Random Generation of DFAs

Jean-Marc Champarnaud and Thomas Paranthoën

University of Rouen, LIFAR, F-76821 Mont-Saint-Aignan Cedex, France

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 $\mathfrak{K}_{(m,n)}$ of generalized tuples. We show that this bijection can be built on any total prefix order on $\Sigma^\star$. Then we give the relations that exist between the elements of $\mathfrak{K}_{(m,n)}$ 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.

Key words: Complete deterministic automata, Random generation, Catalan families, m-ary trees

Introduction

The random generation of DFAs allows us to get some empirical observations that lead to theoretical results in the average case on the classical algorithms that are applied on DFAs. Although the random generation of unary DFAs is trivial, Nicaud has used their natural structure to give the average state complexity of the classical operations on unary DFAs [12]. Moreover he describes in his thesis [13] a method for randomly generating complete accessible DFAs on an alphabet of size 2. We show in this paper how this method can be extended to the case of DFAs built on an alphabet of an arbitrary size.

Nicaud's method deals with binary trees and an other Catalan family: the n-tuples. The n-tuples allow one to count the number of deterministic structures that can be produced from a given binary tree (a deterministic structure

Email address: {`Jean-Marc.Champarnaud, Thomas.Paranthoen`}`@univ-rouen.fr` (Jean-Marc Champarnaud and Thomas Paranthoën).

is a DFA without final states). In this paper these two Catalan families are extended to an alphabet of an arbitrary size. And we thus restate the algorithms presented in [13] in this case. With these algorithms, we carry out some experiments that enlight the fact that most of the DFAs are minimal as far as the size of the alphabet is greater than 2.

Let us mention that this work is a part of a more general study on the random generation of finite automata [4].

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 $\Sigma^\star$, 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 $\Sigma^\star$, with $|\Sigma| \geq 2$, and the set $\mathfrak{R}_{(m,n)}$ of generalized 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 [16]. We denote by $[\![i,j]\!] \quad \forall (i,j) \in \mathbb{N}^2$, the set $[i,j] \cap \mathbb{N}$.

A *finite non-deterministic automaton* is a 5-tuple $\mathscr{A} =< Q, \Sigma, \delta, I, F >$ 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 its 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 state q. An automaton is said to be *co-accessible* if and only if there exists a path from state q to one of the final states. An automaton that is both accessible and co-accessible is a *trim* automaton.

An automaton $\mathscr{D}$ 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 $\mathscr{D}$ 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

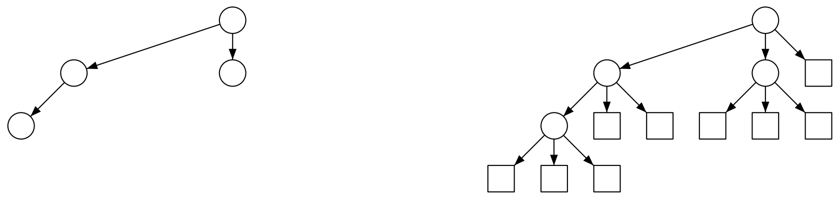


Figure 1. A 3-ary tree and its associated extended 3-ary tree.

$\mathscr{D} =< Q, \Sigma, \delta, i, F >$ for a deterministic automaton (DFA) with a unique initial state i.

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

An *m-ary tree* is an acyclic directed graph $\mathscr{T} =< V, E >$ where $V = \{v_1, v_2, \ldots, v_t\}$ 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. An *extended* m-ary tree of *order* n is a tree with a partitioning of its vertices $V = N \uplus L$, with $|N| = n$, such that $v \in N \Rightarrow d^+(v) = m$ and $v \in L \Rightarrow d^+(v) = 0$. The set $N = \{r_1, r_2, \ldots, r_n\}$ is the set of *nodes*, and $L = \{\ell_1, \ell_2, \ldots, \ell_s\}$ is the set of *leaves*. There exists a bijection between m-ary trees with n vertices and extended m-ary trees of order n. Indeed it suffices to attach to each vertex v of an m-ary tree $m - d^+(v)$ leaves in order to obtain an extended m-ary tree of order n (Figure 1).

A set of words X of $\Sigma^\star$ is *prefix* if it contains all words $u \in \Sigma^\star$ such that there exists $w \in \Sigma$ such that $uw \in X$.

Let Σ be an alphabet of size m. 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, as it is done for tRIes or within prefix codes [9, 3]. 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 $P(\mathscr{T})$. We can show easily that the set $P(\mathscr{T})$ is prefix. There exists a well known bijection between the set of prefix subsets of $\Sigma^\star$ of cardinality n and the set of m-ary trees of order n. In the following, $\mathfrak{T}_{(m,n)}$ will denote either one of these two sets.

