

Random DFAs over a non-unary alphabet

Jean-Marc Champarnaud and Thomas Paranthoën

LIFAR, University of Rouen, 76821 Mont-Saint-Aignan, France.

`{jmc,paranth}@univ-rouen.fr`

Abstract

This document gives a generalization on the alphabet size of the method that is described in Nicaud's thesis for randomly generating complete DFAs. First we recall some properties of m -ary trees and we give a bijection between the set of m -ary trees and the set $\mathfrak{R}_{(m,n)}$ of generalized n -tuples. We show that this bijection can be built on any prefix total order on Σ^* . Then we give the relations that exist between the elements of $\mathfrak{R}_{(m,n)}$ and complete DFAs built on an alphabet of size greater than 2. We give algorithms that allow us to randomly generate accessible complete DFAs. Finally we provide experimental results that show that most of the accessible complete DFAs built on an alphabet of size greater than 2 are minimal.

Introduction

The aim of this article is to generalize the method for randomly generating complete DFAs (on an alphabet of size 2) that was described in Nicaud's thesis [8]. This method is based on the existence of a bijection between binary trees of order n on one hand, prefix sets of cardinality n built on Σ^* with the lexicographic order on the other hand, and finally the set \mathfrak{R}_n of n -tuples. Let us remark first that it is possible to replace the lexicographic order on Σ^* by any prefix total order. We describe in this paper how Nicaud's study can be extended to the case of an alphabet of size greater than 2. We consider the set $\mathfrak{R}_{(m,n)}$ of generalized n -tuples and we show that it allows us to describe the set $\mathfrak{D}_{(m,n)}$ of the complete accessible DFAs of size n on an alphabet of size m . We restate the algorithms described in [8] in the case of a random generation with m symbols. These algorithms allow us to carry out some experiments that show that most of the elements of $\mathfrak{D}_{(m,n)}$ are minimal for $m \geq 3$.

Let us mention that this work is a part of a more general study of random generation of finite automata [2].

Section 1 introduces definitions and notation that are necessary to the comprehension of this document. Section 2 gives some properties of m -ary trees and generalizes the bijection that exists between the set of binary trees, the set of prefix subsets of Σ^* , with Σ of size 2, and the set \mathfrak{R}_n of n -tuples to a bijection between the set of m -ary trees, with $m \geq 2$, the set of prefix subsets of Σ^* , with $|\Sigma| \geq 2$, and the set $\mathfrak{R}_{(m,n)}$ of generalized n -tuples. Section 3 makes explicit the relation between the elements of $\mathfrak{R}_{(m,n)}$ and the deterministic transition structures of size n on an alphabet of size m . Finally Section 4 describes the algorithms for constructing random

transition structures, and reports a set of experimental results based on this random generation method.

1 Definitions and notation

Readers who are not familiar with automata theory are referred to [11].

A *finite non-deterministic automaton* is a 5-tuple $\mathcal{A} = \langle Q, \Sigma, \delta, I, F \rangle$ where $Q = \{q_1, q_2, \dots, q_n\}$ is the finite set of *states*, Σ is the *alphabet* on which the automaton is defined, δ is the *transition function* ($\delta : Q \times \Sigma \rightarrow 2^Q$) (where 2^Q denotes the set of all subsets of Q) that associates a subset of Q to each element of $Q \times \Sigma$, I is a non-empty subset of Q whose elements are the *initial states* and F is a subset of Q whose elements are the *final states*. In this paper the *size* of an automaton is the number of states.

An automaton is said to be *accessible* if and only if for all states $q \in Q$ there exists a path from one of the initial state to this state. An automaton is said to be *co-accessible* if and only if there exists a path from this state to one of the final states. An automaton that is both accessible and co-accessible is a *trim* automaton.

An automaton \mathcal{D} is *deterministic* if it has a unique initial state and if $|\delta(q, x)| \leq 1, \forall q \in Q, \forall x \in \Sigma$. Moreover \mathcal{D} is *complete* if $|\delta(q, x)| = 1, \forall q \in Q, \forall x \in \Sigma$. In what follows, $\mathfrak{D}_{(m,n)}$ will denote the set of accessible complete deterministic automata of size n on an alphabet of size m . We will write $\mathcal{D} = \langle Q, \Sigma, \delta, i, F \rangle$ for a deterministic automaton (DFA) with a unique initial state i .

A *deterministic transition structure* is a 4-tuple $\mathcal{S} = \langle Q, \Sigma, \delta, i \rangle$, that is a DFA without final state set. Thus 2^n DFAs can be produced from a transition structure since there exist 2^n possible final state sets.

