty subsets. These combinatorial transformations show that the order of magnitude of the cardinality $|\mathcal{A}_n|$ of the set $\mathcal{A}_n$ is related to the Stirling numbers of the second kind that can be be used to reformulate an asymptotic estimate of $|\mathcal{A}_n|$ due to Korshunov [13]. They also provide a uniform random generator for the automata of $\mathcal{A}_n$, based on Boltzmann samplers [6, 7], that is more efficient than former ones [4, 17] using a recursive algorithm [9, 18].

This paper generalizes the study [1] of deterministic, complete and accessible automata to possibly incomplete automata. It simplifies the algorithmic part of [1]. More

*The authors were supported by the ANR (GAMMA - project BLAN07-2_195422).

precisely the combinatorial transformations are changed, a unique bijection allows us to directly transform specific set partitions into accessible and deterministic automata. A careful analysis of the complexity is done to ensure that the generator obtained is still efficient; as in the case of complete automata, its average complexity is $\mathcal{O}(n^{3/2})$, where n is the number of states of automata. An interesting intermediate result is that for any finite alphabet, the proportion of complete automata with n states amongst deterministic and accessible ones is greater than a positive constant.

The paper is organized as follows. Bijections used to change complete automata into set partitions are presented in Section 3. The ones used to transform possibly incomplete automata are given in Section 4. Section 5 is devoted to enumeration results used in the analysis of the complexity of the generator. The random generator, together with the analysis of its efficiency, is given in Section 6. The paper closes with some experimental results obtained with the `C++` library `REGAL`[1] [2].

2 Definitions and basic properties

Our goal is to study from a combinatorial point of view the set of accessible and deterministic automata with n states. Therefore we first recall some definitions about finite automata, referring the readers to [12, 20] for basic elements of this theory. We also introduce boxed diagrams and specific set partitions that will be used to enumerate and generate automata.

2.1 Deterministic and accessible automata

A *deterministic finite automaton* $\mathcal{A} = (A, Q, \cdot, q_0, F)$ over a finite alphabet A is a quintuple where Q is a finite set of *states*, $q_0 \in Q$ is the initial state, $F \subset Q$ is the set of final states and the *transition function* $\cdot$ is an element of $Q \times A \mapsto Q \cup \emptyset$. If $p \cdot a = \emptyset$ for a given state $p \in Q$ and a letter $a \in A$, then $p \cdot a$ is an *undefined transition*. A deterministic finite automaton without undefined transition is *complete*. If $\mathcal{A} = (A, Q, \cdot, q_0, F)$ is a deterministic finite automaton, its transition function is extended by morphism to $Q \times A^*$ making use of the convention $\emptyset \cdot a = \emptyset$ for every $a \in A$.

A deterministic finite automaton $\mathcal{A}$ is *accessible* when for each state q of $\mathcal{A}$, there exists a word $u \in A^*$ such that $q_0 \cdot u = q$.

Two deterministic finite automata $\mathcal{A} = (A, Q, \cdot, q_0, F)$ and $\mathcal{A}' = (A, Q', \cdot, q'_0, F')$ over the same alphabet are *isomorphic* when there exists a bijection τ from $Q \cup \emptyset$ to $Q' \cup \emptyset$ such that, $\tau(q_0) = q'_0$, $\tau(\emptyset) = \emptyset$, $\tau(F) = F'$ and for each $(q, \alpha) \in Q \times A$, $\tau(q \cdot \alpha) = \tau(q) \cdot \alpha$. Two isomorphic automata only differ by the labels of their states.

2.2 Transition structures

Now we introduce a representation of accessible and deterministic automata that uses the minimal labels of simple paths and allows us to enumerate and generate them easily.

[1] available at: `http://regal.univ-mlv.fr/`

More precisely a *simple path* in a deterministic automaton $\mathcal{A}$ is a path labelled by a word u, such that all prefixes v and v' of u, with $v \neq v'$, satisfy $q_0 \cdot v \neq q_0 \cdot v'$. In other words, in the graphical representation of $\mathcal{A}$ the path labelled by u does not go twice through the same state. Let $\mathcal{A}$ be an accessible and deterministic automaton over the alphabet A and let w be the map from Q to A^* defined for every state $q \in Q$ by

$$w(q) = \min_{lex}\{u \in A^* \mid q_0 \cdot u = q \text{ and } u \text{ is a simple path in } \mathcal{A}\},$$

where the mininum is taken according to the lexicographic order. Note that $w(q)$ always exists since $\mathcal{A}$ is accessible. An automaton $\mathcal{A} = (A, Q, \cdot, q_0, F)$ is called a *base automaton* when $Q \subset A^*$ (the states are labelled by words) and for all $u \in Q$, $w(u) = u$. Note that by construction, if $u \in Q$ and v is a prefix of u, then $v \in Q$. As two distinct base automata cannot be isomorphic, we can directly work on isomorphism classes using base automata.

The *transition structure* of an automaton $\mathcal{A} = (A, Q, \cdot, q_0, F)$ is $\mathcal{D} = (A, Q, \cdot, q_0)$: in $\mathcal{D}$ there is no more distinguished final states. We can define similarly accessible and deterministic transition structures.

Denote by $\mathcal{D}_n$ the set of accessible and deterministic transition structures of base automata with n states, and by $\mathcal{C}_n$ the set of complete transition structures belonging to $\mathcal{D}_n$.

Given an element $\mathcal{D}$ of $\mathcal{D}_n$, there are exactly 2^n automata whose transition structure is $\mathcal{D}$, since the accessibility prevents distinct choices of final sets to form the same automaton. Therefore the number of deterministic and accessible automata, up to isomorphism, is $2^n|\mathcal{D}_n|$.

Note that forbidding or not the set of final states to be empty does not basically change the results, since the probability of this event is $1/2^n$.

Our purpose is to enumerate the elements in $\mathcal{D}_n$ and to generate them randomly for the uniform distribution on $\mathcal{D}_n$.