We assume that Σ is equipped with a total order $<$. Let $\Sigma^\star$ be the free monoid over Σ and $\prec$ be a total order on $\Sigma^\star$ induced by $<$. Let P be a prefix subset of

$\Sigma^\star$, and $\mathscr{T}$ be the m-ary tree associated with P. Let $P_\prec$ be the list of words of P ordered by the relation $\prec$. Since the elements of $P_\prec$ are in bijection with the vertices of $\mathscr{T}$, the order $\prec$ defines a *traversal* of the vertices of the tree $\mathscr{T}$. The order in which the words appear in $P_\prec$ corresponds to the order in which the vertices appear throughout the traversal.

We let $u = u_1 u_2 \cdots u_m$ and $w = w_1 w_2 \cdots w_n$ be words of $\Sigma^\star$ and $\mathscr{T}$ a $|\Sigma|$-ary tree. We define:

$u \prec w$ 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 $\mathscr{T}$.

$u \prec w$ 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 $\mathscr{T}$.

An order $\prec$ on $\Sigma^\star$ is a *prefix order* if

$$(\ \forall u \in \Sigma^\star\ )(\ \forall x \in \Sigma\ ) \quad u \prec ux.$$

The lexicographic order and the graded lexicographic order are prefix orders. We call *prefix traversal* of a tree a traversal induced by a total prefix order.

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

2 Complete m-ary trees and generalized tuples

We first present some properties of extended m-ary trees. Then will follow the generalization of the classical n-tuples, that permits us to deduce a bijection between the set $\mathfrak{R}_{(m,n)}$ of generalized tuples and the set $\mathfrak{T}_{(m,n)}$ of extended m-ary trees.

Proposition 1 *An extended m-ary tree of order n has $(m-1)n+1$ leaves.*

The proof of this proposition can be found in [2].

Lemma 2 *We consider a prefix traversal of an extended m-ary tree $\mathscr{T}$ of order n. Let k (resp. r) be the number of nodes (resp. leaves) visited at a step of the prefix traversal. The following properties hold:*

(i) $(\ r \leq (m-1)k+1 \ )$,
(ii) $(\ r = (m-1)k+1 \ ) \Rightarrow k = n$.

PROOF. In the subgraph of $\mathscr{T}$ induced by the prefix traversal the sum of the in-degrees is $r+k-1$. 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

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

We assume that at the current step we have $k < n$ and $r = (m-1)k+1$. Let v be the vertex visited next. We let k' (resp. r') be the new number of visited nodes (resp. leaves).

We distinguish two cases:

v is a leaf: we get $k' = k$ and $r' = r+1$. Since by hypothesis $r = (m-1)k+1$, we thus have $r' > (m-1)k'+1$, which is in contradiction with (1).
v is a node: we get $k' = k+1$ and $r' = r$. Since no successor of v has been visited before v itself, the number of edges is less or equal to mk and the sum of the in-degrees is equal to $k'+r'-1$. Thus we obtain $k'+r'-1 \leq mk$, and $r \leq (m-1)k$, which is in contradiction with the assumptions. □

Let $\mathscr{T}$ be a tree and L be its set of leaves. Let $L_\prec$ be the list of leaves met during the prefix traversal of $\mathscr{T}$ induced by $\prec$. Let the function $\phi : L \to \mathbb{N}$ that associates with each leaf of $\mathscr{T}$ the number of nodes visited before it during this traversal. We have $\phi(\ell_{i+1}) \geq \phi(\ell_i), \forall \ell_i \in L_\prec$.

Proposition 3 *Let $\mathscr{T}$ be an extended 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 than or equal to $\lceil \frac{i}{m-1} \rceil$. More precisely we have:*

(i) $(\ \forall \ell_i \in L_\prec \ )(\ 1 \leq i < (m-1)n+1 \ ) \quad n \geq \phi(\ell_i) \geq \left\lceil \dfrac{i}{m-1} \right\rceil$,
(ii) $\phi\Big(\ell_{(m-1)n+1}\Big) = n$.

PROOF. The proof of *(i)* is by induction on the number of nodes visited before a leaf during the prefix traversal. Let $|L| = (m-1)n+1$ 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 $\mathscr{T}$ is zero.
Induction step $|L| - 2 \geq i \geq 1$: we assume that the property is true for the *i-th* leaf. We get

$$\phi(\ell_{i+1}) \geq \phi(\ell_i) \geq \left\lceil \frac{i}{m-1} \right\rceil. \tag{2}$$

We then distinguish two cases:
i mod $(m-1) \neq 0$: 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 mod $(m-1) = 0$: if at least one of the inequalities of the assumption (2) is strict, we get $\phi(\ell_{i+1}) > \frac{i}{m-1}$ and consequently $\phi(\ell_{i+1}) \geq \lceil \frac{i+1}{m-1} \rceil$. Thus the property holds for the $(i+1)$-th leaf. Otherwise we get $\phi(\ell_{i+1}) = \phi(\ell_i) = \frac{i}{m-1}$. This implies $i+1 = (m-1)\phi(\ell_{i+1}) + 1$. According to Lemma 2.*(ii)*, we obtain $\phi(\ell_{i+1}) = n$ and thus $i + 1 = |L|$. But by assumption $i + 1 < |L|$. Therefore the contradiction.
Thus the property holds for all leaves except for the last one.