An *m-ary tree* is an acyclic directed graph $\mathcal{T} = \langle V, E \rangle$ where $V = \{v_1, v_2, \dots, v_t\}$ is the set of *vertices* of the tree and $E \subseteq V \times V$ is the set of *edges* of the tree. We recall that the *out-degree* (resp. *in-degree*) of a vertex is the number of edges that are incident from (resp. to) this vertex. We let $d^+(v)$ (resp. $d^-(v)$) be the out-degree (resp. in-degree) of a vertex v . The in-degree of each vertex of an m -ary tree is equal to 1, except for one vertex called the *root* and denoted by v_1 that has a zero in-degree. The out-degree of each vertex of an m -ary tree is less than or equal to m . A *complete m-ary tree of order n* is a tree with a partitioning of its vertices $V = N \uplus F$, with $|N| = n$, such that $v \in N \Rightarrow d^+(v) = m$ and $v \in F \Rightarrow d^+(v) = 0$. The set $N = \{r_1, r_2, \dots, r_n\}$ is the set of *nodes*, and $F = \{f_1, f_2, \dots, f_s\}$ is the set of *leaves*. There exists a bijection between m -ary trees with n vertices and complete m -ary trees of order n . Indeed it suffices to attach to each vertex of an m -ary tree $m - d^+(v)$ leaves in order to obtain a complete m -ary tree of order n (Figure 1).

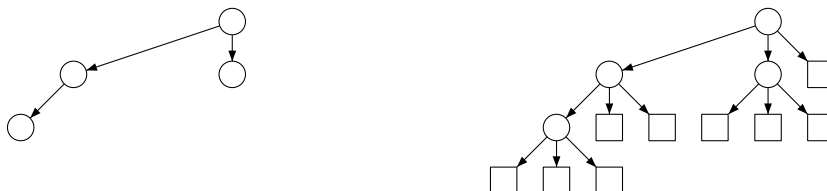


Figure 1: A 3-ary tree and its associated complete 3-ary tree.

A set of words X of Σ^* is *prefix* if it contains all words $u \in \Sigma^*$ such that there

exists $w \in \Sigma$ such that $uw \in X$.

A symbol of Σ can be attached to each edge of an m -ary tree such that for all vertices v and all symbols x , there is at most one edge outgoing from v that is labeled by x . Thus each vertex of an m -ary tree can be labeled by a word w . The label of each vertex v is the label of the path that leads from the root to this vertex. The set of these labels is denoted by $P(\mathcal{T})$. We can show easily that $P(\mathcal{T})$ is prefix. There exists a bijection between the set of prefix subsets of Σ^* of cardinality n and the set of m -ary trees of order n . In the following, $\mathfrak{X}_{(m,n)}$ will denote either one of these two sets.

We assume that Σ is equipped with a total order $<$. Let Σ^* be the free monoid over Σ and $<$ be a total order on Σ^* . Let P be a prefix subset of Σ^* , and \mathcal{T} be the m -ary tree associated with P . Let $P_{<}$ be the list of words of P ordered by the relation $<$. Since the elements of $P_{<}$ are in bijection with the vertices of \mathcal{T} , the order $<$ defines a *traversal* of the vertices of the tree \mathcal{T} . The order in which the words appear in $P_{<}$ corresponds to the order in which the vertices appear throughout the traversal.

We let $u = u_1u_2 \cdots u_m$ and $w = w_1w_2 \cdots w_n$ be words of Σ^* . We define:

$u < w$ for *the lexicographic order* if one of the two following conditions holds:

- (i) there exists an integer $1 \leq k \leq \min(m, n)$ such that $(\forall i, 1 \leq i < k, u_i = w_i)$ and $u_k < w_k$,
- (ii) $m < n$, and $(\forall i, 1 \leq i \leq m, u_i = w_i)$.

The lexicographic order induces a *depth-first traversal* of \mathcal{T} .

$u < w$ for *the graded lexicographic order* if one of the two following conditions holds:

- (i) $m < n$,
- (ii) $n = m$ and there exists $k \leq n$ such that $(\forall i, 1 \leq i < k, u_i = w_i)$ and $u_k < w_k$.

The graded lexicographic order induces a *breadth-first traversal* of the tree \mathcal{T} .

An order $<$ on Σ^* is a *prefix order* if $(\forall u \in \Sigma^*)(\forall x \in \Sigma) u < ux$. The lexicographic order and the graded lexicographic order are prefix orders. We call *prefix traversal* of a tree a traversal induced by a prefix total order.

In what follows, we assume that Σ is an alphabet of size greater or equal to 2 and that Σ^* is equipped with a prefix total order $<$. By convention a complete m -ary tree of order n is such that $m \geq 2$ and $n \geq 1$.

2 Complete m -ary trees and generalized n -tuples

We first present some properties of complete m -ary trees by using the bijection that exists between complete m -ary trees and prefix sets. From the generalization of the classical n -tuples, we deduce a bijection between the set $\mathfrak{X}_{(m,n)}$ of generalized n -tuples and the set $\mathfrak{X}_{(m,n)}$ of complete m -ary trees.

Proposition 2.1 *A complete m -ary tree of order n has $(m - 1)n + 1$ leaves.*

Proof: In any digraph, the sum of the in-degrees is equal to the sum of the out-degrees, because they are both equal to the number of edges. Since in a complete m -ary tree of order n with $|N|$ nodes and $|F|$ leaves, the sum of the in-degrees is equal to $|F| + |N| - 1$, and the sum of the out-degrees is equal to $m|N|$, we obtain $|F| = (m - 1)n + 1$.

