

The average state complexity of the star of a finite set of words is linear

Frédérique Bassino¹, Laura Giambruno², and Cyril Nicaud³.

¹ LIPN UMR CNRS 7030, Université Paris-Nord, 93430 Villetaneuse, France

² Dipartimento di Matematica e Applicazioni, Università di Palermo, 90100, Italy

³ IGM, UMR CNRS 8049, Université Paris-Est, 77454 Marne-la-Vallée, France
bassino@lipn.univ-paris13.fr, lgiambr@math.unipa.it, nicaud@univ-mlv.fr

Abstract. We prove that, for the uniform distribution over all sets X of m (that is a fixed integer) non-empty words whose sum of lengths is n , \mathcal{D}_X , one of the usual deterministic automata recognizing X^* , has on average $\mathcal{O}(n)$ states and that the average state complexity of X^* is $\Theta(n)$. We also show that the average time complexity of the computation of the automaton \mathcal{D}_X is $\mathcal{O}(n \log n)$, when the alphabet is of size at least three.

1 Introduction

This paper addresses the following issue: given a finite set of words X on an alphabet A and a word $u \in A^*$, how to determine efficiently whether $u \in X^*$ or not?

With a non-deterministic automaton, one can determine whether a word u is in X^* or not in time proportional to the product of the lengths of u and X , where the length of X is the sum of the lengths of its elements.

With a deterministic automaton recognizing X^* , one can check whether a word u is in X^* or not in time proportional to the size of u , once the automaton is computed. But in [5], Ellul, Krawetz, Shallit and Wand found an example where the state complexity of X^* , *i.e.* the number of states of the minimal automaton of X^* , is exponential. More precisely, for every integer $h \geq 3$, they gave a language X_h of length $\Theta(h^2)$, containing $\Theta(h)$ words, whose state complexity is $\Theta(h2^h)$. Using another measure on finite sets of words, Campeanu, Culik, Salomaa and Yu proved in [2, 3] that if the set X is a finite language of state complexity $n \geq 4$, the state complexity of X^* is $2^{n-3} + 2^{n-4}$ in the worst case, for an alphabet with at least three letters. Note that the state complexity of X^* is $2^{n-1} + 2^{n-2}$ in the worst case when X is not necessarily finite [14, 15].

An efficient alternative using algorithms related to Aho-Corasick automaton was proposed in [4] by Clément, Duval, Guaiana, Perrin and Rindone. In their paper, an algorithm to compute all the decompositions of a word as a concatenation of elements in a finite set of non-empty words is also given.

This paper is a contribution to this general problem, called the noncommutative Frobenius problem by Shallit [10], from the name of the classical problem [8,

9] of which it is a generalization. Our study is made from an average point of view. We analyse the average state complexity of X^* , for the uniform distribution of sets of m non-empty words, whose sum of lengths is n , and as n tends towards infinity. We use the general framework of analytic combinatorics [6] applied to sets of words and classical automata constructions. Our main result is that, on average, the state complexity of the star of a set X of m non-empty words is linear with respect to the length of X . For an alphabet with at least three letters, we also provide an algorithm to build a deterministic automaton recognizing X^* in average time $\mathcal{O}(n \log n)$, where n is the length of X .

The paper is organized as follows. In Section 2 we recall some definitions, usual automata constructions and combinatorial properties about words. In Section 3 we sketch the proof of the linearity of the average number of states of a deterministic automaton \mathcal{D}_X recognizing X^* . As a consequence of our construction, in Section 4, we prove that the average time complexity for the construction of the automaton \mathcal{D}_X is in $\mathcal{O}(n \log n)$ when the size of the alphabet is at least three. In Section 5, we establish that the average state complexity of the star of a finite set with m non-empty words whose sum of lengths is n is proportional to n . In the case of sets of two words, we prove a stronger result: the average size of the minimal automaton of X^* is equivalent to n . Finally, in Section 6 we give an algorithm to randomly and equiprobably generate sets X of m non-empty words whose sum of lengths is n , and use it to obtain some experimental results about the average number of states of \mathcal{D}_X .