On the other hand *(ii)* is a direct consequence of the Lemma 2.*(ii)*. $\square$

The set $\mathfrak{K}_n$ of the *n-tuples* of elements of $[\![1, n]\!]$ is defined as

$$\mathfrak{K}_n = \{(k_1, \ldots, k_n) \in [\![1, n]\!]^n \mid \forall i \in [\![2, n]\!], k_i \geq k_{i-1} \textit{ and } k_i \geq i\}.$$

This set can be generalized to the set $\mathfrak{K}_{(m,n)}$ of the *generalized tuples* of elements of $[\![1, n]\!]$ defined as

$$\mathfrak{K}_{(m,n)} = \left\{ (k_1, \ldots, k_s) \in \Big[\!\!\Big[1, n\Big]\!\!\Big]^s \;\middle|\; \forall i \in [\![2, s]\!], k_i \geq \left\lceil \frac{i}{m-1} \right\rceil \textit{ and } k_i \geq k_{i-1} \right\},$$

where $s = n(m-1)$. In the following $\mathcal{K}$ will denote an element of $\mathfrak{K}_{(m,n)}$.

We consider the function $\varphi : \mathfrak{T}_{(m,n)} \to \mathfrak{K}_{(m,n)}$ that associates with an extended m-ary tree $\mathscr{T}$ of order n the element of $\mathfrak{K}_{(m,n)}$ defined by

$$\varphi(\mathscr{T}) = \Big(\phi(\ell_1), \phi(\ell_2), \ldots, \phi(\ell_s)\Big).$$

Note that φ is defined w.r.t. the same order $\prec$ as ϕ. Furthermore $\phi(\ell_{s+1})$ is not inserted in φ since it is always equal to n according to Proposition 3.*(ii)*.

Proposition 4 *For all $n \geq 1$, $m \geq 2$ the function φ is a bijection from $\mathfrak{T}_{(m,n)}$ to $\mathfrak{K}_{(m,n)}$.*

PROOF. According to Proposition 3 and definition of $\mathfrak{K}_{(m,n)}$, φ has its values in $\mathfrak{K}_{(m,n)}$. On the other hand let us consider $\mathscr{T}$ and $\mathscr{T}'$ two distinct trees (on the same alphabet) of $\mathfrak{T}_{(m,n)}$, and L and L' the sets of words that label the leaves of these two trees. Let u be the smallest word according to $\prec$ such that $u \in L \cup L'$ and $u \notin L \cap L'$. We assume that $u \in L$. We let $\varphi(\mathscr{T}) = \big(\phi(\ell_1), \phi(\ell_2), \ldots, \phi(\ell_s)\big)$ and $\varphi(\mathscr{T}') = \big(\phi(\ell'_1), \phi(\ell'_2), \ldots, \phi(\ell'_s)\big)$. By definition there exists r such that ℓ_r is the leaf labeled by u, and such that for all $i < r$, $\phi(\ell_i) = \phi(\ell'_i)$. Thus $\phi(\ell_r) < \phi(\ell'_r)$ and φ is injective.

Let us consider a generalized tuple $\mathcal{K} = (k_1, k_2, \ldots, k_s)$. We have to show that we can build an m-ary tree $\mathscr{T}$ associated with it. We consider a prefix order $\prec$. We first give some general consideration on a construction of a tree $\mathscr{T}$ according to an order $\prec$ then will follow the construction of $\mathscr{T}$ according to $\mathcal{K}$.

Let P and L be the sets of words that label respectively the nodes and the leaves of the tree $\mathscr{T}$ during its construction. Moreover let G be the set define such that $G = \{\, ux \in \Sigma^\star \mid u \in N \textit{ and } x \in \Sigma \,\}$. By the completness property we have $|G| = |N|m$. For more convenience we let $C = G \backslash ((L \cup N) \backslash \{\epsilon\})$. Intuitively C denotes the set of the labels of the paths that are not ended by a leaf. It is clear that if $C = \{\emptyset\}$ the tree $\mathscr{T}$ is extended. If $C \neq \{\emptyset\}$ we can add to it a new vertex *according to the order* $\prec$. That is, if we add a node, the set N becomes: $N = N \cup \{\min_\prec(C)\}$, and the sets G and C are redefined from this new set. And if we add a leaf, the set L becomes $L = L \cup \{\min_\prec(C)\}$, and the set C is redefined from this new set.