Lemma 1 *We consider a prefix traversal of a complete m -ary tree \mathcal{T} of order n . Let k (resp. l) be the number of nodes (resp. leaves) visited at a given moment. The following properties hold:*

$$(l \leq (m - 1)k + 1) \text{ and}$$

$$(l = (m - 1)k + 1) \Rightarrow k = n$$

Proof: In the subgraph of \mathcal{T} induced by the prefix traversal the sum of the in-degrees is $l + k - 1$. Moreover the out-degree of each of the k nodes is not greater than m . Thus the sum of the out-degrees is not greater than mk , and we get:

$$l \leq (m - 1)k + 1 \tag{1}$$

We assume that at a given moment t we have $k < n$ and $l = (m - 1)k + 1$. Let v be the vertex visited at the moment $t + 1$. We let k' (resp. l') be the number of nodes (resp. leaves) visited at the instant $t + 1$.

We distinguish two cases:

v is a leaf: We get $k' = k$ and $l' = l + 1$. Since by hypothesis $l = (m - 1)k + 1$, we thus have $l' > (m - 1)k' + 1$, which is in contradiction with (1).

v is a node: We get $k' = k + 1$ and $l' = l$. Since the number of edges is less or equal to mk and the sum of the in-degrees is equal to $k' + l' - 1$, we obtain $k' + l' - 1 \leq mk$, and $l \leq (m - 1)k$, which is in contradiction with the assumptions.

Let \mathcal{T} be a tree and F be its set of leaves. Let F_{\prec} be the list of leaves collected during the prefix traversal of \mathcal{T} induced by \prec . Let the function $\phi : F \rightarrow \mathbb{N}$ that associates with each leaf of \mathcal{T} the number of nodes visited before it during this traversal. We have $\phi(f_{i+1}) \geq \phi(f_i), \forall f_i \in F_{\prec}$.

Proposition 2.2 *Let \mathcal{T} be a complete m -ary tree of order n . The number of nodes visited before the i -th leaf (except for the last one) during a prefix traversal is greater or equal to $\lceil \frac{i}{m-1} \rceil$.*

$$(\forall f_i \in F_{\prec})(1 \leq i < (m - 1)n + 1) \quad n \geq \phi(f_i) \geq \left\lceil \frac{i}{m - 1} \right\rceil$$

$$\phi(f_{(m-1)n+1}) = n$$

Proof: The proof is by induction on the number of nodes visited before a leaf during the prefix traversal. Let $s = (m - 1)n + 1$ be the number of leaves.

Basis $i = 1$: The number of nodes that are visited before the first leaf is strictly positive, otherwise the order of tree is zero.

Induction step $s - 2 \geq i \geq 1$: We assume that the property is true for the i -th leaf. We get:

$$\phi(f_{i+1}) \geq \phi(f_i) \geq \left\lceil \frac{i}{m-1} \right\rceil \quad (2)$$

We then distinguish two cases:

$i \bmod (m-1) \neq 0$: We have $\lceil \frac{i}{m-1} \rceil = \lceil \frac{i+1}{m-1} \rceil$, and the property is true for the $(i+1)$ -th leaf.

$i \bmod (m-1) = 0$: If at least one of the inequalities of the assumption (2) is strict, we get $\phi(f_{i+1}) > \frac{i}{m-1}$ and consequently $\phi(f_{i+1}) \geq \lceil \frac{i+1}{m-1} \rceil$. Thus the property holds for the $(i+1)$ -th leaf. Else we get $\phi(f_{i+1}) = \phi(f_i) = \frac{i}{m-1}$. This implies $i+1 = (m-1)\phi(f_{i+1}) + 1$. According to Lemma 1, we obtain $\phi(f_{i+1}) = n$ and thus $i+1 = s$. But by assumption $i+1 < s$. Therefore the contradiction.

Thus the property holds for all leaves except for the last one.

The set \mathfrak{R}_n of the n -tuples of elements of $\llbracket 1, n \rrbracket$ is defined as:

$$\mathfrak{R}_n = \{(k_1, \dots, k_n, n) \in \llbracket 1, n \rrbracket^{n+1} \mid \forall i \in \llbracket 2, n \rrbracket, k_i \geq k_{i-1} \text{ and } \forall i \in \llbracket 1, n \rrbracket, k_i \geq i\}$$

This set can be generalized to the set $\mathfrak{R}_{(m,n)}$ of the *generalized* n -tuples of elements of $\llbracket 1, n \rrbracket$ defined as:

$$\mathfrak{R}_{(m,n)} = \left\{ (k_1, \dots, k_{s-1}, n) \in \llbracket 1, n \rrbracket^s \mid \forall i \in \llbracket 1, s-1 \rrbracket, k_i \geq \left\lceil \frac{i}{m-1} \right\rceil \text{ and } \forall i \in \llbracket 2, s \rrbracket, k_i \geq k_{i-1} \right\}$$

where $s = n(m-1) + 1$.

