Enumeration and random generation of accessible automata

Frédérique Bassino [a] and Cyril Nicaud [a]

[a] *Institut Gaspard Monge*
Université de Marne-la-Vallée
77454 Marne-la-Vallée Cedex 2 - France

Abstract

We present a bijection between the set $\mathcal{A}_n$ of deterministic and accessible automata with n states on a k-letters alphabet and some diagrams, which can themselves be represented as partitions of a set of $kn+1$ elements into n non-empty subsets. This combinatorial construction shows that the asymptotic order of the cardinality of $\mathcal{A}_n$ is related to the Stirling number $\left\{ {kn \atop n} \right\}$. Our bijective approach also yields an efficient random sampler of automata with n states for the uniform distribution: its complexity is $\mathcal{O}(n^{3/2})$, using the framework of Boltzmann samplers.

Key words: finite automata, bijections, asymptotic enumeration, random generation, Boltzmann samplers

---

1 Introduction

To any regular language, one can associate in a unique way its minimal automaton, that has the minimal number of states amongst all deterministic automata recognizing this language. Therefore the space complexity of a regular language can be seen as the number of states of its minimal automaton. The worst case complexity of algorithms handling finite automata is most of time known [27]. But the average case analysis of algorithms often requires the enumeration of the objects that are handled [11] and a good knowledge of their combinatorial properties. From a theoretical and practical point of view, a precise enumeration (see [7]) and algorithms of random generation of minimal automata are useful for the study of regular languages.

Email addresses: `bassino@univ-mlv.fr` (Frédérique Bassino), `nicaud@univ-mlv.fr` (Cyril Nicaud).

Preprint submitted to Elsevier Science *14 May 2007*

In this paper we address the problem of the enumeration of the set $\mathcal{A}_n$ of non-isomorphic accessible (also called initially connected) complete and deterministic automata with n states on a k-letters alphabet. These automata are not all minimal, but they contain minimal automata and experimentally, a constant proportion of them seems to be minimal [22,4]. Moreover these automata constitute a very often used representation of regular languages even if they have more states than minimal automata. Empirically again, the minimization of such an automaton provides in average a gain of only one or two states.

The enumeration of finite automata according to various criteria (with or without initial state [17], non-isomorphic [15], up to permutation of the labels of the edges [15], with a strongly connected underlying graph [20,17,24,18], acyclic [21], accessible [19,17,24], ...) is a problem that was studied since 1959 [26]. In particular Korshunov obtained [17] an asymptotic estimate of the cardinality $|\mathcal{A}_n|$ of $\mathcal{A}_n$ by successive estimations of the cardinalities of classes of graphs that approximate the underlying graphs of this class of automata.

In the following, we present a bijection between the set $\mathcal{A}_n$ of deterministic and accessible automata with n states on a k-letters alphabet and some diagrams, which can themselves be represented as partitions of the set $[\![1, (kn+1)]\!]$ into n non-empty parts. Making use of these combinatorial transformations, we establish by a simple, but technical, estimation of the exact enumeration formula [22,4] that $|A_n|$ is $\Theta\left(n2^n \left\{ {kn \atop n} \right\}\right)$, where $\left\{ {kn \atop n} \right\}$ is a number of Stirling of second kind. We also reformulate the asymptotic estimate due to Korshunov [17] in the same terms as the bounds we obtained.

To generate uniformly at random accessible complete and deterministic automata with n states one can use a recursive algorithm [22,4]. But this kind of method, introduced by Nijenhuis and Wilf [23] and systematized by Flajolet, Zimmermann and Van Custem [13], requires an important memory space. In this paper we present an algorithm, based on Boltzmann samplers [8], for the uniform random generation of the elements of $\mathcal{A}_n$ that runs in $\mathcal{O}(n^{3/2})$ time complexity with almost no precalculus.

The paper is organized as follows. In Section 2 we present a bijection between the set $\mathcal{A}_n$ of deterministic and accessible automata with n states on a k-letters alphabet and some diagrams that can easily be defined recursively. These diagrams can themselves be represented as partitions of a set of $kn+1$ elements into n non-empty subsets. The corresponding bijection is given in Section 3. This combinatorial construction shows that the asymptotic order of the cardinality of $\mathcal{A}_n$ is related to the Stirling number $\left\{ {kn \atop n} \right\}$ (see Section 4). Our bijective approach also yields an efficient random sampler of automata with n states, of complexity $\mathcal{O}(n^{3/2})$, using the framework of Boltzmann samplers (see Section 5).

A preliminary version of this work has been presented in [1].

2 Bijective construction of accessible automata

For every $n, m \in \mathbb{N}$ with $n \geq m$, we denote by $[\![m, n]\!]$ the set of integers $\{i \in \mathbb{N} \mid m \leq i \leq n\}$.

First recall some definitions about finite automata. Basic elements of theory of finite automata can be found in [16,25]. A *deterministic finite automaton* $\mathcal{A}$ over the finite alphabet A is a quintuple $\mathcal{A} = (A, Q, \cdot, q_0, F)$ 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 $Q \times A \mapsto Q$. If $\mathcal{A} = (A, Q, \cdot, q_0, F)$ is a deterministic finite automaton, we extend by morphism its transition function to $Q \times A^* \mapsto Q$. 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 $q_0 \cdot u = q$. A finite automaton $\mathcal{A}$ is *complete* when for each $(q, \alpha) \in Q \times A$, $q \cdot \alpha$ is defined.

Two complete 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 ϕ from Q to Q' such that, $\phi(q_0) = q_0'$, $\phi(F) = F'$ and for each $(q, \alpha) \in Q \times A$, $\phi(q \cdot \alpha) = \phi(q) \cdot \alpha$. Two isomorphic automata only differ by the labels of their states.

Our goal is to count the number $|\mathcal{A}_n|$ of accessible complete and deterministic automata with n states up to isomorphism and to generate these automata at random for the uniform distribution on $\mathcal{A}_n$.

2.1 The set $\mathcal{D}_\mathbf{n}$ of structure automata

We introduce a representation of the elements of $\mathcal{A}_n$, that allows us to enumerate them easily. A *simple path* in a deterministic automaton $\mathcal{A}$ is a path labelled by a word u such that for every prefix v and v' of u such that $v \neq v'$, $q_0 \cdot v \neq q_0 \cdot v'$. In other words, on 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 complete and deterministic finite automaton on the alphabet A and w be the map from Q to A^* defined for every state q of 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 a *base automaton* when $Q \subset A^*$ (the states are labelled by words) and for all $u \in Q$, $w(u) = u$. As two distinct base automata cannot be isomorphic, we can directly work on isomorphism classes using base automata.

The *transition structure* of a base 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. Such structures exactly correspond to 2^n base automata, since the accessibility prevents distinct choices of final sets to form the same automaton.

Lemma 1 *Denote by $\mathcal{D}_n$ the set of all the accessible complete and deterministic transition structures of base automata with n states, then $|\mathcal{A}_n| = 2^n|\mathcal{D}_n|$.*

Note that forbiding 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 at random for the uniform distribution on $\mathcal{D}_n$.

2.2 A first bijection

In the following we establish a bijection between the transition structures of $\mathcal{D}_n$ and pairs of integer sequences represented by boxed diagrams. We basically give an algorithm that performs this operation. This construction is an improvement of the ones given in [22,4] where the complete proof of its validity can be found.

A *diagram* of width m and height n is a sequence $(x_1, \ldots, x_m)$ of weakly increasing nonnegative integers such that $x_m = n$, represented classically 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, in other words $y_i \leq x_i$ (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 such that its first coordinate $(x_1, \ldots, x_{(k-1)n+1})$ is a k-Dyck diagram of size n.

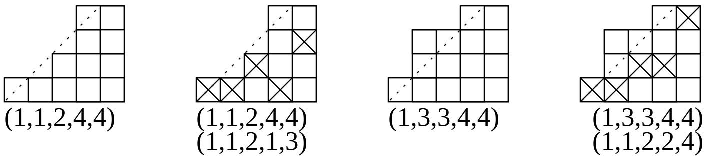


Fig. 1. A diagram of width 5 and height 4, a boxed diagram, a 2-Dyck diagram and a 2-Dyck boxed diagram

Theorem 2 ([22]) *The set $\mathcal{D}_n$ of accessible, complete and deterministic transition structures of size n on a k-letters alphabet is in bijection with the set $\mathcal{B}_n$ of k-Dyck boxed diagrams of size n.*

As a consequence, we get the following exact enumeration formula for $\mathcal{A}_n$ due to Nicaud [22] for two-letters alphabets and generalized to finite alphabets in [4].