In the following we only consider accessible and deterministic automata, and complete and accessible deterministic transition structures. Consequently, these objects will often be called respectively *automata* or *transition structures*.

2.3 Boxed diagrams

Boxed diagrams were introduced in [1] to characterize transition structures with objects that are easier to enumerate.

A *diagram* of width m and height n is a sequence $(x_1, \ldots, x_m)$ of nondecreasing nonnegative integers such that $x_m = n$, classically represented as a diagram of boxes, see Figure 1. A *k-Dyck diagram* of size n is a diagram of width $(k-1)n+1$ and height n such that $x_i \geq \lceil i/(k-1) \rceil$ for each $i \leq (k-1)n$. A *boxed diagram* is a pair of sequences $((x_1, \ldots, x_m), (y_1, \ldots, y_m))$ where $(x_1, \ldots, x_m)$ is a diagram and for each $i \in [\![1..m]\!]$, the y_ith box of the column i of the diagram is marked, see Figure 1. As a consequence, a diagram gives rise to $\prod_{i=1}^{m} x_i$ boxed diagrams. A *k-Dyck boxed diagram* of size n is a boxed diagram whose first coordinate $(x_1, \ldots, x_{(k-1)n+1})$ is a k-Dyck diagram of size n.

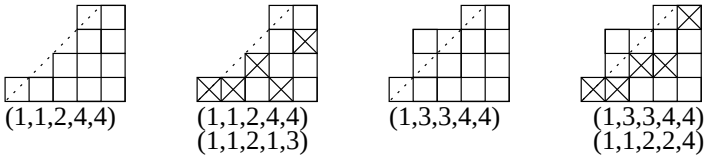

Figure 1: A diagram of width 5 and height 4, a boxed diagram, a 2-Dyck diagram and a 2-Dyck boxed diagram

As it will be recalled in Section 3, there exists a bijection between k-Dyck boxed diagrams and complete transition structures.

2.4 Set partitions

Denote by $\mathcal{P}_{n,m}$ the set of all set partitions of the set $\{1,\dots,n\}$ into m nonempty subsets. The cardinality of the set $\mathcal{P}_{n,m}$ is equal to $\{{n \atop m}\}$, the Stirling numbers of the second kind.

Let $P = \{P_1,\dots,P_m\}$ be a set partition in $\mathcal{P}_{n,m}$ and $\ell \in \{1,\dots,n\}$, the *ℓ-subpartition* of P, denoted by $P^{(\ell)}$, is the set partition of the set $\{1,\dots,\ell\}$ whose elements are the nonempty subsets $P_i \cap \{1,\dots,\ell\}$ where $i \in \{1,\dots,m\}$. Therefore the set partition $P^{(\ell)}$ contains at most m nonempty subsets.

Let $k \geq 2$ be an integer. A set partition $P = \{P_1,\dots,P_m\}$ of $\mathcal{P}_{n,m}$, where the P_i are sorted according to their smallest element, is said to be a *k-Dyck set partition* when it satisfies the k-Dyck condition: for every $j \in \{1,\dots,m\}$, the smallest integer in P_j is smaller than or equal to $k(j-1)+1$.

For example, for the set $\{1,\dots,13\}$ and $m = 4$, the set partition

$$P = \{\{1,11,13\},\{2,3,6,9\},\{4,8,10\},\{5,7,12\}\}$$

is a 3-Dyck set partition.

We show, in the sequel, that there exists a bijection between transition structures over a k-letter alphabet and k-Dyck set partitions (Theorem 2).

3 Complete automata

In this section we propose a new bijection that directly build a k-Dyck set partition from a complete transition structure. For the sake of completeness, we also detail the bijection between transition structures and k-Dyck boxed diagrams.

3.1 Complete transition structures and boxed diagrams

First recall the bijection established in [17] for a two-letter alphabet and generalized to any finite alphabet in [4]:

Theorem 1 *There exists a bijection between the set $\mathcal{C}_n$ of accessible, complete and deterministic transition structures with n states over a k-letter alphabet A and the set of k-Dyck boxed diagrams of size n. This transformation and its inverse can be computed in linear time.*

Proof: In the following we denote by ε the empty word. For $n \geq 1$, let $\mathcal{D} = (A, Q, \cdot, \varepsilon) \in \mathcal{C}_n$ be the transition structure of a deterministic, accessible and complete base automaton over a k-letter alphabet. Since $\mathcal{D}$ is complete, it contains kn transitions of the form (u, α), with $u \in Q$ and $\alpha \in A$. We partition the set of these transitions depending on whether they belong to the spanning tree induced by the depth-first traversal according to the lexicographical order of the structure or not. Using the properties of the labelling of the states of $\mathcal{D}$, the set partition can be described as follows, for any $u \in Q$:

- If $u\alpha \in Q$ then $u \cdot \alpha = u\alpha$ and (u, α) is a *tree transition*. It belongs to the spanning tree.
- If $u\alpha \notin Q$ then $u \cdot \alpha <_{lex} u\alpha$, and (u, α) is called a *missing transition*. It does not belong to the spanning tree.

There are $n - 1$ tree transitions and $(k - 1)n + 1$ missing transitions.