We consider the function $\varphi : \mathfrak{Z}_{(m,n)} \rightarrow \mathfrak{R}_{(m,n)}$ that associates with a complete m -ary tree \mathcal{T} of order n the element of $\mathfrak{R}_{(m,n)}$ defined by:

$$\varphi(\mathcal{T}) = (\phi(f_1), \phi(f_2), \dots, \phi(f_{n(m-1)}), \phi(f_{n(m-1)+1}))$$

In the following \mathcal{K} will denote an element of $\mathfrak{R}_{(m,n)}$.

Proposition 2.3 *For all $n \geq 1$, $m \geq 2$ the function φ is a bijection from $\mathfrak{Z}_{(m,n)}$ to $\mathfrak{R}_{(m,n)}$.*

Proof: According to Proposition 2.2 and definition of $\mathfrak{R}_{(m,n)}$, φ has its values in $\mathfrak{R}_{(m,n)}$. On the other hand let us consider \mathcal{T} and \mathcal{T}' two distinct trees of $\mathfrak{Z}_{(m,n)}$, and F and F' the sets of words that label these two trees. Let u be the smallest word according to \prec such that $u \in P \cup P'$ and $u \notin P \cap P'$. We assume that $u \in P$. We let $\varphi(\mathcal{T}) = (\phi(f_1), \phi(f_2), \dots, \phi(f_{n(m-1)+1}))$ and $\varphi(\mathcal{T}') = (\phi(f'_1), \phi(f'_2), \dots, \phi(f'_{n(m-1)+1}))$. By definition there exists l such that f_l is the leaf labeled by u , and such that for all $i < l$, $\phi(f_i) = \phi(f'_i)$. Thus $\phi(f_l) < \phi(f'_l)$ and φ is injective.

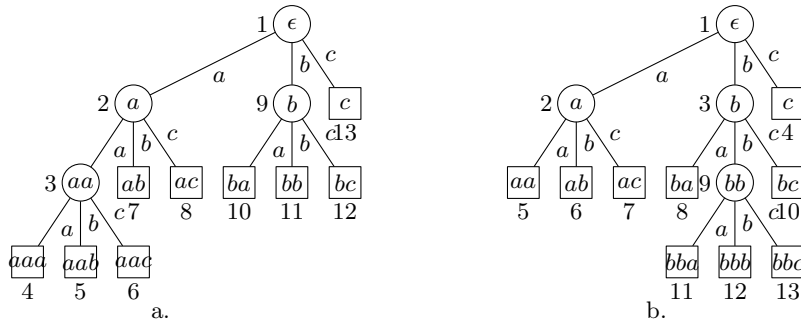


Figure 2: 3-ary trees equivalent to the generalized n -tuple: $(3, 3, 3, 3, 3, 4, 4, 4, 4)$, according to the lexicographic order (a) or to the graded lexicographic one (b).

From Proposition 2.2, we can associate with each element $\mathcal{K} = (k_1, k_2, \dots, k_{n(m-1)}, n)$ at least one element of $\mathfrak{X}_{(m,n)}$, therefore φ is surjective, and then bijective.

Figure 2 illustrates the construction of a complete tree from a generalized n -tuple. Tree vertices are labeled in the order of their creation.

We have today a good knowledge of the different objects in bijection with m -ary trees. We close this section with other known results on m -ary trees.

We define for all $(m, n) \in \mathbb{N}^2$ the *generalized Catalan numbers* [10, 5] as:

$$C_n^{(m)} = \frac{1}{mn+1} \binom{mn+1}{n}$$

These numbers describe the number of m -ary trees of order n . On the other hand, the bijection that exists between binary trees and *Dyck words*, can be generalized to well balanced bracketed words that contain $m-1$ right brackets for one left bracket (Figure 3.d). The grammar of these words for an alphabet of size m is:

$$S \rightarrow a \underbrace{SbSb \dots Sb}_{m-1 \text{ terms } Sb} S \mid \epsilon$$

These words can also be viewed as sequences $u = x_1 x_2 \dots x_{n(m-1)+n}$ of 0s and 1s called *well m -balanced sequences* that satisfy the following properties [10]:

- (i) u contains $(m-1) \times n$ 1s for n 0s,
- (ii) for all i , such that $1 \leq i \leq n(m-1) + n$ we have:

$$|\{j \mid 1 \leq j \leq i, x_j = 0\}| \geq \frac{|\{j \mid 1 \leq j \leq i, x_j = 1\}|}{m-1}$$

These sequences have been studied in probabilistic mathematics in the general case, and in combinatorics in the case of binary trees (“ballot problem” [4], “Dyck word” [7]). They are in bijection with the *walks above the sea level* that have an increasing slope $m-1$ times greater than the decreasing one (Figure 3.b). Computer scientists also call them *Dyck paths*. Finally the graphical representation of the n -tuples gives rise to the *player sequence* which is a set of blocks that are contained in a rectangle and that contains the negative slope diagonal of this rectangle (Figure 3.c).