We can now describe how a tree $\mathscr{T}$ is built from the generalized tuple $\mathcal{K}$. We consider that $k_0 = 1$ and that initially $N = \{\epsilon\}$, $G = \{\, x \mid x \in \Sigma \,\}$, and $L = \emptyset$. We build the tree $\mathscr{T}$ according to $\mathcal{K}$ such that at each step $t \in [\![1, s]\!]$ of the construction we add consecutively $k_t - k_{t-1}$ nodes and one leaf according to the order $\prec$. In order to show the correctness of this construction, at each step t the set C must be different from $\{\emptyset\}$. It is clear that initially $C = G \neq \{\emptyset\}$. At the end of each step t, $|N|$ and $|L|$ are respectively equal to k_t and t. Moreover from the definition of the generalized tuple, we have $k_t \geq \lceil \frac{t}{m-1} \rceil$. Thus $(m-1)k_t \geq (m-1)\lceil \frac{t}{m-1} \rceil$, and we have $mk_t - k_t - t + 1 \geq (m-1)\lceil \frac{t}{m-1} \rceil - t + 1$. Since $(m-1)\lceil \frac{t}{m-1} \rceil \geq t$, we get $(m-1)\lceil \frac{t}{m-1} \rceil - t + 1 \geq 1$. By replacing the value of the cardinals by their notation in $mk_t - k_t - t + 1 > 0$, we obtain $|G| - |L| - |N| + |\{\epsilon\}| > 0$, and $|C| > 0$. Thus the correctness of the construction. Finally if we add to the tree $\mathscr{T}$ a leaf after the s-th step, $\mathscr{T}$ is extended since $|L| = (m-1)|N| + 1$ (Lemma 2.*(ii)*). Therefore φ is surjective, and then bijective. $\square$

Figure 2 illustrates the construction of an extended tree from a generalized 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, these objects are called *Catalan families*. In addition to the gen-

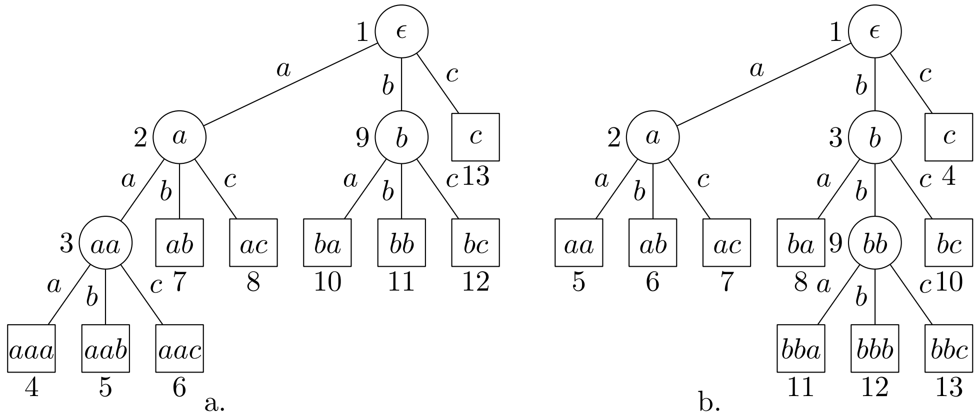


Figure 2. 3-ary trees equivalent to the generalized tuple: $(3, 3, 3, 3, 3, 4, 4, 4)$, according to the lexicographic order (a) or to the graded lexicographic one (b).

eralized tuples we close this section with some of known Catalan families extended to the case of an alphabet of an arbitrary size m.

We define for all $(m, n) \in \mathbb{N}^2$ the *generalized Catalan* numbers [15, 8] 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:

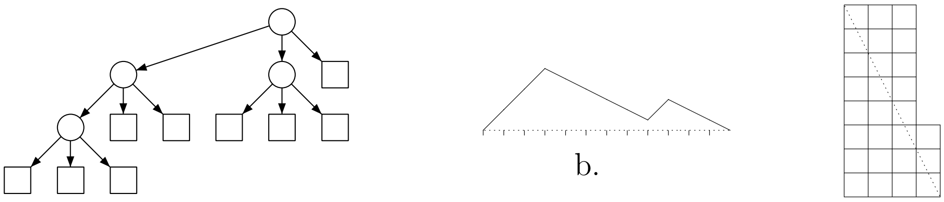
$$S \rightarrow a \underbrace{SbSb \cdots Sb}_{m-1 \ terms \ Sb} S \mid \epsilon$$

These words can also be viewed as sequences $u = x_1 x_2 \cdots x_{n(m-1)+n}$ of 0s and 1s called *well m-balanced sequences* that satisfy the following properties [15]:

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

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

These sequences have been studied in probabilistic mathematics in the general case, and in combinatorics in the case of binary trees ("ballot problem" [6], "Dyck word" [11]). They are in bijection with the *walks above the sea level* that have increasing slopes $m-1$ times greater than the decreasing ones (Figure 3.b). Computer scientists also call them *Dyck paths*. Finally the graphical representation of the generalized tuples (Figure 3.e) 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).



a. b. c.

$$((()))))()))$$
(d)

$$(3,3,3,3,3,4,4,4)$$
(e)

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

3 Relation between complete deterministic automata and extended 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 $\mathfrak{K}_{(m,n)}$ and those of $\mathfrak{D}_{(m,n)}$, and to give some bounds of $|\mathfrak{D}_{(m,n)}|$.