Corollary 3 ([22,4]) *For any integer $n \geq 1$, the number $|\mathcal{A}_n|$ of accessible, complete and deterministic non-isomorphic automata of size n on a k-letters alphabet is equal to $2^n|\mathcal{B}_n|$.*

From transition structures to k-Dyck boxed diagrams: we associate to any transition structure $\mathcal{D}$ of size n on a k-letters alphabet, using a depth-first algorithm, a k-Dyck boxed diagram of size n. Starting from q_0, recursively visit for each state q that has not yet been visited, every $q \cdot a$, following the lexicographical order. If $q \cdot a$ has already been visited, store the current number of already visited states and the position of $q \cdot a$ in the prefix order as a part of the result, respectively in the first (Max) and second ($Boxed$) sequences of the boxed diagram.

```
FromDFAtoBoxedDyck(D)
  Max = (); Boxed = ();
  for every q
    Visited[q] = false
    Number[q] = 0 // Number[q] is the position of q in the prefix order
  nbr = 0 // nbr is the number of already visited states
  DepthFirst(D, q0, Max, Boxed, nbr)
  return(Max, Boxed)
```

```
DepthFirst(D, q, Max, Boxed, nbr)
  Visited[q] = true
  nbr = nbr + 1
  Number[q] = nbr
  for each a ∈ A, in the lex. order,
    if (Visited[q · a])
      Append(Max, nbr)
      Append(Boxed, Number[q · a])
    else
      DepthFirst(D, q · a, Max, Boxed, nbr)
```

In the execution of the algorithm FromDFAtoBoxedDyck(), two kinds of transitions are distinguished in the structure: the ones belonging to the covering tree induced by the depth-first algorithm and the other ones producing the integers of the result.

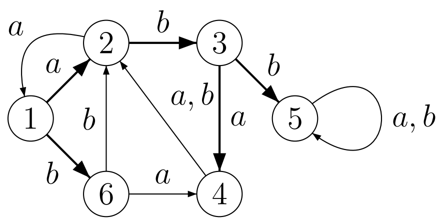


Fig. 2. A transition structure on a 2-letters alphabet having 1 as initial state

In the example given in Fig.2, the states are numbered following the prefix order and the bold edges correspond to the covering tree. Starting from state 1, consider first the transition $1 \cdot a = 2$, and then $2 \cdot a = 1$ that has already been visited. Therefore set $x_1 = 2$, since two states have already been visited and $y_1 = 1$ since $2 \cdot a = 1$. Next, consider the transitions $2 \cdot b = 3$ and $3 \cdot a = 4$. As $4 \cdot a = 2$, set $x_2 = 4$ and $y_2 = 2$, and so on. The result for this transition structure is the 2-Dyck boxed diagram of size 5:

$$((2, 4, 4, 5, 5, 6, 6), (1, 2, 2, 5, 5, 4, 2)).$$

From an accessible complete and deterministic transition structure $\mathcal{D}$ of size n on a k-letters alphabet, the algorithm produces a k-Dyck boxed diagram, since there are kn transitions in $\mathcal{D}$ and $(n-1)$ of them belong to the covering tree of root q_0. The growth condition on the first sequence is due to the fact that the automata is deterministic and complete on a k-letters alphabet.

From k-Dyck boxed diagrams to transition structures: the idea is to reconstruct from any k-Dyck boxed diagram of size n of $\mathcal{B}_n$ its associated transition structure of size n on k-letters alphabet in $\mathcal{D}_n$.

We define a *missing transition* as a transition of the transition structure that has not yet been defined. The algorithm uses a stack S of missing transitions, initialized with all the transitions going from the initial state, put in reverse lexicographical order of their labels. The transition (i, a) where a is then the smallest element of the alphabet is the first one to be selected. The stack S, at any time, contains some missing transitions of the automaton, with respect to the depth-first order. Moreover, when S is empty, the automaton is completely defined.

Two indexes $i \in [\![1, (k-1)n+1]\!]$ and $j \in [\![1, n]\!]$ indicate the current position in the graphical representation of the k-Dyck boxed diagram of size n

$$\mathcal{B} = \Big((x_1, \cdots, x_{(k-1)n}, x_{(k-1)n+1}), (y_1, \cdots, y_{(k-1)n}, y_{(k-1)n+1})\Big).$$

As long as $j < x_i$, the first element (q, a) (q is the state and a the letter of the missing transition) of the stack S is in the covering tree. Therefore the algorithm creates a new state q' and a transition $q \cdot a = q'$; moreover j is incremented by one and all the missing transitions (q', a) are added to the stack, in reverse lexicographical order of their labels.

When $j = x_i$, the first element of the stack is a transition that does not belong to the covering tree, then y_i becomes the image of the top of the stack $q \cdot a$ and i is incremented by one.

The algorithm runs while the stack S is not empty.

In the description of the algorithm `kDickBoxedToTransitionStructure(Max[],Boxed[])`, `Max[]` and `Boxed[]` are two arrays representing respectively the first and second tuple of a k-Dyck boxed diagram.

```
kDickBoxedToTransitionStructure(Max[], Boxed[])
  S = empty stack ; q = 1 // q is the last created state
  Create the initial state 1
  foreach a ∈ A in reversed lex. order
    Push (q, a) into S // add the missing transitions from the initial state
  end foreach
  i = 1; j = 1
  while S is not empty
    (p, a) = Pop from S // take the last pushed missing transition
    if j <Max[i] // creation of a new state
      q = q + 1
      Create a new state q
      Add a transition from p to q labelled by a
      foreach a ∈ A in reversed lex. order
        Push (q, a) into S // add the missing transitions from q
      end foreach
      j = j + 1
    else // directed toward an already existing state
      Add transition from p to Boxed[i] labelled by a
      i = i + 1
    end if
  end while
```

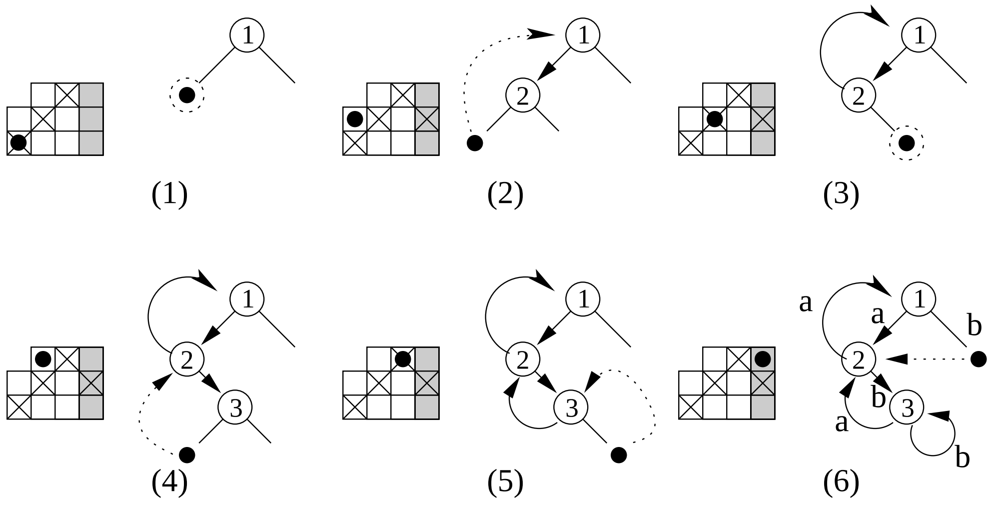


Fig. 3. From a 2-Dyck boxed diagram B' to a transition structure of $\mathcal{D}_n$

Fig. 3 shows an example of the execution of the algorithm on a two-letters alphabet, for

$$\mathcal{B} = ((2,3,3,3),(1,2,3,2)).$$

The grey column corresponds to the last transition. First create the initial state, set $i = j = 1$. At steps (1) and (3): as $j < x_i$ (the dot can go up), create a new state and its missing transitions, j is incremented (the dot goes up). At steps (2) and (4-6): $j = x_i$ (the dot can not go up): the missing transition is directed to the state y_i, and i is incremented (the dot goes right). At the end of step (6), the stack is empty. The algorithm ends.

The complexity of the algorithm kDickBoxedToTransitionStructure() is linear in time and space.

Consequently, the set $\mathcal{D}_n$ of accessible, complete and deterministic transition structures of size n on a k-letters alphabet is in bijection with the set $\mathcal{B}_n$ of k-Dyck boxed diagrams of size n. Moreover for any integer $n \geq 1$, the number $|\mathcal{D}_n|$ of accessible complete and deterministic transition structures of size n on a k-letters alphabet is equal to the number $|\mathcal{B}_n|$ of k-Dyck diagrams of size n and $|\mathcal{A}_n| = 2^n|\mathcal{B}_n|$ as stated in Corollary 3.

3 Representation of set partitions

We describe in this part a bijection between boxed diagrams of width m and height n and set partitions of $n + m$ elements into n non-empty subsets, based on a construction due to Bernardi [2]. This transformation will be used in Section 5 to build a Boltzmann sampler for deterministic and accessible automata. Recall that set partitions are enumerated by Stirling numbers of the second kind (see Section 4).