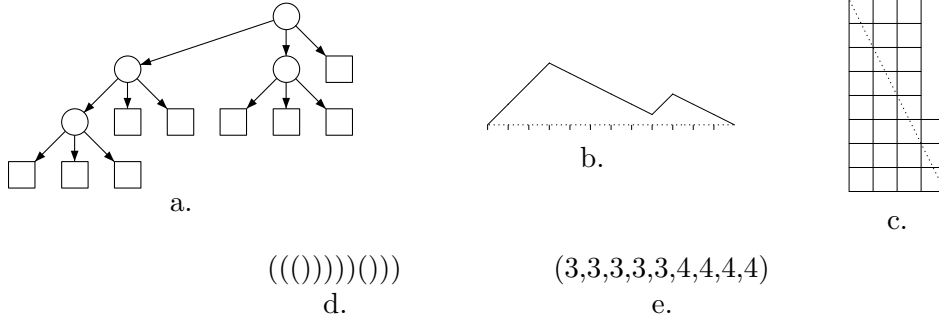


Figure 3: Illustration of the different objects in bijection: (a) complete m -ary tree, (b) path above the sea level, (c) player sequence, (d) well balanced sequence, (e) generalized n -tuple.

3 Relation between complete deterministic automata and complete m -ary trees

Nicaud's study shows that the classical n -tuples allow us to build and to count the DFAs on an alphabet of size 2. We show that the notion of canonical labeling extends naturally to the case of an alphabet of size $m \geq 2$. This permits us to establish the relations that exist between the elements of $\mathfrak{R}_{(m,n)}$ and those of $\mathfrak{D}_{(m,n)}$, and to give some bounds of $|\mathfrak{D}_{(m,n)}|$.

Let $\mathcal{D} = \langle Q, \Sigma, \delta, i, T \rangle$, $\mathcal{D} \in \mathfrak{D}_{(m,n)}$ be an accessible deterministic automaton. We recall that Σ^* is equipped with a prefix total order. We associate with each state q of this automaton the word:

$$w(q) = \min_{<} \{ u \in \Sigma^* \mid \delta(i, u) = q \text{ and } u \text{ is a simple path on } \mathcal{D} \}$$

Since the automaton is accessible this word exists. Since the automaton is deterministic and the order is total, this word is unique. The labeling induced by the application w is canonical. Two distinct complete accessible deterministic automata that are canonically labeled cannot be isomorphic (if the labelings of their states are identical, their transition tables are necessarily different).

We denote by $P(\mathcal{D})$ the set of labels of the states of \mathcal{D} by w :

$$P(\mathcal{D}) = \{ w(q) \mid q \in Q \}$$

Proposition 3.1 *For all automata \mathcal{D} of $\mathfrak{D}_{(m,n)}$ the set $P(\mathcal{D})$ is prefix.*

Proof: We assume that there exists a word $uv \in P(\mathcal{D})$ such that $u \notin P(\mathcal{D})$. Since the automaton is complete, $w(\delta(q_0, u))$ exists, and $w(\delta(q_0, u)) < u$. Since the order $<$ is prefix $w(\delta(q_0, u))v < uv$. This leads to a contradiction.

Prefix sets are in bijection with complete m -ary trees, thus the transition structures reduced to the smallest paths from the initial state to each one of the states are in bijection with complete m -ary trees.

Proposition 3.2 *The set of the accessible complete deterministic transition structures of size n on an alphabet of size m can be generated with the elements of $\mathfrak{R}_{(m,n)}$.*

Each element \mathcal{K} of $\mathfrak{R}_{(m,n)}$ can generate:

$$\|\mathcal{K}\| = \|(k_1, \dots, k_{n(m-1)}, n)\| = n \prod_{i=1}^{n(m-1)} k_i \quad \text{structures, thus}$$

$$|\mathfrak{D}_{(m,n)}| = 2^n \sum_{\mathcal{K} \in \mathfrak{R}_{(m,n)}} \|\mathcal{K}\|$$

Proof: Let \mathcal{K} be an element of $\mathfrak{R}_{(m,n)}$, and $\mathcal{T} = (V, E)$ be its unique associated complete tree. We denote by N and F respectively the sets of nodes and of leaves of \mathcal{T} . The transition structure defined by $\mathcal{S} = \langle N, \Sigma, E \cap (N \times N), v_1 \rangle$ contains $n - 1$ transitions and is accessible. In order to obtain a complete deterministic transition structure, we add to this structure the $(m - 1) \times n + 1$ transitions corresponding to the edges that lead from a node to a leaf.

Let f be a leaf of the tree, and u be its label. Let p be the parent of f . We consider the edge (p, f) that is labeled by x . The addition of the edge (p, r) , $r \in N$ labeled by x to the transition structure \mathcal{S} does not change the labeling of the states of \mathcal{S} if $w(r) \prec u$. The number of different edges (p, r) that can be added is thus equal to the number of nodes r whose labels are smaller than u . This number is equal to k_i for the leaf f_i , $1 \leq i \leq (m - 1)n + 1$. Hence the expression of the number of transition structures that can be built from a generalized n -tuple.