Let $\mathscr{D} =< Q, \Sigma, \delta, i, F >$, $\mathscr{D} \in \mathfrak{D}_{(m,n)}$ be an accessible complete deterministic automaton. We recall that $\Sigma^\star$ is equipped with a total prefix order $\prec$. We associate with each state q of this automaton the word

$$w(q) = \min\nolimits_{\prec} \{\, u \in \Sigma^\star \mid \delta(i,u) = q \textit{ and } u \textit{ is the label of a simple path} \,\}\,.$$

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 of w is canonical. Two distinct complete accessible deterministic automata that are canonically labeled cannot be isomorphic (if the labellings of their states are identical, their transition tables are necessarily different).

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

$$P(\mathscr{D}) = \{\, w(q) \mid q \in Q \,\}\,.$$

Proposition 5 *For all automata $\mathscr{D}$ of $\mathfrak{D}_{(m,n)}$ the set $P(\mathscr{D})$ is prefix.*

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

Prefix sets are in bijection with extended m-ary trees. Thus the transition structures reduced to the set of the smallest paths from the initial state to each one of the DFA states are in bijection with extended m-ary trees.

Theorem 6 *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 $\mathfrak{K}_{(m,n)}$. Each element $\mathcal{K}$ of $\mathfrak{K}_{(m,n)}$ can generate a number of structures equal to*

$$n \times \|\mathcal{K}\| = n \times \|(k_1, \ldots, k_{n(m-1)})\| = n \times \prod_{i=1}^{n(m-1)} k_i.$$

Thus, we have

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

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

Let ℓ be a leaf of the tree, and u be its label. Let p be the parent of ℓ. We consider the edge (p, ℓ) that is labeled by x. The addition of the transition (p, x, r), with $r = \delta(p, x)$ and $r \in N$ to the transition structure $\mathscr{S}$ does not change the labeling of the states of $\mathscr{S}$ if $w(r) \prec u$. The number of different transitions (p, x, 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 ℓ_i, $i \in [\![1, (m-1)n]\!]$. The product of these numbers is multiplied by n, since the last implicit component $k_{(m-1)n+1}$ is equal to n. Hence the expression of the number of transition structures that can be built from a generalized tuple.

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

This result permits to define some bounds on the number of automata of a given size. These bounds are quit different from those obtained by Harrison [7] by considering that two DFAs are isomorphic if by relabeling their alphabet one can make them identical.

Proposition 7 *We have the following inequalities:*

(i) $(2\pi)^{\frac{m-2}{2}} e^{1-s} m^{n+\alpha-1} n^{s-1+\alpha+m/2} \le \frac{\mid \mathfrak{D}_{(m,n)} \mid}{2^n} \le \frac{e}{\sqrt{2\pi}} m^{n+\alpha-1} n^{s-1/2+\alpha}$

$$\textit{with} \quad s = (m-1)n \qquad \alpha = \Big(s + \frac{3}{2}\Big)\Big(1 - \frac{\log(s+1)}{\log(mn)}\Big),$$

(ii) $\sqrt{2}4^n e^{-n} n^n (1+o(1)) \le \frac{\mid \mathfrak{D}_{(2,n)} \mid}{2^n} \le \frac{1}{\sqrt{\pi}} 4^n n^{n-1/2}(1+o(1))$,

(iii) $\frac{\mid \mathfrak{D}_{(m,n)} \mid}{2^n} \le \frac{n^{mn}}{(n-1)!}$.

PROOF. The product of the component of a tuple $\mathcal{K}$ of $\mathfrak{K}_{(m,n)}$ is bounded by

$$(n!)^{(m-1)} \le \|\mathcal{K}\| \le n^{n(m-1)}.$$

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

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

Thanks to some simplifications and using Stirling approximation we get the bounds *(i)* by using the following approximation of the generalized Catalan numbers:

$$\binom{mn+1}{n}\frac{1}{mn+1} = \frac{e}{\sqrt{2\pi}} m^{\alpha+n-1} n^{\alpha-3/2}.$$

In the case of a binary alphabet, the above expression can be approximated and we get the bounds *(ii)* given by Nicaud [13]. 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)* [5]. Notice that a better upper bound, based on accessible DFAs, is presented in [10, 14, 5]. □

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 tuple according to the number of different transition structures that can be deduced from this tuple.