Proposition 4 *The set $\mathcal{S}_{m,n}$ of boxed diagrams of width m and height n and the set of set partitions of $n + m$ elements into n non-empty subsets are in bijection.*

From a boxed diagram to a set partition: given a boxed diagram of width m and height n, add n boxed columns c_1, c_2, …, c_n. Each c_i is of height i and its highest box is marked. Each column is inserted at the left most position that statisfies the weakly increasing condition. Figure 4 gives an example of such a transformation.

The associated set partition is obtained from the sequence $(y_1, \ldots, y_{m+n})$ of the second coordinates corresponding to the marked boxes: two elements i and j are in the same part if and only if $y_i = y_j$.

From a set partition to a boxed diagram: we now present an algorithm that transforms a set partition $\mathcal{P}$ of a set with $m+n$ elements into n parts into its corresponding boxed diagram

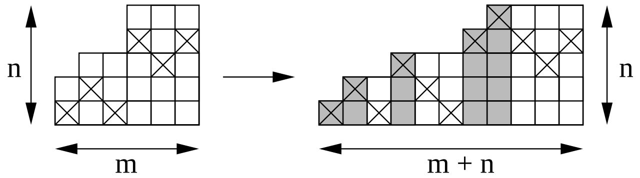


Fig. 4. From a boxed diagram to the set partition $\{\{1,3,6\},\{2,5\},\{4,10\},\{7,9,11\},\{8\}\}$

of width m and height n.

The input of the algorithm is a partition $\mathcal{P}$ given by an array `part`, with indices from 1 to $m+n$ and values in $[\![1,n]\!]$, such that $\texttt{part}[i] = \texttt{part}[j]$ if and only if i and j are in the same part of $\mathcal{P}$ and for every $j \in [\![2, m+n]\!]$ such that $\texttt{part}[j] \geq 2$, there exists $i < j$ such that $\texttt{part}[i] = \texttt{part}[j] - 1$. In other words, the parts of $\mathcal{P}$ are sorted in the order of their smallest element.

For instance, for $m = 3$ and $n = 4$, the partition $\{\{1,3,6\},\{5\},\{2,7\},\{4\}\}$ is represented by the array `part`:

`part`	1	2	1	3	4	1	2

Then, to each i in $[\![1, m+n]\!]$, associate the maximum m_i of $\texttt{part}[j]$ for $j \leq i$ and denote by `max` the new array containing the m_i's. Following with the previous example, we get:

`max`	1	2	2	3	4	4	4
`part`	1	2	1	3	4	1	2

Finally remove the columns with the first occurence of each value in `max`. In the example, we obtain:

2	4	4
1	1	2

The set partition $\{\{1,3,6\},\{2,7\},\{4\},\{5\}\}$ is transformed into the boxed diagram of size 3 $((2,4,4),(1,1,2))$. The complexity in time and space of this algorithm is $\mathcal{O}(n+m)$.

4 Asymptotic order

In this section we give upper and lower bounds of the same order of magnitude for the numbers $|\mathcal{B}_n|$ of k-Dyck boxed diagrams of size n and therefore for $|\mathcal{A}_n|$. More precisely we obtain an upper bound for $|\mathcal{B}_n|$ by counting all boxed diagrams of width $(k-1)n+1$ and height n whose last column is of height n. This overestimation of $|\mathcal{B}_n|$ shows a strong relation between the objects that we enumerate and the Stirling numbers of the second kind. The computation of a lower bound for $|\mathcal{B}_n|$, which is more technical, is based on an overestimation of the contribution to the number of the boxed diagrams that are not k-Dyck boxed diagrams. Next we reformulate a stronger result due to Korshunov [17] in the same terms as the bounds we obtained for $|\mathcal{A}_n|$. Finally we present some numerical results.

The Stirling numbers of the second kind

Recall that the *Stirling number of the second kind*, denoted by $\left\{ {n \atop m} \right\}$, is the number of ways of partitioning a set of n elements into m non-empty subsets. By convention $\left\{ {0 \atop 0} \right\} = 1$, and for $n \geq 1$ we have $\left\{ {n \atop 0} \right\} = 0$. The Stirling numbers of the second kind can be recursively obtained using the following recurrence relation

$$\forall n, m > 0, \quad \left\{ {n \atop m} \right\} = m \left\{ {n-1 \atop m} \right\} + \left\{ {n-1 \atop m-1} \right\}.$$

By induction we obtain the following lemma:

Lemma 5 *For all integer* $0 \leq i \leq n - m$, $\left\{ {n-i \atop m} \right\} \leq \frac{1}{n^i} \left\{ {n \atop m} \right\}$.

The Stirling numbers of the second kind can also be computed from the identity

$$\sum_{n \geq m \geq 0} \left\{ {n \atop m} \right\} \frac{z^n}{n!} = \frac{1}{m!}(e^z - 1)^m$$

or, equivalently, from the sum

$$\left\{ {n \atop m} \right\} = \frac{1}{m!} \sum_{i=0}^{m-1} (-1)^i \binom{m}{i} (m-i)^n.$$

Recall that the LambertW-function [3] is the inverse of the function $x \to xe^x$. 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

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

The Stirling numbers of the second kind $\left\{ {kn \atop n} \right\}$ can be asymptotically estimated with the saddle point method [12]. The following lemma is a special case of the asymptotic expansion obtained by Good [14] for Stirling numbers of the second kind $\left\{ {n \atop m} \right\}$ when n and m tend towards infinity with $n/m = \Theta(1)$.

Lemma 6 *Setting* $\zeta_k = W_0(-ke^{-k}) + k$*, then* $(\zeta_k - k)e^{\zeta_k} = -k$ *and one has*

$$\left\{ {kn \atop n} \right\} = \alpha_k \beta_k^n n^{(k-1)n-1/2}\left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right)$$

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

Remark 7 *When* k *tends towards* $+\infty$, $\zeta_k = k\left(1 - \frac{1}{e^k} + \mathcal{O}\left(\frac{k}{e^{2k}}\right)\right)$, $\alpha_k = \sqrt{\frac{1}{2\pi}}\left(1 - \frac{k}{2e^k} + \mathcal{O}\left(\frac{k^2}{e^{2k}}\right)\right)$ *and* $\beta_k = e\left(1 - \frac{1}{e^k} + \mathcal{O}\left(\frac{k^2}{e^{2k}}\right)\right)$.

4.1 Bounds

In this section, we establish the following result.

Theorem 8 *The number* $|\mathcal{A}_n|$ *of accessible, complete and deterministic automata with* n *states on a* k*-letters alphabet is* $\Theta\left(n\, 2^n \left\{ {kn \atop n} \right\}\right)$.

Recall that from Corollary 3

$$|\mathcal{A}_n| = 2^n |\mathcal{B}_n|$$

where $\mathcal{B}_n$ is the set of k-Dyck boxed diagrams of size n.

Denote $\mathcal{F}^{(k)}_{m,n}$, or $\mathcal{F}_{m,n}$ when there is no ambiguity, the set of boxed diagrams of width m and height n that satisfy the *k-Dyck condition*: for each $i \le m$, $x_i \ge \lceil i/(k-1) \rceil$. As the last column of any k-Dyck boxed diagram of size n is of height n, we uniquely decompose the elements of $\mathcal{B}_n$ into the cartesian product of an element of $\mathcal{F}^{(k)}_{(k-1)n,n}$ and a boxed column of height n. From this elementary decomposition, we obtain

$$|\mathcal{B}_n| = n\, |\mathcal{F}^{(k)}_{(k-1)n,n}|. \tag{1}$$

In the following, we prove that

$$|\mathcal{F}^{(k)}_{(k-1)n,n}| = \Theta\left(\left\{ {kn \atop n} \right\} \right).$$

An upper bound: we obtain an upper bound by relaxing the Dyck condition. In other words, we use the fact that $\mathcal{F}^{(k)}_{(k-1)n,n} \subset \mathcal{S}_{(k-1)n,n}$. As from Proposition 4 the set $\mathcal{S}_{m,n}$ of boxed diagrams of width m and height n is in bijection with the set of set partitions of $n+m$ elements into n non-empty subsets, we get:

Lemma 9 *For any* $n \ge 1$*, one has* $|\mathcal{F}^{(k)}_{(k-1)n,n}| \le \left\{ {kn \atop n} \right\}$.

A lower bound: we now give an asymptotic lower bound, which is of the same order of magnitude as the upper bound, for the numbers $|\mathcal{F}_{(k-1)n,n}|$.