Finally there exist 2^n different final sets, hence the number of complete deterministic automata of size n on an alphabet of size m .

This result permits to define some bounds on the number of automata of a given size:

Proposition 3.3 *We have the following inequalities:*

$$(i) \quad (2\pi)^{\frac{m-2}{2}} e^{-(m-1)n+1} m^{n-1-\alpha-\beta} n^{(m-1)n-1-\alpha+m/2} \leq \frac{|\mathfrak{D}_{(m,n)}|}{2^n}$$

$$\frac{|\mathfrak{D}_{(m,n)}|}{2^n} \leq \frac{e}{\sqrt{2\pi}} m^{n-1-\alpha-\beta} n^{(m-1)n-1/2-\alpha}, \quad \text{with}$$

$$\alpha = \left((m-1)n + \frac{3}{2} \right) \left(\frac{\log \left(\frac{(m-1)n+1}{(m-1)n} \right)}{\log((m-1)n)} \right) \quad \beta = \left((m-1)n + \frac{3}{2} + \alpha \right) \left(\frac{\log \left(\frac{m-1}{m} \right)}{\log(m)} \right)$$

$$(ii) \quad [8] \quad \sqrt{2} 4^n e^{-n} n^n (1 + o(1)) \leq \frac{|\mathfrak{D}_{(2,n)}|}{2^n} \leq \frac{1}{\sqrt{\pi}} 4^n n^{n-1/2} (1 + o(1))$$

$$(iii) \quad [3] \quad \frac{|\mathfrak{D}_{(m,n)}|}{2^n} \leq \frac{n^{mn}}{(n-1)!}$$

Proof: The product of the elements of an element \mathcal{K} of $\mathfrak{R}_{(m,n)}$ is bounded by:

$$n \times (n!)^{(m-1)} \leq \|\mathcal{K}\| \leq n \times n^{n(m-1)}$$

Thus, by using the fact that the generalized Catalan numbers describe the number of elements of $\mathfrak{R}_{(m,n)}$, we get the following inequalities:

$$n \times \frac{(n!)^{(m-1)}}{mn+1} \binom{mn+1}{n} \leq \frac{|\mathfrak{D}_{(m,n)}|}{2^n} \leq n \times \frac{n^{n(m-1)}}{mn+1} \binom{mn+1}{n}$$

Thanks to some simplifications and using Stirling approximation we get the bounds (i). In the case of a binary alphabet, the above expression can be approximated and we get the bounds (ii) given by Nicaud. Finally, (i) can be improved, since the number of accessible transition structures is smaller than the number n^{nm} of sets of m deterministic but not necessarily accessible transition functions. And since there exist $(n-1)!$ different ways to label these structures, we deduce the inequality (iii). Notice that a better upper bound, based on accessible DFAs, is presented in [6, 9, 3].

4 Algorithms for the construction of transition structures

We give first a recurrence relation that expresses the number of deterministic complete transition structures of size n on an alphabet of size m . We deduce from this relation an algorithm that computes this class of numbers; this allows us to give an algorithm that randomly generates a generalized n -tuple according to the number of different transition structures that can be deduced from this n -tuple.

4.1 Construction of the elements of $\mathfrak{R}_{(m,n)}$

In [8], it is shown that n -tuples can be computed via recursive formulae. Following this approach, we define the following generalization of $\mathfrak{R}_{(m,n)}$:

$$\mathfrak{R}_{(m,t,p)} = \left\{ (k_1, k_2, \dots, k_t) \in \llbracket 1, p \rrbracket^t \mid \forall i \in \llbracket 1, t \rrbracket k_i \geq \left\lceil \frac{i}{m-1} \right\rceil \text{ and } \forall i \in \llbracket 2, t \rrbracket k_i \geq k_{i-1} \right\}$$

Notice that for all m and n , an element of $\mathfrak{R}_{(m,n)}$ (up to its last element) is an element of $\mathfrak{R}_{(m,n(m-1),n)}$.

$$\text{We let for all } m, t \text{ and } p: c_{(m,t,p)} = \sum_{\mathcal{K}_{(m,t,p)} \in \mathfrak{R}_{(m,t,p)}} \|\mathcal{K}_{(m,t,p)}\|$$

Proposition 4.1 *For all $t, p \geq 1$ and $m \geq 2$, the following relations hold:*

$$\begin{cases} c_{(m,t,p)} = 0 & \text{if } p < \left\lceil \frac{t}{m-1} \right\rceil, \\ c_{(m,t,p)} = \frac{1}{2}p(p+1) & \text{if } t = 1, \\ c_{(m,t,p)} = c_{(m,t,p-1)} + p \times c_{(m,t-1,p)} & \text{otherwise.} \end{cases}$$

Proof: If $p < \left\lceil \frac{t}{m-1} \right\rceil$ then $k_i \leq p < \left\lceil \frac{t}{m-1} \right\rceil$, and the condition $k_i \geq \left\lceil \frac{i}{m-1} \right\rceil$ cannot be satisfied. If $t = 1$ then $c_{(m,1,p)} = \sum_{i=1}^p i = \frac{1}{2}p(p+1)$.