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

In [13], it is shown that n-tuples can be computed via recursive formulae. Following this approach, we define the following generalization of $\mathfrak{K}_{(m,n)}$:

$$\mathfrak{K}_{(m,t,p)} = \left\{ (k_1, k_2, \ldots, k_t) \in [\![1,p]\!]^t \;\middle|\; \forall i \in [\![2,t]\!], k_i \geq \left\lceil \frac{i}{m-1} \right\rceil \textit{ and } k_i \geq k_{i-1} \right\}$$

Notice that for all m and n, an element of $\mathfrak{K}_{(m,n)}$ is an element of $\mathfrak{K}_{(m,n(m-1),n)}$. In the following $\mathcal{K}$ will denote an element of $\mathfrak{K}_{(m,t,p)}$.

We let for all m, t and p

$$c_{(m,t,p)} = \sum_{\mathcal{K}_{(m,t,p)} \in \mathfrak{K}_{(m,t,p)}} \|\mathcal{K}_{(m,t,p)}\|.$$

Proposition 8 *For all $m \geq 2$,$t \geq 1$ and $p \geq 0$, the following relations hold:*

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

PROOF. If $p < \lceil \frac{t}{m-1} \rceil$ then $k_i \leq p < \lceil \frac{t}{m-1} \rceil$, and the condition $k_i \geq \lceil \frac{i}{m-1} \rceil$ cannot be satisfied. If $t = 1$ then $c_{(m,1,p)} = \sum_{i=1}^{p} i = \frac{1}{2}p(p+1)$. Note that $p < \lceil \frac{t}{m-1} \rceil$ and $p \geq 1$ imply $t > 1$.

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

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 the $c_{(m,t,p)}$ 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 $c_{(m,t,p)}$. Table 1 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 structures of transition of size 2 on an alphabet of size 3.

```
Function arrayOfTheC(m : integer, t : integer, p : integer) → array
  var
    T : array [1, t] [0, p] of integer
  Begin
    for j ← 1 to p do
      T[1][j] ← (1/2) j(j + 1)
    od
    for i ← 2 to t do
      for j ← 0 to p do
        if j < ⌈i/(m−1)⌉
          then  T[i][j] ← 0
          else  T[i][j] ← T[i][j − 1] + jT[i − 1][j]
        fi
      od
    od
    return T
  End
```

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

Table 1
The $c_{(3,t,p)}$ elements for $t \in [\![1, 16]\!]$ and $p \in [\![1, 8]\!]$.

$t \backslash p$	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

From the bounds given in Proposition 7 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 store the table becomes quickly huge; for example the table necessary to randomly generate DFAs 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 tuple $\mathcal{K}$ uses the array built by the previous algorithm, and produces a random element of $\mathfrak{K}_{(m,n)}$ according to a probability distribution which reflects the number of transition structures it can generate (Figure 5). It assumes that we have a function *append* which concatenates an integer e to the end of an integer list l and returns the new list: $append(l: \ list, e: \ integer) \ \rightarrow \ list$.

```
1  Function randomElementOfK(m : integer, t : integer, p : integer)→ integer list
2    Begin
3      if p < ⌈t/(m−1)⌉ then return ∅
4      fi
5      if t = 1
6        then
7              Dei ← Random([[1, T[1][p]]])
8              De ← Dei
9              x ← 1
10             while De > x do
11               De ← De − x
12               x ← x + 1
13             od
14             return (x)
15       else
16             De ← Random([[1, T[t][p]]])
17             if (De ≤ T[t][p − 1]) and (p > 1)
18               then  return randomElementOfK(m, t, p − 1)
19               else  return append(randomElementOfK(m, t − 1, p), p)
20             fi
21     fi
22   End
```

Figure 5. Algorithm that randomly generates a generalized tuple according to the number of deterministic structures associated with.

Proposition 9 *The algorithm of Figure 5 randomly builds an element $\mathcal{K}$ of $\mathfrak{K}_{(m,t,p)}$ such that each $\mathcal{K}$ has a probability equal to $\frac{\|\mathcal{K}\|}{c_{(m,t,p)}}$ to be generated.*

PROOF. (Lines 3-4) Since there is no element $\mathcal{K}$ such that $p < \lceil \frac{t}{m-1} \rceil$ the function returns an empty list if it is called with such parameters.

The proof of the sequel of the algorithm is by induction since the function is recursive. We assume that $c_{(m,t,p)} = T[t][p]$ and $m \geq 2$.