Let ν be the unique increasing bijection from the set Q (lexicographically ordered) to $\{1, \ldots, n\}$, that is, $\nu(q)$ is the number of elements of Q smaller or equal to q for the lexicographical order. To any missing transition $t = (q, \alpha)$ we associate the pair of integers (x_t, y_t) defined by

$$\begin{cases} x_t &= |\{u \in Q \mid u <_{lex} q\alpha\}| \\ y_t &= \nu(q \cdot \alpha). \end{cases}$$

Next we order the transitions of $\mathcal{D}$ according to the relation: $(u, \alpha) < (v, \beta)$ if and only if $u\alpha <_{lex} v\beta$. Then the bijection Ψ between $\mathcal{C}_n$ and the set of k-Dyck boxed diagrams of size n can be defined as follows: let $(t_1, \ldots, t_{(k-1)n+1})$ be the ordered sequence of missing transitions of $\mathcal{D}$,

$$\Psi(\mathcal{D}) = ((x_{t_1}, \ldots, x_{t_{(k-1)n+1}}), (y_{t_1}, \ldots, y_{t_{(k-1)n+1}})).$$

The map Ψ is a bijection (see [17, 4] for details). The sequence $(x_{t_1}, \ldots, x_{t_{(k-1)n+1}})$ represents the depth-first spanning tree of $\mathcal{D}$ and defines the labelling of the states of $\mathcal{D}$; the sequence $(y_{t_1}, \ldots, y_{t_{(k-1)n+1}})$ carries all the informations about missing transitions:

$$u \cdot \alpha = \nu^{-1}(y_{(u,\alpha)}).$$

This completes the proof. □

3.2 Complete transition structures and set partitions

We now introduce a bijection between complete transitions structures over a k-letter alphabet and k-Dyck set partitions. This new bijection allows us to simplify, from an

algorithmic point of view, the random generator of complete automata presented in [1]. Let $\mathcal{D} = (A, Q, \cdot, \varepsilon)$ be complete transition structure with n states over a k-letter ordered alphabet $A = \{a_1, \ldots, a_k\}$. We formally add the transition $(\emptyset, \varepsilon)$, which can be seen as the arrow indicating the initial state on the graphical representation of $\mathcal{D}$. By convention $\emptyset \cdot \varepsilon = \varepsilon$, the initial state of $\mathcal{D}$. Let $\mathcal{T}_\mathcal{D}$ be the set of transitions of $\mathcal{D}$, including the new one. By construction $|\mathcal{T}_\mathcal{D}| = kn + 1$. To each transition $t = (u, a)$ of $\mathcal{T}_\mathcal{D}$ is uniquely associated an integer $n(t)$ of $\{1, \ldots, kn + 1\}$ making use of the depth-first traversal with respect to the lexicographical order of $\mathcal{D}$: $n((\emptyset, \varepsilon)) = 1$ and for every $t = (u, a)$ and $t' = (u', a')$ in $\mathcal{T}_\mathcal{D} \setminus \{(\emptyset, \varepsilon)\}$, $n(t) < n(t')$ if and only if $ua <_{lex} u'a'$.

Let $P_\mathcal{D}$ be the set partition of $\mathcal{P}_{kn+1,n}$ such that for any i and j of $\{1, \ldots, kn + 1\}$, i and j are in the same part of $\mathcal{P}_\mathcal{D}$ if and only if $u \cdot a = u' \cdot a'$, where $n((u, a)) = i$ and $n((u', a')) = j$. In other words, i and j are in the same part when they are the numbers associated to two transitions going into the same state of $\mathcal{D}$.

We denote by χ the map from the set of complete transition structures with n states to $\mathcal{P}_{kn+1,n}$ where $\chi(\mathcal{D}) = P_\mathcal{D}$ is the partition defined above.

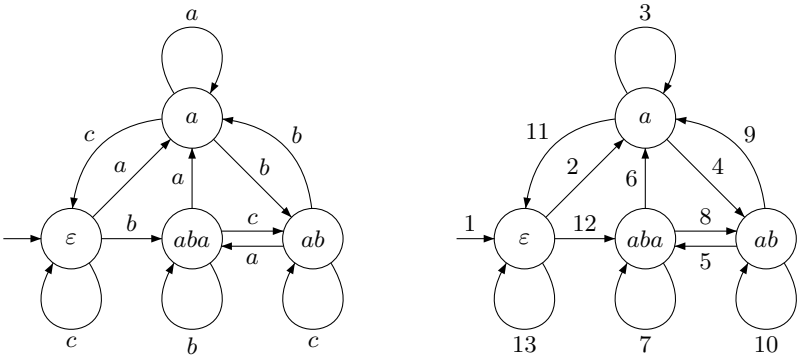

Figure 2: On the left side: a transition structure $\mathcal{D}$. On the right side: the numbered transitions. The set partition $P_\mathcal{D} = \{\{1, 11, 13\}, \{2, 3, 6, 9\}, \{4, 8, 10\}, \{5, 7, 12\}\}$ is obtained by grouping together the edges according to their ending state.

Theorem 2 *For any $n \geq 1$ and $k \geq 2$, the map χ that transforms $\mathcal{D}$ into $P_\mathcal{D}$ is a bijection from the set of transition structures of complete base automata with n states over a k-letter alphabet onto the set of k-Dyck set partitions of $\mathcal{P}_{kn+1,n}$.*

Proof: A construction to transform a set partition $P = \{P_1, \ldots, P_n\}$ of $\mathcal{P}_{kn+1,n}$ into a boxed diagram of width $kn + 1$ and heigth n is given in [1]. If we assume that the sets P_i's are sorted according to their smallest element, the boxed diagram $B = ((x_1, \ldots, x_{kn+1}), (y_1, \ldots, y_{kn+1}))$ is such that, for any $i \in \{1, \ldots, kn + 1\}$, x_i is equal to the number of subsets in the subpartition $P^{(i)}$, and y_i is the index of the subset of P to which belongs i. A k-Dyck boxed diagram is then obtained by removing from B the columns corresponding to the smallest element of each part of P (we refer the reader to [1] for more details and related algorithms). As this construction is a bijection from

the set of k-Dyck set partitions of $\mathcal{P}_{kn+1,n}$ onto the k-Dyck boxed diagrams of size n, making use of Theorem 1 we conclude that the two sets considered in the statement of Theorem 2 have the same cardinality.