Notations are the ones introduced in Lemma 6. Recall that if $|z| < 1$, the *polylogarithm* function is defined as $\mathrm{polylog}(s, z) = \sum_{i=1}^{\infty} z^i/i^s$. In the following we establish that

Proposition 10 *For all n large enough, one has the inequality*

$$|\mathcal{F}^{(k)}_{(k-1)n,n}| \geq C_k \left\{ {kn \atop n} \right\}$$

with $C_k = 1 - \sqrt{\frac{k-1}{2\pi k}}\, \mathrm{polylog}\left(\frac{1}{2}, \mu_k\right) + \mathcal{O}\left(\frac{1}{\sqrt[3]{n}}\right)$ *and* $\mu_k = \frac{k^k}{e^{k-1}(k-1)^{k-1}\beta_k}$.

Remark 11 *Note that μ_k is a decreasing function of k whose first values are $\mu_2 \approx 0.647$, $\mu_3 \approx 0.355$ and $\mu_4 \approx 0.177$. Moreover when k tends towards infinity*

$$\mu_k = \frac{k}{e^{k-1}}\left(1 - \frac{1}{2k} + \mathcal{O}\left(\frac{1}{k^2}\right)\right),$$

and $\sqrt[k-1]{\mu_k}$ tends monotonically towards $1/e$.

Noticing that, from Proposition 4,

$$|\mathcal{F}_{(k-1)n,n}| = |\mathcal{S}_{(k-1)n,n}| - |\mathcal{S}_{(k-1)n,n} \setminus \mathcal{F}_{(k-1)n,n}| = \left\{ {kn \atop n} \right\} - |\mathcal{S}_{(k-1)n,n} \setminus \mathcal{F}_{(k-1)n,n}|, \qquad (2)$$

a lower bound can be computed overestimating the cardinality of $\mathcal{S}_{(k-1)n,n} \setminus \mathcal{F}_{(k-1)n,n}$.

We decompose the diagrams of $\mathcal{S}_{(k-1)n,n} \setminus \mathcal{F}_{(k-1)n,n}$ depending upon the smallest index i such that $x_i < \lceil \frac{i}{k-1} \rceil$. As $x_i \geq x_{i-1}$ and $x_{i-1} \geq \lceil \frac{i-1}{k-1} \rceil$, we necessarily get $\lceil \frac{i}{k-1} \rceil > \lceil \frac{i-1}{k-1} \rceil$, thus $i = h(k-1) + 1$ with $1 \leq h \leq n-1$ and $x_i = h$.

To describe the decomposition obtained, we define the set $\mathcal{S}^{(h)}_{m,n}$ of the boxed diagrams of width m and height n whose first column is of height greater or equal to h. Note that $\mathcal{S}^{(1)}_{m,n} = \mathcal{S}_{m,n}$.

Any boxed diagram $\mathcal{S}$ of $\mathcal{S}_{(k-1)n,n} \setminus \mathcal{F}_{(k-1)n,n}$ can then be seen as the cartesian product of a k-Dyck boxed diagram of size h and an element of $\mathcal{S}^{(h)}_{(k-1)(n-h)-1,n}$, as shown on Figure 5.

The cardinality of $\mathcal{S}_{(k-1)n,n} \setminus \mathcal{F}_{(k-1)n,n}$ is then:

$$|\mathcal{S}_{(k-1)n,n} \setminus \mathcal{F}_{(k-1)n,n}| = \sum_{h=1}^{n-1} |\mathcal{B}_h|\, |\mathcal{S}^{(h)}_{(k-1)(n-h)-1,n}|. \qquad (3)$$

For $n, m \geq 1$ and $0 \leq h \leq n$, denote $s^{(h)}_{m,n}$ the cardinality of $|\mathcal{S}^{(h)}_{m,n}|$, $s_{m,n}$ the cardinality of $|\mathcal{S}_{m,n}|$ and $f_{m,n}$ the one of $|\mathcal{F}_{m,n}|$. Using Equations (2), (3) and (1), we then can write

$$f_{(k-1)n,n} = \left\{ {kn \atop n} \right\} - \sum_{h=1}^{n-1} f_{(k-1)h,h}\, h\, s^{(h)}_{(k-1)(n-h)-1,n}. \qquad (4)$$

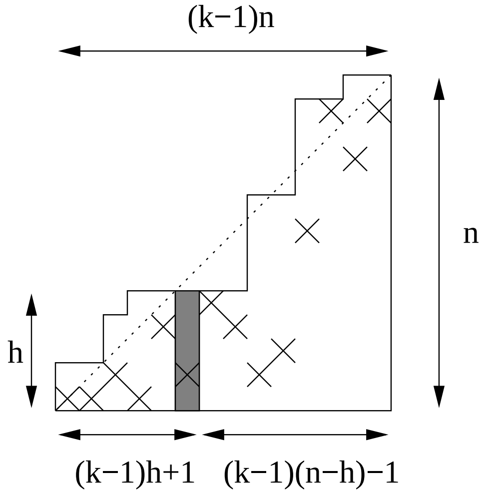


Fig. 5. Representation of the decomposition: the left part is a k-Dyck boxed diagram of size h

In the following, we compute an upper bound for $\sum_{h=1}^{n-1} f_{(k-1)h,h}\, h s^{(h)}_{(k-1)(n-h)-1,n}$, partitioning this summation in three parts, $h \in [\![\, 1, n/e \,]\!], h \in [\![\, n/e, n - \sqrt[3]{n} \,]\!]$ or $h \in [\![\, n - \sqrt[3]{n}, n-1 \,]\!]$. We prove that the contribution of the two first parts is negligible (Lemmas 13 and 15) and that only the third part of the sum has the same order of magnitude as $\left\{ {kn \atop n} \right\}$ (Lemma 16).

Note that $s^{(h)}_{m,n} \leq s_{m,n}$. Moreover the diagrams of width m and height n whose all columns are higher than h are in bijection with the family of combinations with repetitions of size m drawn from a set of $n-h+1$ distinct elements. Therefore there are $\binom{n+m-h}{m}$ such boxed diagrams and we obtain the following bounds for $s^{(h)}_{m,n}$:

Lemma 12 *For all $n, m \geq 1$ and $1 \leq h \leq n$, one has*

$$\binom{n+m-h}{m} h^m \leq s^{(h)}_{m,n} \leq \binom{n+m-h}{m} n^m.$$

We set, for $h \in [\![\, 1, n-1 \,]\!]$, $\Lambda_h = f_{(k-1)h,h}\, h\, s^{(h)}_{(k-1)(n-h)-1,n}$.

Lemma 13 *For all n big enough, $\sum_{h=1}^{n/e} \Lambda_h = O\left(\frac{1}{n} \left\{ {kn \atop n} \right\}\right)$.*

PROOF. We shall estimate $\sum_{h=1}^{n/e} \Lambda_h$ where $\Lambda_h = f_{(k-1)h,h}\, h\, s^{(h)}_{(k-1)(n-h)-1,n}$.

As for all $m, n, h \geq 0$ $s^{(h)}_{m,n} \leq s_{m,n}$ and $f_{m,n} \leq s_{m,n}$, from Proposition 4 we get

$$\Lambda_h \leq h \left\{ {kh \atop h} \right\} \left\{ {kn-(k-1)h-1 \atop n} \right\},$$

and, from Lemma 5, we have for $h \geq 1$,

$$\Lambda_h \leq \frac{1}{n}\left(\frac{h}{n^{(k-1)h}} \left\{ {(k-1)h \atop h} \right\}\right) \left\{ {kn \atop n} \right\}.$$

Moreover, using the asymptotic estimation of $\left\{ {kn \atop n} \right\}$ given in Lemma 6, there exists a positive real number C such that

$$\forall h \geq 1, \quad \frac{h}{n^{(k-1)h}} \left\{ {kh \atop h} \right\} \leq C\sqrt{h}\left(\beta_k \left(\frac{h}{n}\right)^{k-1}\right)^h.$$

As $\beta_k < e$, when $h \leq n/e$, we get $\beta_k \left(\frac{h}{n}\right)^{k-1} \leq \beta_k e^{-(k-1)} < 1$ and

$$\sum_{h=1}^{n/e} \sqrt{h}\left(\beta_k \left(\frac{h}{n}\right)^{k-1}\right)^h \leq \operatorname{polylog}\left(-\frac{1}{2}, \frac{\beta_k}{e^{k-1}}\right).$$

Finally, we obtain

$$\sum_{h=1}^{n/e} \Lambda_h \leq \frac{C}{n} \operatorname{polylog}\left(-\frac{1}{2}, \frac{\beta_k}{e^{k-1}}\right) \left\{ {kn \atop n} \right\},$$

concluding the proof.