(Lines 7-14) Basis, $\mathcal{K} \in \mathfrak{K}_{(m,1,p)}$: the elements $\mathcal{K}$ of $\mathfrak{K}_{(m,1,p)}$ are the lists with a unique element: $(1), (2), \ldots, (p)$. In order to satisfy the property, the algorithm has to determine an integer $x \in [\![1, p]\!]$ such that the probability that x is equal to r ($r \in [\![1, p]\!]$) is equal to $\frac{r}{c_{(m,1,p)}} = \frac{r}{\frac{p(p+1)}{2}}$ (first equality of Proposition 8). Thus in Line 7 we choose a random integer $Dei \in [\![1, c_{(m,1,p)}]\!]$, and at the end of the loop (Lines 10-13), x is the value such that $\frac{x(x-1)}{2} < Dei \leq \frac{x(x+1)}{2}$. Hence the property holds.

(Lines 17,19) Basis, $\mathcal{K} \in \mathfrak{K}_{(m,t,1)}$: necessarily $\mathcal{K}$ ends with 1, thus the condition $p > 1$. Since $c_{(m,t,1)} = 1, \forall t$ such that $1 < \lceil \frac{t}{m-1} \rceil$, the property holds.

(Lines 16-20) Induction step: we assume that the algorithm returns the elements of $\mathfrak{K}_{(m,t-1,p)}$ and the elements of $\mathfrak{K}_{(m,t,p-1)}$ with the property. That is each element $\mathcal{I}$ of $\mathfrak{K}_{(m,t,p-1)}$ is chosen with a probability equal to $\frac{\|\mathcal{I}\|}{c_{(m,t,p-1)}}$, and each element $\mathcal{J}$ of $\mathfrak{K}_{(m,t-1,p)}$ is chosen with a probability equal to $\frac{\|\mathcal{J}\|}{c_{(m,t-1,p)}}$. We thus have to show that each element $\mathcal{K}$ of $\mathfrak{K}_{(m,t,p)}$ is chosen with a probability equal to $\frac{\|\mathcal{K}\|}{c_{(m,t,p)}}$. We let $\mathcal{K} = (k_1, k_2, \ldots, k_t)$ be any element of $\mathcal{K}_{(m,t,p)}$. We distinguish two cases:

(Lines 16-18) Case $\mathcal{K} \in \mathfrak{K}_{(m,t,p-1)}$: the probability that the algorithm returns a $\mathcal{K}$ that belongs to $\mathfrak{K}_{(m,t,p-1)}$ is equal to $\frac{c_{(m,t,p-1)}}{c_{(m,t,p)}}$. Moreover by assumption each element $\mathcal{I}$ of $\mathfrak{K}_{(m,t,p-1)}$ is chosen with a probability equal to $\frac{\|\mathcal{I}\|}{c_{(m,t,p-1)}}$. Thus each element $\mathcal{K}$ of $\mathfrak{K}_{(m,t,p)}$ that belongs to $\mathfrak{K}_{(m,t,p-1)}$ is chosen with a probability equal to $\frac{c_{(m,t,p-1)}}{c_{(m,t,p)}} \times \frac{\|\mathcal{K}\|}{c_{(m,t,p-1)}}$, and the property holds.

(Lines 16-17,19) Case $\mathcal{K} = (k_1, \ldots, k_{t-1}, p)$ and $(k_1, \ldots, k_{t-1}) \in \mathfrak{K}_{(m,t-1,p)}$: the probability that the algorithm returns a $\mathcal{K}$ that ends with p and such that $(k_1, k_2, \ldots, k_{(t-1)}) \in \mathfrak{K}_{(m,t-1,p)}$ is equal to $\frac{c_{(m,t,p)} - c_{(m,t,p-1)}}{c_{(m,t,p)}} = \frac{p \times c_{(m,t-1,p)}}{c_{(m,t,p)}}$ (third equality of Proposition 8). Moreover by assumption each element $\mathcal{J}$ of $\mathfrak{K}_{(m,t-1,p)}$ is chosen with a probability equal to $\frac{\|\mathcal{J}\|}{c_{(m,t-1,p)}}$. Thus each element $\mathcal{K}$ that has such a form is chosen with a probability equal to $\frac{p \times c_{(m,t-1,p)}}{c_{(m,t,p)}} \times \frac{\|(k_1,k_2,\ldots,k_{t-1})\|}{c_{(m,t-1,p)}} = \frac{\|(k_1,k_2,\ldots,k_{t-1})\| \times p}{c_{(m,t,p)}} = \frac{\|\mathcal{K}_{(m,t,p)}\|}{c_{(m,t,p)}}$, and the property holds. $\square$

In order to generate a random $\mathcal{K}$ of $\mathfrak{K}_{(m,n)}$ we call this recursive function as follows: randomElementOfK$(m, n(m-1), n)$. Once an element $\mathcal{K}$ is randomly generated and its associated tree is built according to Proposition 4, one of the DFA associated with this tree can be built according to Theorem 6. This DFA is generated equally likely.

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 6) that accessible complete DFAs are minimal with a probability of 0.8. That is consonant with Nicaud's results.