Now we prove that χ is an injection. Suppose that $\mathcal{D} = (A, Q, \cdot, \varepsilon)$ and $\mathcal{D}' = (A, Q', *, \varepsilon)$ are the transition structures of two distinct accessible, deterministic and complete base automata with n states. Let $e = (p, a)$ be the first transition in depth-first order in both $\mathcal{D}$ and $\mathcal{D}'$ such that $p \cdot a \neq p * a$. Note that the states whose labels are smaller than or equal to p for the lexicoraphic order are exacly the same in $\mathcal{D}$ and $\mathcal{D}'$. Let $q = p \cdot a$ and $q' = p * a$. Assume by symmetry that q is strictly smaller than q' for the lexicographic order. Therefore q is smaller than or equal to p, and hence it belongs to Q and Q'. Consequently, in $P_{\mathcal{D}}$ $n(e)$ is in the same subset as the first edge ending in q, and this is not true in $P'_{\mathcal{D}}$. Thus χ is an injection.

To conclude, we prove that the image by χ of a complete transition structure with n states over a k-letter alphabet is a k-Dyck set partition of $\mathcal{P}_{kn+1,n}$. Suppose that there exist a complete transition structure $\mathcal{D}$ of a base automaton for which it is not true. Let $P_{\mathcal{D}} = \{P_1, \ldots, P_n\}$, such that the P_i's are sorted according to their smallest element. Let ℓ be the smallest nonnegative integer in $\{2, \ldots, m\}$ such that the smallest element of P_ℓ is strictly greater than $k(\ell - 1) + 1$. Therefore, the $k(\ell - 1)$ first transitions in $\mathcal{D}$ end in a state of number in $\{1, \ldots, \ell - 1\}$, forming a complete transition structure with $\ell - 1$ states. Hence $\mathcal{D}$ is not accessible which is a contradiction. Thus χ is also a surjection onto the subset of $\mathcal{P}_{kn+1,n}$ made of k-Dyck set partitions. □

We give in Section 6.2 an algorithm that transform a k-Dyck set partition of $\mathcal{P}_{kn+1,n}$ into a transition structure with n states over a k-letter alphabet.

4 Possibly incomplete transition structures

In this section we present two combinatorial transformations of possibly incomplete transition structures. The first one is a non-classical way to obtain complete transition structure. The second one links possibly incomplete transition structures with a subsclass of k-Dyck boxed diagrams and is used to enumerate these structures in Section 5.

4.1 From transition structures to complete transitions structures

In theory of automata an incomplete automaton is classically changed into a complete one recognizing the same language by the addition of a sink state. This transformation is not suitable for our combinatorial construction. Indeed if two incomplete automata have the same depth-first spanning tree, they may not have the same one after the addition of a sink state, as shown on Fig. 3.

Therefore we introduce another transformation denoted by ϕ and defined as follows: to any $\mathcal{D} \in \mathcal{D}_n$, with $\mathcal{D} = (A, Q, \cdot, \varepsilon)$, we associate with the complete transition structure $\phi(\mathcal{D}) = (A, Q', *, \varepsilon)$ in $\mathcal{C}_{n+1}$ with $Q' = \{\varepsilon\} \cup a_k Q$ where $a_k = \max_{lex}\{\alpha \in A\}$ and whose

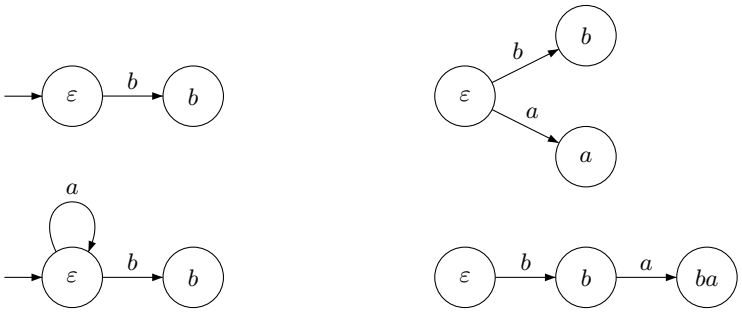

Figure 3: On the left two automata with the same spanning tree. On the right, their spanning trees once completed by adding a sink state.

transitions are defined by:

$$\begin{cases} \varepsilon * \alpha = \varepsilon & \text{if } \alpha \neq a_k \\ \varepsilon * a_k = a_k & \\ q' * \alpha = a_k(q \cdot \alpha) & \text{if } \exists q \in A^*,\ q' = a_k q \text{ and } q \cdot \alpha \neq \emptyset \\ q' * \alpha = \varepsilon & \text{if } \exists q \in A^*,\ q' = a_k q \text{ and } q \cdot \alpha = \emptyset. \end{cases}$$

This construction consists of

- adding a new state, that becomes the initial state ε of $\phi(\mathcal{D})$ and a transition $\varepsilon * a_k = q_0$, labelled by the greatest letter and where q_0 is the initial state $\mathcal{D}$,
- relabelling the transition structure to obtain the transition structure of a base automaton,
- changing any undefined transition $q \cdot \alpha = \emptyset$ into $q * \alpha = \varepsilon$.

Note that ϕ does not preserve the language recognized.

Lemma 1 *Denote by $\mathcal{E}_n$ the subset of transition structures of $\mathcal{C}_n$ such that $\varepsilon \cdot \alpha = \varepsilon$ for $\alpha \in A \setminus \{\max_{lex}\{\alpha \in A\}\}$. The function ϕ is a bijection from $\mathcal{D}_n$ to $\mathcal{E}_{n+1}$.*

By definition of ϕ, $\mathcal{E}_{n+1} = \phi(\mathcal{D}_n)$. Moreover the inverse of ϕ is obtained by removing the initial state, making the state a_k initial, and relabelling the states.