Lemma 14 *For every h such that there exist two constants c_1 and c_2 such that $0 < c_1 \leq c_2 < 1$ and, for every n large enough, $c_1 n \leq h \leq c_2 n$, one has $\Lambda_h \leq \Delta_h$ with*

$$\Delta_h = \sqrt{\frac{k-1}{2\pi k}}\left(\frac{k^k}{(k-1)^{k-1}\beta_k}\right)^{n-h} \frac{1}{\sqrt{n-h}}\left(\frac{h}{n}\right)^{(k-1)h+1/2} \left\{ {kn \atop n} \right\} \left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right).$$

PROOF. Recall that $\Lambda_h = f_{(k-1)h,h}\, h s^{(h)}_{(k-1)(n-h)-1,n}$. From Lemma 12, we have

$$s^{(h)}_{(k-1)(n-h)-1,n} \leq \binom{k(n-h)-1}{(k-1)(n-h)-1} n^{(k-1)(n-h)-1}$$

and making use of the Stirling approximation [9, p.54], we get, for $0 \leq h < n$,

$$\binom{k(n-h)-1}{(k-1)(n-h)-1} < \sqrt{\frac{k-1}{2\pi k(n-h)}}\left(\frac{k^k}{(k-1)^{k-1}}\right)^{n-h}.$$

On the other hand, from Proposition 4, we have $f_{(k-1)h,h} \leq \left\{ {kh \atop h} \right\}$, and from Lemma 6, we can write for $0 < c_1 n \leq h \leq c_2 n < n$

$$\left\{ {kh \atop h} \right\} = \left(\frac{1}{\beta_k n^{k-1}}\right)^{n-h}\left(\frac{h}{n}\right)^{(k-1)h-1/2} \left\{ {kn \atop n} \right\} \left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right),$$

and the announced result follows.

Lemma 15 *For all n large enough, one has* $\sum_{h=n/e}^{n-\sqrt[3]{n}} \Delta_h = \mathcal{O}\Big(\frac{1}{n}\left\{ {kn \atop n} \right\}\Big)$.

PROOF. We set

$$\Delta_h = \sqrt{\frac{k-1}{2\pi k}}\, v_h \,\left\{ {kn \atop n} \right\} \Big(1 + O\big(\frac{1}{n}\big)\Big), \tag{5}$$

$$\text{with} \quad v_h = \left(\frac{k^k}{(k-1)^{k-1}\beta_k}\right)^{n-h} \frac{1}{\sqrt{n-h}} \left(\frac{h}{n}\right)^{(k-1)h+1/2}.$$

Recall that $\mu_k = \frac{k^k}{e^{k-1}(k-1)^{k-1}\beta_k}$. In the following we show that

- when $k = 2$ or 3, the sequence (v_h) is decreasing for $h \leq \sqrt[k-1]{\mu_k}n - 1$ and increasing for $h \geq \sqrt[k-1]{\mu_k}n - 1$,
- when $k \geq 4$, the sequence (v_h) is decreasing for $h \leq \sqrt[k-1]{\mu_k}n$ and increasing for $h \geq \sqrt[k-1]{\mu_k}n$.

and that in both cases $\sum_{h=n/e}^{n-\sqrt[3]{n}} v_h = \mathcal{O}\big(\frac{1}{n}\big)$ proving in this way Lemma 15.

By definition,

$$\frac{v_h}{v_{h+1}} = \Big(1 - \frac{1}{h+1}\Big)^{(k-1)(h+1)} \mu_k e^{k-1} \Big(\frac{n}{h}\Big)^{k-1} \sqrt{(1 - \frac{1}{n-h})(1 - \frac{1}{h+1})}.$$

When $k = 2$ or 3, we write

$$\frac{v_h}{v_{h+1}} = \Big(1 - \frac{1}{h+1}\Big)^{(k-1)(h+1)-(k-2)} \mu_k\, e^{k-1} \Big(\frac{n}{h+1}\Big)^{k-1} \sqrt{\frac{(1 - \frac{1}{n-h})}{(1 - \frac{1}{h+1})}},$$

and as $\Big(1 - \frac{1}{h+1}\Big)^{(k-1)(h+1)-(k-2)} < e^{-(k-1)}$, we have

$$\frac{v_h}{v_{h+1}} < \mu_k \Big(\frac{n}{h+1}\Big)^{k-1} \sqrt{\frac{1 - 1/(n-h)}{1 - 1/(h+1)}}.$$

Moreover, when $h \geq \sqrt[k-1]{\mu_k}n - 1$, we have $\mu_k \left(\frac{n}{h+1}\right)^{k-1} < 1$ and

$$\sqrt{\frac{1 - 1/(n-h)}{1 - 1/(h+1)}} < \sqrt{\frac{1 - 1/((1 - \sqrt[k-1]{\mu_k})n + 1)}{1 - 1/\sqrt[k-1]{\mu_k}n}}$$

with

$$\sqrt{\frac{1 - 1/((1 - \sqrt[k-1]{\mu_k})n + 1)}{1 - 1/\sqrt[k-1]{\mu_k}n}} = 1 - \frac{2\sqrt[k-1]{\mu_k} - 1}{2\sqrt[k-1]{\mu_k}(1 - \sqrt[k-1]{\mu_k})} \frac{1}{n} + \mathcal{O}\Big(\frac{1}{n^2}\Big).$$

Note that, from Remark 11, $\mu_2 > 1/2$ and $\mu_3 > 1/4$. Therefore, there exists a positive constant C_1 such that, for all n large enough and $h \geq \sqrt[k-1]{\mu_k}n - 1$, we get $\frac{v_h}{v_{h+1}} < 1 - \frac{C_1}{n} < 1$. Consequently the sequence $(v_h)_{h \geq \sqrt[k-1]{\mu_k}n-1}$ is increasing and

$$\sum_{h=\sqrt[k-1]{\mu_k}n-1}^{n-\sqrt[3]{n}} v_h \leq C_1\, n v_{n-\sqrt[3]{n}}.$$

When $k \geq 4$, as $\left(1 - \frac{1}{h+1}\right)^{(k-1)(h+1)} < e^{-(k-1)}$, we get

$$\frac{v_h}{v_{h+1}} < \mu_k \left(\frac{n}{h}\right)^{k-1} \sqrt{(1 - \frac{1}{n-h})(1 - \frac{1}{h+1})}.$$

Moreover when $h \geq \sqrt[k-1]{\mu_k}n - 1$, we have $\mu_k \left(\frac{n}{h}\right)^{k-1} < 1$ and

$$\sqrt{(1 - \frac{1}{n-h})(1 - \frac{1}{h+1})} < 1 - \frac{1}{3\sqrt[k-1]{\mu_k}(1 - \sqrt[k-1]{\mu_k})}\frac{1}{n},$$

thus the sequence $(v_h)_{h \geq \sqrt[k-1]{\mu_k}n}$ is increasing and we obtain

$$\sum_{h=\sqrt[k-1]{\mu_k}n}^{n-\sqrt[3]{n}} v_h \leq C_0\, n v_{n-\sqrt[3]{n}}.$$

By definition, $v_{n-\sqrt[3]{n}} = \left(1 - n^{-2/3}\right)^{(k-1)(n-\sqrt[3]{n})+1/2} \left(\mu_k e^{k-1}\right)^{\sqrt[3]{n}} n^{-1/6}$, therefore we get

$$v_{n-\sqrt[3]{n}} < \mu_k^{\sqrt[3]{n}} \left(1 + \frac{k-1}{2\sqrt[3]{n}}\right) n^{-1/6}.$$

Thus we obtain

$$\sum_{h=\mu_k n - i}^{n-\sqrt[3]{n}} v_h \leq\ C_i\, n^{5/6} \mu_k^{\sqrt[3]{n}} \left(1 + \frac{k-1}{2\sqrt[3]{n}}\right). \tag{6}$$

where $i \in [\![\, 0, 1 \,]\!]$ depending upon the value of k.

On the other hand,

$$\frac{v_h}{v_{h-1}} = \frac{1}{\mu_k e^{k-1}} \left(\frac{h}{n}\right)^{k-1} \tau(h)$$

where

$$\tau(h) = \left(1 + \frac{1}{h-1}\right)^{(k-1)(h-1)+1/2} \sqrt{\left(1 + \frac{1}{n-h}\right)}.$$

When $k \geq 3$ the two factors of τ are increasing functions of h. Indeed the derivative

$$\Big(1+\frac{1}{h-1}\Big)^{(k-1)(h-1)-1/2}(k-1)\Big(\Big(1+\frac{1}{h-1}\Big)\ln\Big(1+\frac{1}{h-1}\Big)-\frac{1}{h-1}-\frac{1}{2(k-1)(h-1)^2}\Big)$$

of $\Big(1+\frac{1}{h-1}\Big)^{(k-1)(h-1)+1/2}$ is positive for n large enough. When $k \geq 2$ and n large enough, writing

$$\tau(h)=\Big(1+\frac{1}{h-1}\Big)^{h-3/4}\Big(\Big(1+\frac{1}{h-1}\Big)^{1/4}\Big(1+\frac{1}{n-h}\Big)^{1/2}\Big),$$

the function τ is the product of two increasing functions of h on the interval $[n/e,\ \sqrt[k-1]{\mu_k}n-1]$.