2 Preliminary

2.1 Definitions and constructions

A *finite automaton* \mathcal{A} over a finite alphabet A is a quintuple $\mathcal{A} = (A, Q, T, I, F)$ where Q is a finite set of *states*, $T \subset Q \times A \times Q$ is the set of *transitions*, $I \subset Q$ is the set of *initial states* and $F \subset Q$ is the set of final states. The automaton \mathcal{A} is *deterministic* if it has only one initial state and for any $(p, a) \in Q \times A$ there exists at most one $q \in Q$ such that $(p, a, q) \in T$. It is *complete* if for each $(p, a) \in Q \times A$, there exists at least one $q \in Q$ such that $(p, a, q) \in T$. A deterministic finite automaton \mathcal{A} is *accessible* when for each state q of \mathcal{A} , there exists a path from the initial state to q . The *size* $\#\mathcal{A}$ of an automaton \mathcal{A} is its number of states. The *minimal automaton* of a regular language is the unique smallest accessible and deterministic automaton recognizing this language. The *state complexity* of a regular language is the size of its minimal automaton. We refer the readers to [7, 13, 1] for elements of theory of finite automata.

Any finite automaton $\mathcal{A} = (A, Q, T, I, F)$ can be transformed into a deterministic automaton $\mathcal{B} = (A, \mathcal{P}(Q), T', \{I\}, F')$ recognizing the same language and in which $F' = \{P \in \mathcal{P}(Q) \mid P \cap F \neq \emptyset\}$ and $T' = \{(P, a, R) \text{ with } P \in \mathcal{P}(Q), a \in A \text{ and } R = \{q \mid \exists p \in P, (p, a, q) \in T\}\}$. To be more precise only the accessible part of the automaton \mathcal{B} is really built in this *subset construction*.

Let $X \subset A^*$ be a finite set of words. Denote by $\text{Pr}(X)$ the set of all prefixes of elements of X . The automaton $(A, \text{Pr}(X), T_X, \{\varepsilon\}, X)$, where $T_X = \{(u, a, ua) \mid$

$u \in \text{Pr}(X)$, $a \in A$, $ua \in \text{Pr}(X)\}$, recognizes the set X and the automaton $\mathcal{A}_X = (A, \text{Pr}(X), T_X \cup T, \{\varepsilon\}, X \cup \{\varepsilon\})$, where $T = \{(u, a, a) \mid u \in X, a \in A \cap \text{Pr}(X)\}$ recognizes X^* (see Fig.1). We denote by \mathcal{A}_S the automaton defined for the set of elements of any sequence S by the above construction. In such an automaton only the states labelled by a letter have more than one incoming transition.

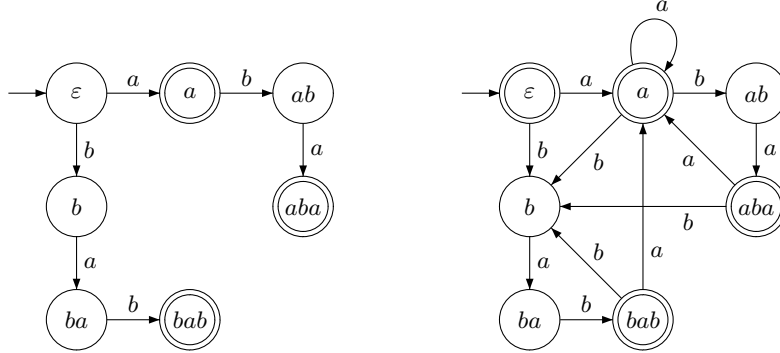


Fig. 1. The automata $(\{a, b\}, \text{Pr}(X), T_X, \{\varepsilon\}, X)$ and \mathcal{A}_X , for $X = \{a, aba, bab\}$

For any finite set of words $X \subset A^*$ (resp. any sequence S), we denote by \mathcal{D}_X (resp. \mathcal{D}_S) the accessible deterministic automaton obtained from the automaton \mathcal{A}_X (resp. \mathcal{A}_S) making use of the subset construction and by \mathcal{M}_X the minimal automaton of X^* .