4.2 The k-Dyck boxed diagrams associated with the elements of $\mathcal{E}_n$

Recall that if $(t_1, \ldots, t_{(k-1)n+1})$ is the ordered sequence of missing transitions of the base transition structure $\mathcal{D}$ with n states over a k-letter alphabet,

$$\Psi(\mathcal{D}) = ((x_{t_1}, \ldots, x_{t_{(k-1)n+1}}), (y_{t_1}, \ldots, y_{t_{(k-1)n+1}})),$$

where for any missing transition $t = (q, \alpha)$

$$\begin{cases} x_t & = |\{u \in Q \mid u <_{lex} q\alpha\}| \\ y_t & = \nu(q \cdot \alpha) \end{cases}$$

$\nu(q)$ being the number of states of $\mathcal{D}$ smaller or equal to q for the lexicographical order.

For $n \geq 2$, the image $\Psi(\mathcal{E}_n)$ is easy to characterize.

Lemma 2 *Let $\mathcal{F}_n$ be the set of k-Dyck boxed diagrams of size n such that for all $i \in \{1, \ldots, k-1\}$, $x_i = 1$ and $y_i = 1$. For any $n \geq 2$, Ψ is a bijection between $\mathcal{E}_n$ and $\mathcal{F}_n$.*

Proof: Let $A = \{a_1 < \ldots < a_k\}$. If $\mathcal{D} = (A, Q, \cdot, \varepsilon) \in \mathcal{E}_n$ then for $i \in \{1, \ldots, k-1\}$, $\varepsilon \cdot a_i = \varepsilon$. Moreover, ε is the only word of Q that does not start with a_k. Thus, the first $k-1$ missing transitions of $\mathcal{D}$ are (ε, a_1), ..., (ε, a_{k-1}). Therefore for any $i \in \{1, \ldots, k-1\}$, $\{u \in Q \mid u <_{lex} a_i\} = \{\varepsilon\}$ and $x_{(\varepsilon,a_i)} = 1$. Moreover since $1 \leq y_{(\varepsilon,a_i)} \leq x_{(\varepsilon,a_i)}$, $y_{(\varepsilon,a_i)} = 1$.

If $\mathcal{D} = (A, Q, \cdot, \varepsilon) \notin \mathcal{E}_n$, let i be the smallest integer such that $\varepsilon \cdot a_i \neq \varepsilon$. Then the word a_i is the smallest word in $Q \setminus \{\varepsilon\}$. If a missing transition (u, α) is such that $u\alpha <_{lex} a_i$, then $u = \varepsilon$ and $\alpha < a_i$: there are exactly $i-1$ such missing transitions. Hence, the i-th missing transition t_i, in the ordered sequence, is such that $x_{t_i} \geq 2$ and $\Psi(\mathcal{D}) \notin \mathcal{F}_n$, concluding the proof. □

5 Enumeration

This section is devoted to enumeration problems. The number of accessible automata is related to the Stirling numbers of the second kind whose definition and asymptotic estimate are recalled.

5.1 The Stirling numbers of the second kind

The Stirling number of the second kind $\left\{ {n \atop m} \right\}$, where n and m are two nonnegative integers, is the number of set partitions of a set with n elements into m nonempty subsets.

Lemma 3 ([1]) *The number of boxed diagrams of width m and height n is equal to $\left\{ {m+n \atop n} \right\}$.*

Recall that the LambertW-function [3] is the inverse of the function $x \mapsto xe^x$. Its principal branch W_0 is real-valuted for x in $[-e^{-1}, +\infty[$ and is the unique branch which is analytic at zero. Its series expansion is

$$W_0(z) = \sum_{n=1}^{\infty} \frac{(-n)^{n-1}}{n!} z^n = z - z^2 + \mathcal{O}\left(z^3\right) \tag{1}$$

The Stirling numbers of the second kind are asymptotically estimated with the saddle point method.

Theorem 3 (Good [10]) *When n and m both tend towards infinity with $n = \Theta(m)$, the following result holds:*

$$\left\{ {n \atop m} \right\} \sim \frac{n!(e^\rho - 1)^m}{m!\rho^n \sqrt{2\pi n(1 - \frac{n}{m}e^{-\rho})}}$$

where $\rho = W_0(-\frac{n}{m}e^{-\frac{n}{m}}) + \frac{n}{m}$ is the unique positive root of the equation $m\rho = n(1 - e^{-\rho})$.

5.2 Enumeration of accessible deterministic automata

Recall that the number of automata in a specific class is equal to the number of transition structures of the same class multiplied by 2^n.

Complete automata

Korshunov [14] gave an asymptotic equivalent of the cardinality $|\mathcal{C}_n|$ of the set of complete, deterministic and accessible transition structures with n states over a k-letter alphabet. This equivalent can be reformulated [1] in terms of the Stirling numbers of the second kind:

Theorem 4 (Korshunov [13, 14]) *The number $|\mathcal{C}_n|$ of accessible complete and deterministic transition structures with n states over a k-letter alphabet satisfies*

$$|\mathcal{C}_n| \sim E_k \, n \left\{ {kn \atop n} \right\} \quad \textit{where} \quad E_k = \frac{1 + \sum_{r=1}^{\infty} \frac{1}{r} \binom{kr}{r-1} \left(e^{k-1}\beta_k\right)^{-r}}{1 + \sum_{r=1}^{\infty} \binom{kr}{r} \left(e^{k-1}\beta_k\right)^{-r}}, \quad \beta_k = \frac{(k\zeta_k)^k}{e^{k-1}(e^{\zeta_k} - 1)}$$

and ζ_k is the positive root of $\rho = k(1 - e^{-\rho})$.

Possibly incomplete automata

To enumerate possibly incomplete transition structures we basically use the following lemma:

Lemma 4 *For any fixed $k \geq 2$, as n tends toward infinity, one has*

$$\left\{ {kn+1 \atop n+1} \right\} \sim e^{\zeta_k} \left\{ {kn \atop n} \right\} \quad \textit{with} \quad \zeta_k = W_0(-ke^{-k}) + k.$$

Proof: The following proof is based on the comparison of the estimations of $\left\{ {kn \atop n} \right\}$ and $\left\{ {kn+1 \atop n+1} \right\}$ obtained with Theorem 3.