When $k=2$ or 3, $n/e \leq h \leq \sqrt[k-1]{\mu_k}n-1$ and n large enough, the function τ is maximal for $h=\sqrt[k-1]{\mu_k}n-1$ and

$$\tau(\sqrt[k-1]{\mu_k}n-1) < e^{k-1}\Big(1+\frac{1}{2n}\frac{1-(k-1)(1-\sqrt[k-1]{\mu_k})}{(1-\sqrt[k-1]{\mu_k})\sqrt[k-1]{\mu_k}}\Big).$$

Moreover as, for $k=2$ or 3,

$$\Big(\frac{h}{n}\Big)^{k-1} \leq \mu_k\Big(1-\frac{1}{\sqrt[k-1]{\mu_k}n}\Big)^{k-1} \leq \mu_k\Big(1-\frac{1}{\sqrt[k-1]{\mu_k}n}\Big),$$

we obtain

$$\frac{v_h}{v_{h-1}} < 1-\frac{1}{2n}\,\frac{(k+1)(1-\sqrt[k-1]{\mu_k})-1}{\sqrt[k-1]{\mu_k}(1-\sqrt[k-1]{\mu_k})} < 1.$$

And as $\sqrt[k-1]{\mu_k} < 1-\frac{1}{k+1}$, we have

$$\sum_{h=n/e}^{\sqrt[k-1]{\mu_k}n-1} v_h \leq C_1'\,n\,v_{n/e}.$$

When $k \geq 4$ and $n/e \leq h \leq \sqrt[k-1]{\mu_k}n$, the function τ is then maximal for $h=\sqrt[k-1]{\mu_k}n$ and

$$\tau(\sqrt[k-1]{\mu_k}n) = e^{k-1}\Big(1-\frac{1}{2n}\frac{(k-1)(1-\sqrt[k-1]{\mu_k})-1}{(1-\sqrt[k-1]{\mu_k})\sqrt[k-1]{\mu_k}}+\mathcal{O}\Big(\frac{1}{n^2}\Big)\Big)$$

Moreover, as $\sqrt[k-1]{\mu_k} < 1-1/(k-1)$, we have

$$\frac{v_h}{v_{h-1}} < 1-\frac{1}{3n}\,\frac{(k-1)(1-\sqrt[k-1]{\mu_k})-1}{\sqrt[k-1]{\mu_k}(1-\sqrt[k-1]{\mu_k})} < 1.$$

Thus

$$\sum_{h=n/e}^{\sqrt[k-1]{\mu_k}n} v_h \leq C'_0\, n\, v_{n/e}.$$

By definition, $v_{n/e} = (e-1)^{-1/2}\left(\left(\frac{1}{\mu_k e^{2(k-1)}}\right)^{1/e} \mu_k e^{k-1}\right)^n n^{-1/2}$, thus we obtain

$$v_{n/e} = (e-1)^{-1/2}\theta^n n^{-1/2}$$

where $\theta < 1$ since $\sqrt[3]{\mu_4} < e^{2/e-1}$ and $\sqrt[k-1]{\mu_k}$ tends monotonically towards $1/e$. Consequently we have

$$\sum_{h=n/e}^{\sqrt[k-1]{\mu_k}n-i} v_h \leq C'_i(e-1)^{-1/2}\, n^{1/2}\, \theta^n \quad \text{with} \quad \theta < 1. \tag{7}$$

where $i \in [\![\, 0, 1 \,]\!]$ depending the value of k.

Finally from Equations (5), (6) and (7), we obtain that $\sum_{h=n/e}^{n-\sqrt[3]{n}} v_h = \mathcal{O}\big(1/n\big)$, concluding the proof.

Lemma 16 *For, all n large enough, one has*

$$\sum_{h=n-\sqrt[3]{n}}^{n-1} \Delta_h = \sqrt{\frac{k-1}{2\pi k}}\, \text{polylog}\left(\frac{1}{2}, \mu_k\right) {kn \brace n} \left(1 + \mathcal{O}\left(\frac{1}{\sqrt[3]{n}}\right)\right).$$

PROOF. Recall that, for $h = \Theta(n)$, one has

$$\Delta_h = \sqrt{\frac{k-1}{2\pi k}}\left(\frac{k^k}{(k-1)^{k-1}\beta_k}\right)^{n-h} \frac{1}{\sqrt{n-h}}\left(\frac{h}{n}\right)^{(k-1)h+1/2} {kn \brace n} \left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right).$$

As, for $m \leq \sqrt[3]{n}$,

$$\left(1 - \frac{m}{n}\right)^{(k-1)(n-m)+1/2} = e^{-(k-1)m}\left(1 + \mathcal{O}\left(\frac{1}{\sqrt[3]{n}}\right)\right),$$

setting $m = n - h$, we get

$$\sum_{m=1}^{\sqrt[3]{n}} \Delta_{n-m} = \sqrt{\frac{k-1}{2\pi k}}\left(\sum_{m=1}^{\sqrt[3]{n}} \left(\frac{k^k}{e^{k-1}(k-1)^{k-1}\beta_k}\right)^m \frac{1}{\sqrt{m}}\right) {kn \brace n} \left(1 + \mathcal{O}\left(\frac{1}{\sqrt[3]{n}}\right)\right),$$

and the result follows.

Then Proposition 10 is a direct consequence of Lemmas 13, 14, 15 and 16.

PROOF. (Proposition 10) From Equation (4), one has

$$f_{(k-1)n,n} = \left\{ {kn \atop n} \right\} - \sum_{h=1}^{n-1} \Lambda_h.$$

Moreover, from Lemmas 14 and 16, for all n large enough, one has

$$\sum_{h=n-\sqrt[3]{n}}^{n-1} \Lambda_h \le (1 - C_k) \left\{ {kn \atop n} \right\} \quad \text{with} \quad C_k = 1 - \sqrt{\frac{k-1}{2\pi k}}\, \text{polylog}\left(\frac{1}{2}, \mu_k\right) + \mathcal{O}\left(\frac{1}{\sqrt[3]{n}}\right).$$

And for n big enough, respectively from Lemma 13 and from Lemmas 14 and 15, both $\sum_{h=1}^{n/e} \Lambda_h$ and $\sum_{h=n/e}^{n-\sqrt[3]{n}} \Lambda_h$ are $O\left(\frac{1}{n} \left\{ {kn \atop n} \right\}\right)$ and therefore are negligible. Thus, we finally obtain

$$f_{(k-1)n,n} \ge C_k \left\{ {kn \atop n} \right\},$$

concluding the proof.

Remark 17 *The constant terms of the lower and upper bounds can be iteratively improved making use of the constant terms already computed. Nevertheless it is not enough to get an asymptotic estimate of $|\mathcal{A}_n|$ when n tends towards infinity.*

4.2 The estimate of Korshunov

We derived from simple bijective constructions the asymptotic order of magnitude of the number of accessible automata, giving a combinatorial interpretation that the asymptotic order is related to the number of set partitions $\left\{ {kn \atop n} \right\}$. Korshunov obtained a more precise result. He gave an asymptotic estimate [17, Theorem 4.8 p.51] of this number. His long proof is based on the estimations, when the number of states tends towards infinity, of cardinalities of classes of graphs that better and better approximate the underlying graphs of this class of automata. A key result [17, Theorem 3.4 p.33] is the estimation of the number of strongly connected graphs.

The link we made between the number of accessible automata and the number of set partitions allows us to reformulate the original estimate of Korshunov in the scale of the Stirling numbers, using their well known asymptotic estimate (see Lemma 6).

Theorem 18 (Korshunov [17,18]) *The number $|\mathcal{A}_n|$ of accessible complete and deterministic automata with n states on a k-letters alphabet satisfies*

$$|\mathcal{A}_n| \sim E_k\, n\, 2^n \left\{ {kn \atop n} \right\} \quad \textit{where} \quad E_k = \frac{1 + \sum_{r=1}^{\infty} \frac{1}{r} \binom{kr}{r-1} \left(e^{k-1}\beta_k\right)^{-r}}{1 + \sum_{r=1}^{\infty} \binom{kr}{r} \left(e^{k-1}\beta_k\right)^{-r}}.$$

PROOF. The statement of the original result of Koshunov [17,18] is the following: the number $|\mathcal{A}_n|$ of accessible complete and deterministic automata with n states over a k-letters alphabet satisfies

$$|\mathcal{A}_n| \sim \left(1 - \frac{ka_k}{1+a_k}\right)^{-1/2} \frac{1 + \sum_{r=1}^{\infty} \frac{1}{r}\binom{kr}{r-1}\Big(e^k \nu(k)\Big)^{-r}}{1 + \sum_{r=1}^{\infty} \binom{kr}{r}\Big(e^k \nu(k)\Big)^{-r}} \frac{2^n\, \nu^n(k)\, n^{kn}}{(n-1)!}, \tag{8}$$

where a_k is the root in $[0,1]$ of the equation $1 + x = xe^{k/(1+x)}$ and

$$\nu(k) = a_k^{a_k}(1+a_k)^{k-1-a_k}.$$

The formula given in Theorem 18 is obtained from Equation (8) using that

$$\zeta_k = \frac{k}{1+a_k} \quad \text{and} \quad a_k = \frac{k}{\zeta_k} - 1 = \frac{k}{\zeta_k} e^{-\zeta_k}.$$

From these equalities we deduce that $\nu(k) = \left(\frac{k}{\zeta_k}\right)^{k-1} e^{\zeta_k - k}$ and $e^k \nu(k) = \beta_k e^{k-1}$. Moreover,

$$\left(1 - \frac{ka_k}{1+a_k}\right)^{-1/2} = \Big(\zeta_k - (k-1)\Big)^{-1/2} = \sqrt{2\pi}\alpha_k.$$

We conclude making use of Stirling's formula for $n!$ and of the asymptotic estimate for the Stirling numbers of the second kind $\left\{ {kn \atop n} \right\}$ mentionned in Lemma 6.