Lemma 1. *For any finite set of words $X \subset A^*$, the states of the deterministic automaton \mathcal{D}_X recognizing X^* are non-empty subsets $\{u_1, \dots, u_\ell\}$ of $\text{Pr}(X)$ such that for all $i, j \in \{1, \dots, \ell\}$, either u_i is a suffix of u_j or u_j is a suffix of u_i .*

2.2 Enumeration

Let $X \subset A^*$ be a finite set of words. We denote by $|X|$ the cardinality of X and by $\|X\|$ the *length* of X defined as the sum of the lengths of its elements: $\|X\| = \sum_{u \in X} |u|$. Let $\mathcal{S}et_{n,m}$ be the set of sets of m non-empty words whose sum of lengths is n :

$$\mathcal{S}et_{n,m} = \{X = \{u_1, \dots, u_m\} \mid \|X\| = n, \forall i \in \{1, \dots, m\} u_i \in A^+\}$$

and $\mathcal{S}_{n,m}$ be the set of sequences of m non-empty words whose sum of lengths is n :

$$\mathcal{S}_{n,m} = \{S = (u_1, \dots, u_m) \mid \|S\| = n, \forall i \in \{1, \dots, m\} u_i \in A^+\}$$

We denote by $\mathcal{S}_{n,m}^\# \subset \mathcal{S}_{n,m}$ the set of sequences of pairwise distinct words. Recall that $f(n) = \mathcal{O}(g(n))$ if there exists a positive real number c such that for all n big enough $|f(n)| \leq c|g(n)|$.

Proposition 1. *For any fixed integer $m \geq 2$,*

$$|\mathcal{S}_{n,m}| = \binom{n-1}{m-1} |A|^n \quad \text{and} \quad |\mathcal{S}et_{n,m}| = \frac{1}{m!} |\mathcal{S}_{n,m}| \left(1 + \mathcal{O}\left(\frac{1}{n^2}\right)\right).$$

Proof. (sketch) Any sequence S of $\mathcal{S}_{n,m}$ can be uniquely defined by a word v of length n , which is the concatenation of the elements of S , and a composition of n into m parts, that indicates how to cut the word of length n into m parts. Therefore $|\mathcal{S}_{n,m}| = \binom{n-1}{m-1}|A|^n$. Using methods from analytic combinatorics [6], one can prove that

$$|\mathcal{S}_{n,m}^\neq| = |\mathcal{S}_{n,m}| \left(1 + \mathcal{O}\left(\frac{1}{n^2}\right)\right).$$

Furthermore since an element of $\text{Set}_{n,m}$ is mapped to exactly $m!$ sequences of $\mathcal{S}_{n,m}^\neq$, we obtain $|\mathcal{S}_{n,m}^\neq| = m!|\text{Set}_{n,m}|$, concluding the proof. \square

We say that the word v is a *proper prefix* (resp. *suffix*) of a word u if v is a prefix (resp. suffix) of u such that $v \neq \varepsilon$ and $v \neq u$. The word v is called a *border* of u if v is both proper prefix and proper suffix of u . We denote by $\text{Pref}(u)$ (resp. $\text{Suff}(u)$) the set of proper prefixes (resp. suffixes) of u and by $\text{Bord}(u)$ the set of borders of u . A word is *primitive* when it is not the power of another one.

Let u , v and w be three non-empty words such that v is a proper suffix of u and w is a proper suffix of v . We define the three following sets:

$$\begin{aligned} Q_u &= \{\{u\} \cup P \mid P \subset \text{Suff}(u)\} \\ Q_{u,v} &= \{\{u\} \cup P \mid P \in Q_v\} \\ Q_{u,v,w} &= \{\{u\} \cup P \mid P \in Q_{v,w}\}. \end{aligned}$$

Note that the cardinalities of Q_u , $Q_{u,v}$ and $Q_{u,v,w}$ are respectively equal to $2^{|u|-1}$, $2^{|v|-1}$ and $2^{|w|-1}$.

In the proof of the main result (Theorem 1) of this paper, we count the number of states of automata according to their labels. This enumeration is based on the following combinatorial properties of words whose proofs derived from classical results of combinatorics on words (see [11, 12]) are omitted.