For the recurrence relation, it is sufficient to remark that an element of $\mathfrak{R}_{(m,t,p)}$ not ending with p is in $\mathfrak{R}_{(m,t,p-1)}$. If it ends with p then it has the form $(k_1, k_2, \dots, k_{t-1}, p)$, with $(k_1, k_2, \dots, k_{t-1}) \in \mathfrak{R}_{(m,t-1,p)}$. Thus $\|(k_1, k_2, \dots, k_{t-1}, p)\| = p \|(k_1, k_2, \dots, k_{t-1})\|$.

```

1 proc calculus_of_the_array_of_the_c(m : integer, t : integer, k : integer)
2   ≡
3   var
4     T : array of [1, t] [0, k]
5   Begin
6     For j ← 1 a k do
7       T[1][j] ←  $\frac{1}{2}j(j+1)$ 
8     od
9     For i ← 2 a t do
10      For j ← 0 a k do
11        If j <  $\lceil \frac{i}{m-1} \rceil$ 
12          then T[i][j] ← 0
13          else T[i][j] ← T[i][j-1] + jT[i-1][j]
14        fi
15      od
16    od
17    return T
18  End

```

Figure 4: Algorithm that builds the $c_{(m,t,p)}$.

The elements $c_{(m,t,p)}$ allow us to compute the number of complete accessible deterministic transition structures on an alphabet of size m and to generate these structures. The algorithm that builds $c_{(m,t,p)}$ elements is described in Figure 4.

The array built by this algorithm can be viewed as a Pascal-like triangle. It avoids computing the same values many times, due to the recursive definition of $c_{(m,t,p)}$. Figure 5 represents $c_{(m,t,p)}$ for $m = 3$, $1 \leq t \leq 16$ and $1 \leq p \leq 8$. It shows for example that there exist $c_{(3,4,2)} \times 2 = 28 \times 2 = 56$ complete deterministic transition structures of size 2 on an alphabet of size 3.

$t \backslash p$	1	2	3	4	5	6	7	8
1	1	3	6	10	15	21	28	36
2	1	7	25	65	140	266	462	750
3	0	14	89	349	1049	2645	5879	11879
4	0	28	295	1691	6936	22806	63959	158991
5	0	0	885	7649	42329	179165	626878	1898806
6	0	0	2655	33251	244896	1319886	5708032	20898480
7	0	0	0	133004	1357484	9276800	49233024	216420864
8	0	0	0	532016	7319436	62980236	407611404	2138978316
9	0	0	0	0	36597180	414478596	3267758424	20379584952
10	0	0	0	0	182985900	2669857476	25544166444	188580846060
11	0	0	0	0	0	16019144856	194828309964	1703475078444
12	0	0	0	0	0	96114869136	1459913038884	15087713666436
13	0	0	0	0	0	0	10219391272188	130921100603676
14	0	0	0	0	0	0	71535738905316	1118904543734724
15	0	0	0	0	0	0	0	8951236349877792
16	0	0	0	0	0	0	0	71609890799022336

Figure 5: Table of the $c_{(3,t,p)}$ for t from 1 to 16 and p from 1 to 8.

From the bounds given in Proposition 3.3 the growth of the numbers $c(m, (m-1)n, n)$ is in the worst case of order $n^{(m-1)n + \frac{n-1}{\log(n)}}$, thus their size is of order $((m-1)n + \frac{n-1}{\log(n)}) \log(n)$. The size of these numbers gives rise to some implementation problems, since the memory space used to build the table becomes quickly huge; for example the table necessary to randomly generate automata of size 1000 on an alphabet of size 2 needs around 250 MB with the *GMP* mathematics library [1].

The algorithm that generates a random generalized n -tuple \mathcal{K} takes as parameters the array built by the previous algorithm, and produces a random element of $\mathfrak{R}_{(m,n)}$ according to the number of transition structures it can generate (Figure 6). It assumes that we have a function *append* which concatenates an integer e to the end

```

1 proc random_element_of_K(m : integer, t : integer, k : integer)
2   ≡
3   Begin
4     If k <  $\lceil \frac{t}{m-1} \rceil$  then return  $\emptyset$ 
5     fi
6     If t = 1
7       then
8         De  $\leftarrow$  Random( $\llbracket 1, T[1][k] \rrbracket$ )
9         x  $\leftarrow$  1
10        While De > x do
11          De  $\leftarrow$  De - x
12          x  $\leftarrow$  x + 1
13        od
14        return(x)
15      else
16        De  $\leftarrow$  Random( $\llbracket 1, T[t][k] \rrbracket$ )
17        If (De  $\leq$   $T[t][k-1]$ )  $\wedge$  (k > 0)
18          then return random_element_of_K(t, k - 1)
19          else return append(random_element_of_K(t - 1, k), k)
20        fi
21      fi
22    End

```

Figure 6: Algorithm that randomly generates an accessible complete DFA.

of an integer list l and returns the new list: $append(l : list, e : integer) \rightarrow list$. Lines 10-13 allow us to generate a random number $1 \leq x \leq k$ such that each integer $1 \leq l \leq k$ has the probability $\frac{l}{k(k+1)}$ to be generated. It is the expression of $c_{(m,t,p)}$ for $t = 1$. Lines 15-20 follow the definition of $c_{(m,t,p)}$. In order to generate a random $\mathcal{K}_{(m,n)}$ we call this procedure as follows: $random_element_of_K(m, n(m-1), n)$.