4.3 Numerical results

In the following array, we compare for alphabets of size $k = 2, 3$ and 4 the values of the ratio $\frac{|\mathcal{A}_n|}{2^n\, n \left\{ {kn \atop n} \right\}}$ for $n = 100$, 200, 300 and 400 with

$$E_k = \lim_{n\to+\infty} \frac{|\mathcal{A}_n|}{2^n n \left\{ {kn \atop n} \right\}}.$$

From Theorem 2, one has $|A_n| = n2^n|\mathcal{F}_{(k-1)n,n}|$ and the numbers $|\mathcal{F}_{(k-1)n,n}|$ can be computed making use of the recurrence formula given in Section 5.1 [22,4].

The values of E_k are obtained from the formula given in Theorem 18. Note that E_k quickly converges towards 1, as k tends towards $+\infty$. For instance, $E_{26} \approx 0.99999999987$.

k	100	200	300	400	E_k
2	0.74490782	0.74497737	0.74498956	0.74499374	0.74499902
3	0.87341820	0.87342408	0.87342509	0.87342543	0.87342586
4	0.93931196	0.93931392	0.93931428	0.93931440	0.93931456

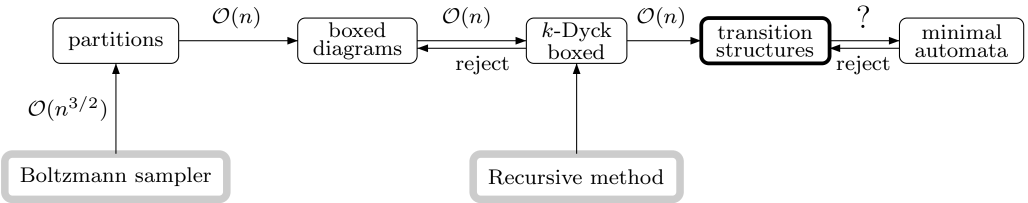
5 Random generation

Our main goal is to provide algorithms to equally likely generate automata of size n. The diagram below describes the differents steps of this generation. Recall that $\mathcal{F}_{m,n}$ is the set of boxed diagrams of width m and height n satisfying the k-Dyck condition, and thus $\mathcal{F}_{(k-1)n,n}$ are obtained by removing the last column from a k-Dyck boxed diagram of size n. In this section, we present two distinct methods to generate elements of $\mathcal{F}_{(k-1)n,n}$. The first one is based on a recursive construction of the elements of $\mathcal{F}_{m,n}$. The other one is based on Boltzmann samplers, a powerful tool introduced in [8]: we generate set partitions that we transform into elements of $\mathcal{F}_{(k-1)n,n}$ using the algorithm of Section 3. A rejection algorithm, whose principle is recalled below, is used to guarantee that the Dyck property is satisfied.

In order to obtain k-Dyck boxed diagrams of size n from boxed diagrams of $\mathcal{F}_{(k-1)n,n}$ it remains to add a column of height n and to randomly choose the box to be marked.

The transformation of k-Dyck diagrams into accessible complete and deterministic transition structures is achieved by the algorithm `kDickBoxedToTransitionStructure(Max[],Boxed[])` of Section 2.2. Finally the random choice of a subset of states of the transition structures produces accesible complete and deterministic automata.

This two random generators of accessible automata can be used to generate minimal automata, using once again a rejection algorithm. Empirically, in average, less than two draws from the set $\mathcal{A}_n$ are enough to obtain a minimal automaton. Nevertheless the efficiency of this rejection algorithm is not yet proved.



Rejection method Suppose that we know how to draw at random, for a given probability distribution, an element of F, that we are able to check whether an element of F is in a subset E of F or not. We want to draw at random an element of the set E, with the probability distribution on E induced by the one on F. A *rejection algorithm* to generate at random an element of E from F is the following.

```
RandomEfromF()        // draw at random an element f of E from F
  repeat
    f = RandomF()       // generate a random element f of F
  until (f ∈ E)
  return(f)        // return a random element f of E from F
```

If p is the probability that an element of F is in E, then, in average, the loop is done $\frac{1}{p}$ times. Moreover if the complexity of the random generation in F is C_F and C_E the one to decide whether an element of F is in E or not, then the average complexity of the algorithm is $\frac{1}{p}C_F C_E$. More detail can be found in [6].

5.1 Random generation of an element of the set $\mathcal{F}_{\mathbf{(k-1)n,n}}$

We give two methods to randomly and equiprobably generate an element F of $\mathcal{F}_{(k-1)n,n}$.

A recursive method

Here we use a simple combinatorial decomposition in order to generate elements of $\mathcal{F}_{m,n}$ at random making use of their enumeration. This kind of recursive method was introduced by Nijenhuis and Wilf [23] and systematized by Flajolet, Zimmermann and Van Custem [13].

The algorithm we describe in the following is due to Nicaud [22] for two-letters alphabets and was generalized to finite alphabets in [4].

Recall that for all positive integers m and n

$$\begin{aligned}\mathcal{F}_{m,n} = \{((x_1,\cdots,x_m),(y_1,\cdots,y_m)) \in [\![1,n]\!]^m \times [\![1,n]\!]^m \mid \text{ for all } i \in [\![2,m]\!], x_i \geq \lceil \frac{i}{k-1} \rceil \\ \text{and } x_i \geq x_{i-1}, \text{ and } \text{ for all } i \in [\![1,m]\!],\, y_i \leq x_i\}.\end{aligned}$$

and that $f_{m,n} = |\mathcal{F}_{m,n}|$.

If $m > 1$ and $n \geq \lceil \frac{m}{k-1} \rceil$, the last element x_m of the first sequence of an element $\mathcal{F} = ((x_1,\cdots,x_m),(y_1,\cdots,y_m))$ of $\mathcal{F}_{m,n}$ is either equal to n and $((x_1,\cdots,x_{m-1}),(y_1,\cdots,y_{m-1}))$ is an element of $\mathcal{F}_{m-1,n}$, or strictly smaller than n and $\mathcal{F} \in \mathcal{F}_{m,n-1}$. From this decomposition and due to the n possible choices for the value of y_m if $x_m = n$, we get the following recurrence formula:

$$\begin{cases} f_{m,n} = 0 & \text{if } n < \lceil \frac{m}{k-1} \rceil \\ f_{m,n} = \frac{1}{2}n(n+1) & \text{if } m = 1 \\ f_{m,n} = nf_{m-1,n} + f_{m,n-1} & \text{otherwise} \end{cases}$$

Therefore we can compute the values of $f_{i,j}$ for $i \in [\![1, (k-1)n]\!]$ and $j \in [\![1, n]\!]$ and store the results in a two-dimensional array. With this precalculus, done once, we easily generate a random element $\mathcal{F} = ((x_1, \cdots, x_{m-1}), (y_1, \cdots, y_{m-1}))$ of $\mathcal{F}_{m,n}$ from right to left, using the decomposition that we just described. When $m > 1$ and $n > \lceil \frac{n}{k-1} \rceil$, to choose whether $x_m = n$ or not, we uniformly draw at random an integer x in $[\![1, f_{m,n}]\!]$ and:

- if $x \leq f_{m,n-1}$, we decide that $x_m < n$ and recursively draw at random $\mathcal{F}$ in $F_{m,n-1}$.
- if $x > f_{m,n-1}$, we set $x_m = n$, y_m is chosen uniformly in $[\![1, n]\!]$, and we recursively choose $((x_1, \cdots, x_{m-1}), (y_1, \cdots, y_{m-1}))$ as a random element of $\mathcal{F}_{m-1,n}$.