Lemma 2. *Let u be a non-empty word of length ℓ . The number of sequences $S \in \mathcal{S}_{n,m}$ such that u is a prefix of a word of S is smaller or equal to $m \binom{n-\ell}{m-1} |A|^{n-\ell}$.*

Lemma 3. *Let $u, v \in A^+$ such that v is not a prefix of u , $|u| = \ell$ and $|v| = i$. The number of sequences $S \in \mathcal{S}_{n,m}$ such that both u and v are prefixes of words of S is smaller or equal to $m(m-1)|A|^{n-\ell-i} \binom{n-\ell-i+1}{m-1}$.*

Lemma 4 ([12] p. 270). *For $1 \leq i < \ell$, there are at most $|A|^{\ell-i}$ pairs of non-empty words (u, v) such that $|u| = \ell$, $|v| = i$ and v is a border of u .*

Lemma 5. *For $1 \leq j < i < \ell$ such that either $i \leq \frac{2}{3}\ell$ or $j \leq \frac{i}{2}$, there are at most $|A|^{\ell-\frac{i}{2}-j}$ triples of non-empty words (u, v, w) with $|u| = \ell$, $|v| = i$, $|w| = j$ such that v is a border of u and w is a border of v .*

Proposition 2. *For $1 \leq j < i < \ell$ such that $i > \frac{2}{3}\ell$ and $j > \frac{i}{2}$ and for any triple of non-empty words (u, v, w) with $|u| = \ell$, $|v| = i$, $|w| = j$ such that v is a border of u and w is a border of v , there exist a primitive word x , with $1 \leq |x| \leq \ell - i$, a prefix x_0 of x and nonnegative integers $p > q > s > 0$ such that $u = x^p x_0$, $v = x^q x_0$ and $w = x^s x_0$.*

3 Main result

In this section we give the proof of the following theorem.

Theorem 1. *For the uniform distribution over the sets X of m (a fixed integer) non-empty words whose sum of lengths is n , the average number of states of the accessible and deterministic automata \mathcal{D}_X recognizing X^* is linear in the length n of X .*

First, note that to prove this result on sets it is sufficient to prove it on sequences:

$$\frac{1}{|\mathcal{S}et_{n,m}|} \sum_{X \in \mathcal{S}et_{n,m}} \#\mathcal{D}_X = \frac{1}{m! |\mathcal{S}et_{n,m}|} \sum_{S \in \mathcal{S}_{n,m}^\#} \#\mathcal{D}_S \leq \frac{1}{m! |\mathcal{S}et_{n,m}|} \sum_{S \in \mathcal{S}_{n,m}} \#\mathcal{D}_S$$

and we conclude using Proposition 1.

Let $Y \subset A^*$ and $S \in \mathcal{S}_{n,m}$, we denote by $\mathfrak{Det}(S, Y)$ the property: Y is the label of a state of \mathcal{D}_S . Let P be a property, the operator $\llbracket \cdot \rrbracket$ is defined by $\llbracket P \rrbracket = 1$ if P is true and 0 otherwise.

To find an upper bound for the average number of states of the deterministic automaton \mathcal{D}_S when the sequence S ranges the set $\mathcal{S}_{n,m}$, we count the states of all automata according to their labels. More precisely we want to estimate the sum

$$\sum_{S \in \mathcal{S}_{n,m}} \#\mathcal{D}_S = \sum_{S \in \mathcal{S}_{n,m}} \sum_{Y \subset A^*} \llbracket \mathfrak{Det}(S, Y) \rrbracket,$$

Taking into account the cardinality of the labels of the states:

$$\sum_{S \in \mathcal{S}_{n,m}} \#\mathcal{D}_S = \sum_{S \in \mathcal{S}_{n,m}} \sum_{|Y|=1} \llbracket \mathfrak{Det}(S, Y) \rrbracket + \sum_{S \in \mathcal{S}_{n,m}} \sum_{|Y| \geq 2} \llbracket \mathfrak{Det}(S, Y) \rrbracket.$$

The first sum deals with states labelled by a single word. Since, for each $S \in \mathcal{S}_{n,m}$, the words that appear in the labels of states of \mathcal{D}_S are prefixes of words of S , we have

$$\sum_{S \in \mathcal{S}_{n,m}} \sum_{|Y|=1} \llbracket \mathfrak{Det}(S, Y) \rrbracket = \sum_{S \in \mathcal{S}_{n,m}} \sum_{\substack{u \text{ prefix of} \\ \text{a word of } S}} \llbracket \mathfrak{Det}(S, \{u\}) \rrbracket \leq (n+1) |\mathcal{S}_{n,m}|.$$