In the case of $\left\{ {kn \atop n} \right\}$, $\zeta_k = W_0(-ke^{-k}) + k$ is the positive root of $\rho = k(1 - e^{-\rho})$. Theorem 3 and Stirling's formula give (see [1] for details):

$$\left\{ {kn \atop n} \right\} \sim \frac{(kn)!}{n!} \frac{(e^{\zeta_k} - 1)^n}{\zeta_k^{kn} \sqrt{2\pi kn(1 - ke^{-\zeta_k})}} \sim \alpha_k \beta_k^n n^{(k-1)n - 1/2}$$

with $\alpha_k = (2\pi(\zeta_k - (k-1)))^{-\frac{1}{2}}$ and $\beta_k = \frac{(k\zeta_k)^k}{e^{k-1}(e^{\zeta_k} - 1)}$.

Denote by f the function $f(x) = W_0(-xe^{-x}) + x$. To use Theorem 3 for $\left\{ {kn+1 \atop n+1} \right\}$ we have to compute $\rho_{n,k} = f\left(\frac{kn+1}{n+1}\right) = f(k - \frac{k-1}{n+1})$. Because of the analycity of f, we can use Taylor expansion:

$$\rho_{n,k} = f\left(k - \frac{k-1}{n+1}\right) = f(k) - \frac{k-1}{n+1}f'(k) + \mathcal{O}\left(\frac{1}{n^2}\right) = \zeta_k - (k-1)f'(k)\frac{1}{n} + \mathcal{O}\left(\frac{1}{n^2}\right).$$

From Theorem 3 we get:

$$\left\{ {kn+1 \atop n+1} \right\} \sim \frac{(kn+1)!(e^{\rho_{n,k}} - 1)^{n+1}}{(n+1)!\rho_{n,k}^{kn+1}\sqrt{2\pi(kn+1)(1 - \frac{kn+1}{n+1}e^{-\rho_{n,k}})}}.$$

Usual estimations and Stirling's formula lead to:

$$\begin{aligned}
\frac{(kn+1)!}{(n+1)!} &\sim e^{-(k-1)n}k^{3/2}k^{kn}n^{(k-1)n} \\
\sqrt{2\pi(kn+1)(1 - \frac{kn+1}{n+1}e^{-\rho_{n,k}})} &\sim \sqrt{2\pi kn(1 - ke^{-\zeta_k})} \\
(e^{\rho_{n,k}} - 1)^{n+1} &\sim (e^{\zeta_k} - 1)^{n+1}e^{-\frac{(k-1)f'(k)e^{\zeta_k}}{e^{\zeta_k}-1}} \\
\rho_{n,k}^{kn+1} &\sim \zeta_k^{kn+1}e^{-\frac{k(k-1)f'(k)}{\zeta_k}}
\end{aligned}$$

Moreover as ζ_k satisfies $\zeta_k = k(1 - e^{-\zeta_k})$, $e^{\zeta_k}/(e^{\zeta_k} - 1) = k/\zeta_k$. Finally we obtain

$$\left\{ {kn+1 \atop n+1} \right\} \sim \alpha'_k \beta_k^n n^{(k-1)n-1/2}$$

with $\alpha'_k = \frac{e^{\zeta_k}}{\sqrt{2\pi(\zeta_k-(k-1))}}$. Thus $\left\{ {kn+1 \atop n+1} \right\} \sim e^{\zeta_k} \left\{ {kn \atop n} \right\}$, concluding the proof. □

Theorem 5 shows that there are $\Theta(n\,2^n \left\{ {kn \atop n} \right\})$ accessible and deterministic base automata with n states over a k-letter alphabet.

Theorem 5 *The number $|\mathcal{D}_n|$ of accessible and deterministic transition structures of base automata with n states is $\Theta(n \left\{ {kn \atop n} \right\})$.*

Proof: First, as $\mathcal{C}_n \subset \mathcal{D}_n$, $|\mathcal{C}_n| \leq |\mathcal{D}_n|$. And Theorem 4 leads to the lower bound.

In Section 3 we exhibited a bijection in two steps between the set $\mathcal{D}_n$ and the set $\mathcal{F}_{n+1}$ of k-Dyck boxed diagrams $((x_1, \ldots, x_{(k-1)(n+1)+1}), (y_1, \ldots, y_{(k-1)(n+1)+1}))$ such that for all $i \in \{1, \ldots, k-1\}$, $x_i = 1$ and $y_i = 1$.

Now the number of elements in $\mathcal{F}_{n+1}$ is smaller than the number of boxed diagrams of width $(k-1)(n+1)+1 = (k-1)n + k$ and height $n+1$, whose $k-1$ first columns have height 1, and the last column has height $n+1$. Note that it is an overestimation of $|\mathcal{F}_{n+1}|$ since diagrams that do not satisfy the diagonal condition are taken into account. Therefore the elements of $\mathcal{F}_{n+1}$ are approximated by boxed diagrams made of $k-1$

columns of height 1, a boxed diagram of width $(k-1)n$ and height $n+1$ and a column of height $n+1$. There are $n+1$ possibilities for the last column. Thus, by Lemma 3, we obtain that $|\mathcal{D}_n| \leq (n+1)\left\{ {kn+1 \atop n+1} \right\}$. We conclude using Lemma 4. □

Corollary 1 *As n tends towards infinity, $|\mathcal{C}_n| = \Theta(|\mathcal{D}_n|)$.*

6 Random generation

In this section, in order to uniformly generate deterministic and accessible automata, we adapt an algorithm described in [1] and used to generate complete automata.

The first step of the algorithm is based on a Boltzmann sampler that generates specific set partitions. The second one consists of the transformation of these set partitions into accessible and deterministic automata.