This method uses a two-dimensional array of size $(k-1)n \times n$, thus $\mathcal{O}(n^2)$ space. But it stores the values of $f_{m,n}$ which grow exponentially fast (see Section 4). Therefore the bit space used to store these values is $\mathcal{O}(n^3 \log n)$. The generation of the array requires, for the computation of each number, at most one addition and one multiplication by a small number, but as these numbers are big, these operations cannot be done in constant time. Thus the time complexity of the precalculus is $\mathcal{O}(n^3 \log n)$. When the array is stored, the generation of a random element itself is done in time $\mathcal{O}(n^2 \log n)$.

In practice, to make this kind of algorithms more efficient [5], one treats integers as real numbers and approximates them using floating-point arithmetic instead of multi-precision one. This leads to a slight loss of uniformity due to the floating-point approximation. But, in general, this loss is not important and one can choose the precision of the floating-point arithmetics used according to the needs of the computation. Here, with floating point arithmetics, the algorithm uses $\mathcal{O}(n^2)$ space, the precalculus requires $\mathcal{O}(n^2)$ time and the random generation runs in $\mathcal{O}(n)$.

Boltzmann samplers

Duchon, Flajolet, Louchard and Schaeffer [8], introduced a method to build random generators for classes of labelled objects that can be described with a combinatorial decomposition. This generators, Boltzmann samplers, can be obtained directly using automatics rules. Note that a recent paper [10] deals with the unlabelled version of Boltzmann samplers.

A Boltzmann sampler of real parameter $x > 0$, in its exponential version, is a process that produces an object γ of a class $\mathcal{C}$ whose exponential generating function is $C(z) = \sum_{\gamma \in \mathcal{C}} z^{|\gamma|}/|\gamma|!$

with probability
$$\mathbb{P}_x(\gamma) = \frac{1}{C(x)} \frac{x^{|\gamma|}}{|\gamma|!}.$$
Boltmann samplers do not generate objects of a fixed size, but they guarantee that two elements of the same size have the same probability to be generated. Moreover for any given an integer n, the value of x can be chosen such that the average size of the generated elements is n.

The value of x can be computed by solving an equation that involves the exponential generating function of the objects and its derivatives. Floating point arithmetics is required. The evaluation of x is the only precalculus needed.

The behavior of Boltzmann samplers is often such that the size of the generated object is between $(1-\varepsilon)n$ and $(1+\varepsilon)n$ with high probability. Therefore, in most cases, an exact size sampler can be obtained using a rejection algorithm.

We use this technique to uniformly generate random set partitions of a set with kn elements into n non-empty subsets. Following the construction of Section 3 we then transform the set partition obtained into a boxed diagram of $\mathcal{F}_{(k-1)n,n}$ using rejections when the diagram does not satisfy the k-Dyck condition.

In order to uniformly generate set partitions of a set with kn elements into n parts, we first consider the set $\mathcal{P}_n$ of partitions of a set into n non-empty sets seen as n sets of non-empty sets. As the exponential generating function of non-empty sets according to their sizes is $N(z) = e^z - 1$, the generating function of $\mathcal{P}_n$ is $P_n(z) = \frac{(e^z-1)^n}{n!}$, the factor $1/n!$ "kills" the order present in sequences of n sets. Note that every set partition into n non-empty subsets exactly correponds to $n!$ sequences of n sets.

Under the Boltzmann exponential model of parameter x, the probability for a non-empty set to be of size s is $\mathbb{P}_x(|\gamma| = s) = (e^x - 1)^{-1}x^s/s!$. Therefore the size of each of the n sets of the partition follows a Poisson law $\mathrm{Pois}_{\geq 1}$ of parameter x (a truncated Poisson variable K, where K is conditionned to be ≥ 1). This ensures that all resulting objects of the same size have the same probability to be generated. The average size of the partition is then (see [8] Proposition 1):
$$\mathbb{E}_x(\text{size of a partition}) = x\frac{P_n'(x)}{P_n(x)} = nx\frac{e^x}{e^x - 1}.$$
Note that

$$\mathbb{E}_x(\text{size of a partition}) = n\mathbb{E}_x(\text{size of a non-empty set}) = n\frac{N'(x)}{N(x)}.$$

Since we want a partition of size kn, we choose $x = x_n$ such that $nx_n\frac{e^{x_n}}{e^{x_n}-1} = kn$. With notations of Lemma 6, we get $x_n = \zeta_k$. Hence x_n is a constant function of n, only depending upon the size k of the alphabet. The Boltzmann sampler algorithm to uniformly generate a set partition $\mathcal{P}$ of a set of size kn into n non-empty subsets $(E_i)_{1\leq i\leq n}$ is then:

```
BoltzmannSampler(n, k)
   computes the value of ζ_k
   repeat
      for i from 1 to n
         size(E_i) = NonZeroPoissonLaw(ζ_k)
      end for
   until (∑_{i=1}^n size(E_i) = kn)
   return P
```

```
NonZeroPoissonLaw(x)
   k = 1 and p = x(e^x − 1)^{−1}
   dice = uniform([0, 1[)
   while (dice >= p)
      dice = dice − p
      k = k + 1 and p = x ∗ p/k
   end while
   return k
```

To complete the task, the sampler labels the structure obtained with a random permutation of $[\![1, kn]\!]$.

Using floating point approximation, the average cost of the generation of a set partition is $\mathcal{O}(n)$. Testing whether the sum of the sizes of the parts of such a partition is equal to kn or not is also linear.

To compute the average complexity of this algorithm, it remains to estimate the probability for a partition to be of the correct size. Since the exponential generating function of these partitions is $P_n(z)$ and the Boltzmann parameter is equal to ζ_k, the probability for a random partition to be of size kn is (see [8] Eq. 5):

$$\mathbb{P}_{\zeta_k}(N = nk) = \frac{\zeta_k^{kn}[z^{kn}]P_n(z)}{P_n(\zeta_k)} = \frac{\left\{ {kn \atop n} \right\} \zeta_k^{kn}}{(kn)!} \frac{n!}{(e^{\zeta_k} - 1)^n},$$

where $[z^m]C(z)$ is the coefficient of z^m in $C(z)$. Using Lemma 6 and Stirling formula, we obtain the following estimate: $\mathbb{P}_{\zeta_k}(N = nk) \sim \frac{\alpha_k}{\sqrt{kn}}$. Thus, the average number of rejections is $\mathcal{O}(\sqrt{n})$ and the average complexity of the random generation of an element $\mathcal{F}$ of $\mathcal{F}_n$ based on the Boltzmann sampler, using floating point approximation, is $\mathcal{O}(n^{3/2})$.

Open problem To conclude, the estimation of the proportion of minimal automata in $\mathcal{A}_n$ remains an important open problem. We conjecture that a constant proportion of accessible complete and deterministic automata of $\mathcal{A}_n$ is minimal. If it is true, the efficiency of the rejection algorithm to generate minimal automata from accessible complete and deterministic ones would be proved and the asymptotic estimation $\Theta\Big(n2^n \left\{ {kn \atop n} \right\}\Big)$ would also hold for minimal automata.

References

[1] F. Bassino, C. Nicaud, Accessible and Deterministic Automata: Enumeration and Boltzmann Samplers, in International Colloquium on Mathematics and Computer Science 2006, *Discrete Mathematics and Theoretical Computer Science Proceedings*, vol. AG, (2006), 151–160.

[2] O. Bernardi, A note on Stirling numbers, *preprint*.

[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] A. Denise, P. Zimmermann, Uniform random generation of decomposable structures using floating-point arithmetics, *Theoret. Comput. Sci.*, 218 (1999), 233–248.

[6] L. Devroye, *Non-uniform random variate generation*, Springer-Verlag, 1986.

[7] 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.

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

[9] W. Feller, *An introduction to probability theory and its applications*, 3rd edition, vol. 1, Wiley, 1968.

[10] P. Flajolet, E. Fusy, C. Pivoteau, *Boltzmann Sampling of Unlabelled Structures*, Proceedings of ANALCO'07 (Analytic Combinatorics and Algorithms) Conference, New Orleans, January 2007. SIAM Press, in print, 11 pages.

[11] P. Flajolet, R. Sedgewick, *An introduction to the analysis of algorithms*, Addison-Wesley Publishing Company, 1996.

[12] P. Flajolet, R. Sedgewick, *Analytic combinatorics*, Book in preparation, (Version of Febuary 14, 2007 is available at http://www.algo.inria.fr/flajolet/publist.html).

[13] 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.

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

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

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

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

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

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

[20] V.A. Liskovets, Enumeration of non-isomorphic strongly connected automata, *Vesci Akad. Navuk BSSR, Ser. Fiz.-Mat. Navuk* 3 (1971), 26-30, in Russian.

[21] V.A. Liskovets, Exact enumeration of acyclic automata, in *FPSAC'03*, available at http://www.i3s.unice.fr/fpsac/FPSAC03/ARTICLES/5.pdf.

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

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

[24] 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.

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

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

[27] S. Yu, Q. Zhuang and K. Salomaa, The state complexities of some basic operations on regular languages, *Theoret. Comput. Sci.*, 125 (1994), 315–328.