It remains to study the sum

$$\Delta = \sum_{S \in \mathcal{S}_{n,m}} \sum_{|Y| \geq 2} \llbracket \mathfrak{Det}(S, Y) \rrbracket.$$

Let $Y \subset A^*$ be a non-empty set which is not a singleton. By Lemma 1, if Y is the label of a state of an automaton \mathcal{D}_S , then Y belongs to a set $Q_{u,v}$, for some non-empty word u and some proper suffix v of u . Therefore

$$\Delta = \sum_{S \in \mathcal{S}_{n,m}} \sum_{u \in A^+} \sum_{v \in \text{Suff}(u)} \sum_{Y \in Q_{u,v}} \llbracket \mathfrak{Det}(S, Y) \rrbracket.$$

Changing the order of the sums we obtain

$$\Delta = \sum_{u \in A^+} \sum_{v \in \text{Suff}(u)} \sum_{Y \in Q_{u,v}} \sum_{S \in \mathcal{S}_{n,m}} [\mathfrak{Det}(S, Y)].$$

We then partition the sum Δ into $\Delta_1 + \Delta_2$ depending on whether the word v is prefix of u or not:

$$\begin{aligned} \Delta_1 &= \sum_{u \in A^+} \sum_{v \in \text{Bord}(u)} \sum_{Y \in Q_{u,v}} \sum_{S \in \mathcal{S}_{n,m}} [\mathfrak{Det}(S, Y)] \\ \Delta_2 &= \sum_{u \in A^+} \sum_{v \in \text{Suff}(u) \setminus \text{Pref}(u)} \sum_{Y \in Q_{u,v}} \sum_{S \in \mathcal{S}_{n,m}} [\mathfrak{Det}(S, Y)] \end{aligned}$$

To prove Theorem 1, we establish in the following that Δ_1 and Δ_2 are both $\mathcal{O}(n |\mathcal{S}_{n,m}|)$.

3.1 Proof for an alphabet of size at least 3

Let $k \geq 3$ be the cardinality of the alphabet A . Using Lemma 3 we have that

$$\Delta_2 \leq \sum_{u \in A^+} \sum_{v \in \text{Suff}(u) \setminus \text{Pref}(u)} \sum_{Y \in Q_{u,v}} m(m-1)k^{n-|u|-|v|} \binom{n-|u|-|v|+1}{m-1}.$$

As $|Q_{u,v}| = 2^{|v|-1}$, with $\ell = |u|$ and $i = |v|$,

$$\Delta_2 \leq \sum_{\ell=2}^{n-m+1} k^\ell \sum_{i=1}^{\ell-1} 2^{i-1} m(m-1)k^{n-\ell-i} \binom{n-\ell-i+1}{m-1}.$$

Moreover, since $2^i k^{-i} \leq 1$ and since $\sum_{\ell=2}^{n-m+1} \sum_{i=1}^{\ell-1} \binom{n-\ell-i+1}{m-1} = \binom{n-1}{m-1}$,

$$\Delta_2 \leq \frac{m(m-1)}{2} k^n \binom{n-1}{m}$$

and thus, by Proposition 1, $\Delta_2 = \mathcal{O}(n |\mathcal{S}_{n,m}|)$.

Now by Lemma 2, we have

$$\Delta_1 \leq \sum_{u \in A^+} \sum_{v \in \text{Bord}(u)} \sum_{Y \in Q_{u,v}} m \binom{n-|u|}{m-1} k^{n-|u|}.$$

Since $|Q_{u,v}| = 2^{|v|-1}$ we get by Lemma 4

$$\Delta_1 \leq \sum_{\ell=2}^{n-m+1} \sum_{i=1}^{\ell-1} m \binom{n-\ell}{m-1} k^{n-\ell} k^{\ell-i} 2^{i-1}.$$

Since $\sum_{i=1}^{\ell-1} \left(\frac{2}{k}\right)^i \leq \frac{2}{k-2}$, when $k \geq 3$, and $\sum_{\ell=2}^{n-m+1} \binom{n-\ell}{m-1} = \binom{n-1}{m-1}$,

$$\Delta_1 \leq \frac{m}{(k-2)} k^n \binom{n-1}{m}.$$

We use Proposition 1 to conclude that $\Delta_1 = \mathcal{O}(n |\mathcal{S}_{n,m}|)$.