6.1 A Boltzmann sampler to generate random set partitions

The Boltzmann sampler used here is a direct application of the work of Duchon, Flajolet, Louchard and Schaeffer [6]. Boltzmann samplers do not generate fixed size objects. They depend on a real parameter $x > 0$ and, for any given integer n, the value of x can be chosen so that the average size of the generated elements is n. The size is not fixed, but Boltzmann samplers guarantee that two elements of the same size have the same probability to be generated.

In order to uniformly generate set partitions of a set with $kn+1$ elements into $n+1$ nonempty subsets, we first consider the set of partitions of a set into $n+1$ nonempty sets. Its exponential generating function is $P_{n+1}(z) = \frac{(e^z-1)^{n+1}}{(n+1)!}$. Using Boltzmann sampler construction, each of the $n+1$ sets are generated assuming that its size follows a Poisson law $\text{Pois}_{\geq 1}$ of parameter x (a truncated Poisson variable K, where K is conditionned to be ≥ 1). The average size of the partition is then:

$$\mathbb{E}_x(\text{size of the partition}) = x\frac{P'_{n+1}(x)}{P_{n+1}(x)} = (n+1)x\frac{e^x}{e^x-1}.$$

Since we want a partition of a set having $kn+1$ elements, the value of the parameter x_n is chosen so that

$$(n+1)x_n\frac{e^{x_n}}{e^{x_n}-1} = kn+1,$$

that is, $x_n = \rho_{n,k}$ (see the proof of Lemma 4). When the Boltzmann parameter x_n is equal to $\rho_{n,k}$, the probability for a random set partition to be of size $kn+1$ is [6]:

$$\mathbb{P}_{\rho_{n,k}}(N = nk+1) = \frac{\rho_{n,k}^{kn+1}[z^{kn+1}]P_{n+1}(z)}{P_{n+1}(\rho_{n,k})} = \frac{\left\{ {kn+1 \atop n+1} \right\}\rho_{n,k}^{kn+1}}{(kn+1)!}\frac{(n+1)!}{(e^{\rho_{n,k}}-1)^{n+1}}.$$

This quantity can be asymptotically estimated using the same method as in the proof of Lemma 4:

$$\mathbb{P}_{\rho_{n,k}}(N = nk+1) \sim \frac{\alpha_k}{\sqrt{kn}}. \tag{2}$$

6.2 Random generator of possibly incomplete automata

The algorithm below is an improvement of two algorithms presented in [1]: starting from a partition, instead of computing its associated boxed diagram and then its transition structure, we directly compute the associated transition structure. In the algorithm, we assume that the parts $P = \{P_1, \ldots, P_n\}$ of the input are sorted according to their smallest element. The algorithm is based on the fact that the numbers in the elements of P correspond to the transitions of the result in depth-first order, and that, when $i \in P_j$, the i-th transition ends in the state j.

Algorithm 1: PartitionToTransitionStructure(P)

input: A k-Dyck partition $P = \{P_1, \ldots, P_n\}$ of $\mathcal{P}_{kn+1,n}$
$S =$ *empty stack*
Create the initial state $q_0 = 1$
$newState = 2$ *// this is the number of the next created state*
forall $a \in A$ *in reverse lexicographical order* **do**
  Push (q_0, a) into S
end
for $i \in \{2, .., kn + 1\}$ **do**
  $(p, a) =$ Pop from S
  $q = j$, where $i \in P_j$
  if $q = newSate$ **then**
    Create the state q
    $newState = newState + 1$
    forall $b \in A$ *in reverse lexicographical order* **do**
      Push (q, b) into S
    end
  end
  Add a transition from p to q labelled by a
end

The following algorithm use `PartitionToTransitionStructure(`P`)` to generate uniformly an accessible and deterministic automaton with n states over a k-letter alphabet. We use the fact that a k-Dyck partition $P = \{P_1, \ldots, P_{n+1}\}$ of $\mathcal{P}_{k(n+1)+1,n+1}$ corresponds to an element of $\mathcal{F}_{n+1}$ if and only if

$$\text{for all } i \in \{1, \ldots, k\},\ i \in P_1. \tag{3}$$

This property follows directly from the fact that the k first transitions in depth-first order of an element of $\mathcal{F}_{n+1}$, taking into account the edge coming into the inital state as in Section 3.2, end in the initial state. Similarly to the construction of [1] but with the additional constraint of Equation (3), `RandomTransitionStructure(`n`)`:

- generates uniformly a partition of $\{1, k+1, k+2, \ldots, k(n+1)\}$, a set with $kn+1$ elements, into $n+1$ nonempty subsets,

- adds the integers from $\{2, \ldots, k\}$ in the first part,
- adds $k(n+1)+1$ uniformly at random in one of the $n+1$ subsets.

Algorithm 2: RandomTransitionStructure(n)

input: The size n of the automaton

1 Compute $\rho_{n,k}$ using Equation (1) p.9.
2 **repeat**
3 **repeat**
4 **for** $i \in \{1, \ldots, n+1\}$ **do** $\lambda_i = \text{Pois}_{\geq 1}(\rho_{n,k})$
5 **until** $\lambda_1 + \cdots + \lambda_{n+1} = kn + 1$
6 $P = \{\{1, \ldots, \lambda_1\}, \{\lambda_1 + 1, \ldots, \lambda_1 + \lambda_2 + 1\}, \ldots, \{\lambda_1 + \ldots \lambda_{n+1}, \ldots, kn+1\}$
7 σ =RandomBijection($\{1, \ldots, kn+1\} \mapsto \{1, k+1, \ldots, k(n+1)\}$)
8 Change each $i \in \{1, \ldots, kn+1\}$ in P into $\sigma(i)$
9 **for** $i \in \{2 \ldots k\}$ **do** Add i in P_1
10 Choose uniformly $j \in \{1, \ldots, n+1\}$ and add $k(n+1)+1$ in P_j
11 **until** *P is a k-Dyck partition*
12 $\mathcal{A}'$ =PartitionToTransitionStructure(P)
13 Compute $\mathcal{A}$ obtain from $\mathcal{A}'$ by removing the initial state, the transitions coming in or out the initial set, and making the second state initial.
14 **for** *q state of $\mathcal{A}$* **do** Choose uniformly whether q is final or not

Note that for fixed n and k, Step 1 can be performed only once.