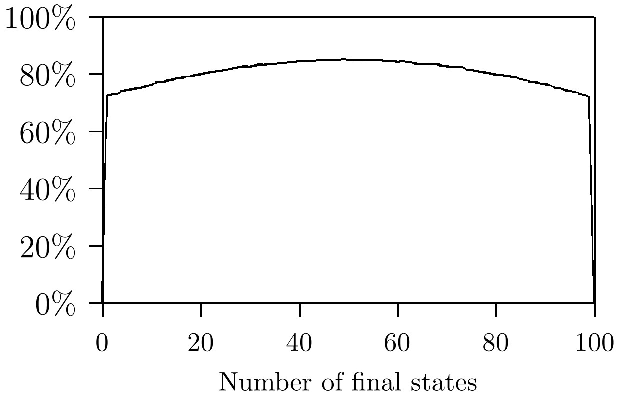


Figure 6. Percentage of complete minimal DFAs of size 100 on an alphabet of size 2, according to the number of final states.

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 7, for DFAs of size 100; notice that it is still valid for DFAs of smaller size.

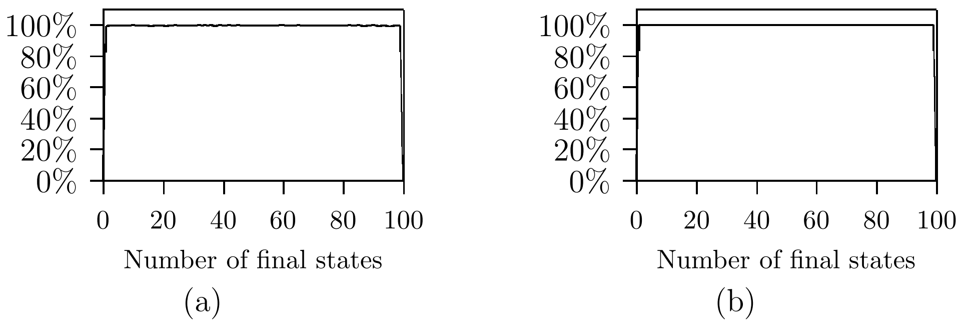


Figure 7. Percentage of complete minimal DFAs of size 100 on an alphabet of size 3 (a) and 5 (b) according to the number of final states.

5 Conclusion

The extension from binary trees to m-ary trees gives rise to a natural generalization of the Catalan families. This generalization allows us to give an algorithm that builds random DFAs on an alphabet of an arbitrary size. 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, as pointed out by our experiments, 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 DFAs. The two empirical observations on the minimality of DFAs are given as conjectures.

Acknowledgements

We want to thank P. Gastin and referees of this paper. The first one for GasTeX, a useful LaTeX package for drawing graphs and automata, and the second ones for their advices that improve this paper and make its readability more deterministic.

References

[1] GMP, gnu multiple precision library. `www.swox.com/gmp/`.

[2] C. Berge. *Théorie des graphes et ses applications*. Dunod Paris, 1967.

[3] J. Berstel and D. Perrin. *Theory of Codes*. Academic Press, 1985.

[4] J.-M. Champarnaud, G. Hansel, T. Paranthoën, and D. Ziadi. Random generation models for nfas. *(To appear in) Journal of Automata, Languages and Combinatorics*, 2004.

[5] M. Domaratzki, D. Kisman, and J. Shallit. On the number of distinct languages accepted by finite automata with n states. *Journal of Automata, Languages and Combinatorics*, 7:67–78, 2002.

[6] W. Feller. *An Introduction to Probability Theory and its Application*. Wiley, 1950.

[7] M. A. Harrison. A census of finite automata. *Canadian Journal of Mathematics*, 17(1):100–113, 1965.

[8] P. Hilton and J. Pedersen. Catalan numbers, their generalization, and their uses. *Math. Intelligencer*, 13(2):64–75, 1991.

[9] D.E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, 1973.

[10] V. A. Liskovets. The number of connected initial automata. *Kibernetika*, 3(5):16–19, 1969.

[11] M. Lothaire. *Combinatorics on Words*. Addison-Wesley, 1983.

[12] C. Nicaud. Average state complexity of operations on unary automata. In M. Kutylowski, L. Pacholski, and T. Wierzbicki, editors, *Lecture Notes in Computer Science, MFCS 1999*, volume 1672, pages 231–240. Springer-Verlag, 1999.

[13] C. Nicaud. *Etude du comportement en moyenne des automates finis et des langages rationnels*. PhD thesis, Université Paris 7, 2000.

[14] R. W. Robinson. Counting strongly connected finite automata. *Graph Theory with Applications to Algorithms and Computer Science*, pages 671–685, 1985.

[15] U. Tamm. Lattice paths not touching a given boundary. *Journal of Statistical Planning and Inference*, 105(2):403–448, 2002.

[16] S. Yu. Regular languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, pages 41–110. Springer-Verlag, 1997.