3.2 Proof for an alphabet of size 2

The study of Δ_2 is the same as in the previous section. Now we partition the sum Δ_1 into two sums $\Delta_{1,1}$ and $\Delta_{1,2}$ depending on whether the set Y contains exactly two elements or not (and therefore belongs to some set $Q_{u,v,w}$). More precisely,

$$\Delta_{1,1} = \sum_{u \in A^+} \sum_{v \in \text{Bord}(u)} \sum_{S \in \mathcal{S}_{n,m}} \llbracket \mathfrak{Det}(S, \{u, v\}) \rrbracket$$

and

$$\Delta_{1,2} = \sum_{u \in A^+} \sum_{v \in \text{Bord}(u)} \sum_{w \in \text{Suff}(v)} \sum_{Y \in Q_{u,v,w}} \sum_{S \in \mathcal{S}_{n,m}} \llbracket \mathfrak{Det}(S, Y) \rrbracket.$$

Using Lemmas 2 and 4, and since $\sum_{i=1}^{\ell-1} 2^{-i} \leq 1$ and $\sum_{\ell=2}^{n-m+1} \binom{n-\ell}{m-1} = \binom{n-1}{m-1}$, we obtain

$$\Delta_{1,1} \leq \sum_{\ell=2}^{n-m+1} \sum_{i=1}^{\ell-1} m \binom{n-\ell}{m-1} 2^{n-\ell} 2^{\ell-i} \leq m 2^n \binom{n-1}{m-1}.$$

Consequently, by Proposition 1, $\Delta_{1,1} = \mathcal{O}(n |\mathcal{S}_{n,m}|)$.

Next we decompose the sum $\Delta_{1,2}$ into the sums $B_{1,2} + N_{1,2}$ depending on whether w is a prefix (and therefore a border) of v or not.

When w is not a prefix of v , the number of sequences $S \in \mathcal{S}_{n,m}$ such that u and w are prefixes of two distinct words of S is at most $m(m-1)2^{n-\ell-j} \binom{n-\ell-j+1}{m-1}$ from Lemma 3.

Since, from Lemma 4, there are less than $2^{\ell-i}$ pairs (u, v) such that v is a border of u and since $|Q_{u,v,w}| = 2^{|w|-1}$, we get:

$$\begin{aligned} N_{1,2} &= \sum_{u \in A^+} \sum_{v \in \text{Bord}(u)} \sum_{w \in \text{Suff}(v) \setminus \text{Pref}(v)} \sum_{Y \in Q_{u,v,w}} \sum_{S \in \mathcal{S}_{n,m}} \llbracket \mathfrak{Det}(S, Y) \rrbracket \\ &\leq \sum_{\ell=3}^{n-m+1} \sum_{i=2}^{\ell-1} \sum_{j=1}^{i-1} 2^{\ell-i} 2^{j-1} m(m-1) 2^{n-\ell-j} \binom{n-\ell-j+1}{m-1} \\ &\leq m(m-1) 2^{n-1} \sum_{\ell=3}^{n-m+1} \sum_{i=2}^{\ell-1} 2^{-i} \sum_{j=1}^{i-1} \binom{n-\ell-j+1}{m-1} \end{aligned}$$

As $\binom{n-\ell-j+1}{m-1} \leq \binom{n-\ell}{m-1}$, we obtain

$$N_{1,2} \leq m(m-1) 2^{n-1} \sum_{\ell=3}^{n-m+1} \binom{n-\ell}{m-1} \sum_{i=2}^{\ell-1} (i-1) 2^{-i}$$

Because of the convergence of the series, $\sum_{i=2}^{\ell-1} (i-1) 2^{-i}$ is bounded. Therefore, as $\sum_{\ell=3}^{n-m+1} \binom{n-\ell}{m-1} = \binom{n-2}{m-1}$ and $|\mathcal{S}_{n,m}| = \binom{n-1}{m-1} 2^n$, we have $N_{1,2} = \mathcal{O}(n |\mathcal{S}_{n,m}|)$.