Once an element \mathcal{K} is randomly generated and its associated tree is built, one of the automata associated to this tree can be built according to Proposition 3.2.

4.2 Experimental results

The tests have been performed with a program written in C++ that uses the library *GMP*. The generated DFAs are of size 100, and for each test and each possible number of final states, 10 000 DFAs have been randomly generated.

For an alphabet of size 2, it appears (Figure 7) that accessible complete DFAs are minimal with a probability of 0.8. That is consonant with Nicaud's results.

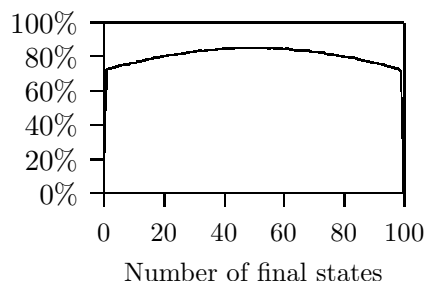


Figure 7: Percentage of complete minimal DFAs of size 100 on an alphabet of size 2, according to the number of final states.

For an alphabet of size greater than 2, we have observed that almost all accessible complete DFAs are minimal (except for those whose final state set is empty or contains all states). This observation is illustrated by Figure 8, for DFAs of size 100; notice that it is still valid for DFAs of small size.

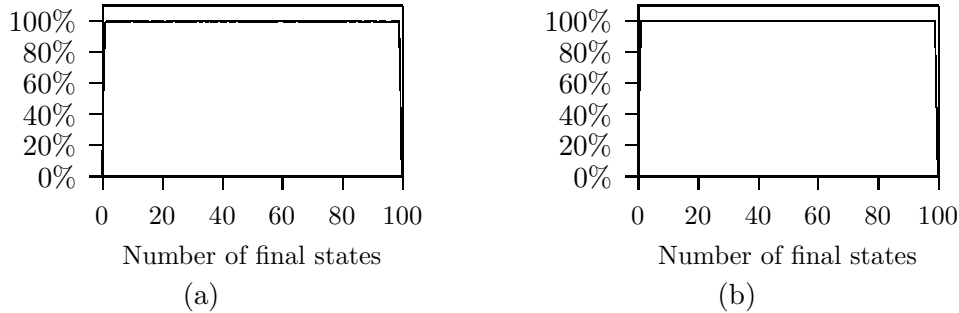


Figure 8: Percentage of complete minimal DFAs of size 100 on an alphabet of size 3 (a) and 5 (b) according to the number of final states.

5 Conclusion

Experimental results show that the use of such a generation method allows us to build random minimal complete automata. Indeed, as observed by Nicaud, automata generated on a binary alphabet are minimal with an empirical probability of 0.8. Moreover almost all automata generated on an alphabet of a larger size are minimal. Thus a random generation method with rejection can be used to randomly generate minimal deterministic automata.

References

1. GMP, gnu multiple precision library. www.swox.com/gmp/.
2. J.-M. Champarnaud, G. Hansel, T. Paranthoën, and D. Ziadi. Nfas bitstream-based random generation. In J. Dassow, M. Hoeberechts, H. Jürgensen, and D. Wotschke, editors, *Proceedings of DCFS 2002 - Descriptive Complexity of Formal Systems, London Ontario Canada*, pages 81–94, 2002.
3. M. Domaratzki, D. Kisman, and J. Shallit. On the number of distinct languages accepted by finite automata with n states. In *Proceedings, Descriptive Complexity of Automata, Grammars and Related Structures (DCAGRS)*, pages 67–78, 2001.
4. W. Feller. *An Introduction to Probability Theory and its Application*. Wiley, 1950.
5. P. Hilton and J. Pedersen. Catalan numbers, their generalization, and their uses. *Math. Intelligencer*, 13(2):64–75, 1991.
6. V. A. Liskovets. The number of connected initial automata. *Kibernetika*, 3(5):16–19, 1969.
7. M. Lothaire. *Combinatorics on Words*. Addison-Wesley, 1983.
8. C. Nicaud. *Etude du comportement en moyenne des automates finis et des langages rationnels*. PhD thesis, Université Paris 7, 2000.
9. R. W. Robinson. Counting strongly connected finite automata. *Graph Theory with Applications to Algorithms and Computer Science*, pages 671–685, 1985.
10. U. Tamm. Lattice paths not touching a given boundary. *Journal of Statistical Planning and Inference*, 105(2):403–448, 2002.
11. S. Yu. Regular languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages, Volume I, Word, Language, Grammar*, pages 41–110. Springer, Berlin, 1997.