Theorem 6 *The average complexity of this random generator is $\mathcal{O}(n^{3/2})$.*

Proof: All steps can be performed in linear time, if we do not take into account the rejects. In a rejection algorithm, if a test is positive with probability p, the average number of rejects is $1/p$. Therefore, as a consequence of Theorem 5 and Lemma 4, the average number of rejects at Step 5 is bounded by a constant. Moreover Equation (2) p. 12 shows that, in average, there are $\mathcal{O}(\sqrt{n})$ rejects at Step 11. □

6.3 Experimental results

The random generator has been implemented in `REGAL`[2] [2], a `C++` library to generate random automata. We made some tests using this library mostly to compare deterministic and accessible automata with complete ones. For each test $10,000$ automata with $10,000$ states have been generated:

- For $k = 2$, 80.1% of automata are not complete. For $k = 3$, this proportion raises to 94.1%. Note that Lemma 4 shows that proportions are similar before the rejection step.
- For $k = 2$, 95.4% of automata are minimal for $k = 2$.

[2]available at: `http://regal.univ-mlv.fr/`

- For $k = 2$, 79.0% are strongly connected, it is about the same as for complete automata.
- For $k = 2$, in average an automaton has about 1.6 undefined transitions.

We implemented `PartitionToTransitionStructure(`P`)` in a new version of `REGAL` and did some benchmarks to compare with older versions, based on two transformations, using boxed diagrams. For automata with more than 500 states, the new algorithm is significantly faster.

n	100	500	1000	5000	10000
$k = 2$	0.0014	0.011	0.035	0.38	1.25
$k = 3$	0.0018	0.015	0.045	0.52	1.3
$k = 4$	0.0022	0.018	0.055	0.60	1.6

Figure 4: Average time in second for the generation of a possibly incomplete deterministic accessible automaton with n states over a k-letter alphabet.

Fig. 4 gives the average times required for the generation of an automaton. Benchmarks were made with an Intel(R) Core(TM)2 CPU 6600 2.40GHz, with 2Go of RAM. We used 10 000 automata of each size.

Open problem

Our algorithm is in $\mathcal{O}(n^{3/2})$ because of the rejects in the construction of a random partition of $\{1, \ldots, kn + 1\}$ into $n + 1$ parts. The natural open problem to improve this algorithm is therefore to find a linear algorithm, if it exists, to randomly and uniformly generate such partitions.

References

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

[2] F. Bassino, J. David, C. Nicaud, REGAL: a library to randomly and exhaustively generate automata. In *12th International Conference on Implementation and Application of Automata (CIAA'07)*, Lect. Notes Comput. Sci. vol. 4783 (2007), 303–305, Springer-Verlag.

[3] R. Corless, G. Gonnet, D. Hare, D. Jeffrey, D. Knuth, On the Lambert W-function, *Adv. in Comput. Math.*, 5 (1996), 329–359.

[4] J.-M. Champarnaud, T. Paranthoën, Random generation of DFAs, *Theoret. Comput. Sci.*, 330 (2005), 221–235.

[5] M. Domaratzki, D. Kisman, J. Shallit, On the number of distinct languages accepted by finite automata with n states, *J. Autom. Lang. Comb.*, no. 4 (2002), 469–486.

[6] P. Duchon, P. Flajolet, G. Louchard, G. Schaeffer, Boltzmann Samplers for the Random Generation of Combinatorial Structures, *Combinatorics, Probability, and Computing*, 13 (2004), 577–625.

[7] P. Flajolet, E. Fusy, C. Pivoteau, *Boltzmann Sampling of Unlabelled Structures*, Proceedings of Analytic Combinatorics and Algorithms Conference 2007 (ANALCO'07), SIAM Press.

[8] P. Flajolet, R. Sedgewick, *Analytic combinatorics*, Cambridge University Press, 2009.

[9] P. Flajolet, P. Zimmermann, B. Van Cutsem, A calculus of random generation of labelled combinatorial structures, *Theoret. Comput. Sci.*, 132 (1994), no. 1-2, 1–35.

[10] I. Good, An asymptotic formula for the differences of the powers at zero, *Ann. Math. Statist.*, 32 (1961), 249–256.

[11] M. A. Harrison, A census of finite automata, *Canad. J. Math.*, 17 (1965), 100–113.

[12] J. E. Hopcroft, J. Ullman, *Introduction to automata theory, languages, and computation*, Addison-Wesley, N. Reading, MA, 1980.

[13] D. Korshunov, Enumeration of finite automata, *Problemy Kibernetiki*, 34 (1978), 5–82, In Russian.

[14] A. D. Korshunov, On the number of non-isomorphic strongly connected finite automata, *Journal of Information Processing and Cybernetics*, 9 (1986), 459–462.

[15] V. Liskovets, The number of connected initial automata, *Kibernetika*, 5 (1969), 16–19, In Russian.

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

[17] C. Nicaud, *Étude du comportement en moyenne des automates finis et des langages rationnels*, Ph.D. thesis, Université Paris 7, 2000.

[18] A. Nijenhuis, H. S. Wilf, *Combinatorial Algorithms*, 2nd ed., Academic Press, 1978.

[19] R. Robinson, Counting strongly connected finite automata, In *Graph theory with Applications to Algorithms and Computer Science*, Y. Alavi et al., Eds., p. 671–685, Wiley, 1985.

[20] J. Sakarovitch, *Eléments de théorie des automates*, Vuibert, 2003. English translation: *Elements of Automata Theory*, Cambridge University Press, to appear.

[21] V. Vyssotsky, A counting problem for finite automata, *Tech. report, Bell Telephone Laboratories*, May 1959.