When w is prefix of v , the associated sum $B_{1,2}$ is partitioned into the following sums:

$$B_{1,2} = \sum_{u \in A^+} \sum_{v \in \text{Bord}(u)} \sum_{w \in \text{Bord}(v)} \sum_{Y \in Q_{u,v,w}} \sum_{S \in \mathcal{S}_{n,m}} \llbracket \mathfrak{Det}(S, Y) \rrbracket = B'_{1,2} + B''_{1,2}$$

with

$$B'_{1,2} = \sum_{u \in A^+} \sum_{\substack{v \in \text{Bord}(u) \\ |v| > \frac{2}{3}|u|}} \sum_{w \in \text{Bord}(v)} \sum_{Y \in Q_{u,v,w}} \sum_{S \in \mathcal{S}_{n,m}} [\mathfrak{Det}(S, Y)]$$

and $B''_{1,2} = B_{1,2} \setminus B'_{1,2}$. Using Lemma 5, the fact that $|Q_{u,v,w}| = 2^{|w|-1}$ and relaxing the constraints on the lengths of the words v and w , we get

$$B''_{1,2} \leq \sum_{\ell=3}^{n-m+1} \sum_{i=2}^{\ell-1} \sum_{j=1}^{i-1} m \binom{n-\ell}{m-1} 2^{n-\ell} 2^{\ell-\frac{i}{2}-j} 2^{j-1}.$$

Since $\sum_{i=2}^{\ell-1} (i-1)2^{-\frac{i}{2}}$ is bounded by a constant M ,

$$B''_{1,2} \leq mM2^{n-1} \sum_{\ell=3}^{n-m+1} \binom{n-\ell}{m-1}.$$

Finally as $\sum_{\ell=3}^{n-m+1} \binom{n-\ell}{m-1} = \binom{n-2}{m-2}$ and $|\mathcal{S}_{n,m}| = \binom{n-1}{m-1} 2^n$, $B''_{1,2} = \mathcal{O}(n |\mathcal{S}_{n,m}|)$.

Now from Lemma 2 and since $|Q_{u,v,w}| = 2^{|w|-1}$, we get:

$$B'_{1,2} \leq \sum_{u \in A^+} \sum_{\substack{v \in \text{Bord}(u) \\ |v| > \frac{2}{3}|u|}} \sum_{\substack{w \in \text{Bord}(v) \\ |w| > \frac{|v|}{2}}} 2^{|w|-1} m \binom{n-|u|}{m-1} 2^{n-|u|}.$$

Moreover, from Proposition 2, the words u, v and w of length respectively ℓ, i and j are powers of a same primitive word x : $u = x^p x_0$, $v = x^q x_0$ and $w = x^s x_0$, with $p > q > s > 0$ and $x_0 \in \text{Pr}(x)$. Let r be the length of x , then there are less than 2^r such words x and since $1 \leq r \leq \ell - i$ and $i > \frac{2}{3}\ell$, $r < \frac{\ell}{3}$. Finally the lengths of v and w can be written $i = \ell - hr$ where $1 \leq h < \ell/3r$ and $j = \ell - h'r$ where $h < h' < \frac{1}{2}(\frac{\ell}{r} + h)$, since $j > i/2$. Therefore

$$\begin{aligned} B'_{1,2} &\leq \sum_{\ell=3}^{n-m+1} \sum_{r=1}^{\frac{\ell}{3}-1} \sum_{h=1}^{\frac{\ell}{3r}} \sum_{h'=h+1}^{\frac{1}{2}(\frac{\ell}{r}+h)} m \binom{n-\ell}{m-1} 2^{n-\ell} 2^r 2^{\ell-h'r-1} \\ &\leq m 2^{n-1} \sum_{\ell=3}^{n-m+1} \binom{n-\ell}{m-1} \sum_{r=1}^{\frac{\ell}{3}-1} 2^r \sum_{h=1}^{\frac{\ell}{3r}} \sum_{h'=h+1}^{\frac{1}{2}(\frac{\ell}{r}+h)} (2^{-r})^{h'}. \end{aligned}$$

As $\sum_{h=1}^{\frac{\ell}{3r}} \sum_{h'=h+1}^{\frac{1}{2}(\frac{\ell}{r}+h)} (2^{-r})^{h'} \leq 4/2^{2r}$ when $r \geq 1$, we obtain

$$B'_{1,2} \leq m 2^{n+1} \sum_{\ell=3}^{n-m+1} \binom{n-\ell}{m-1} \sum_{r=1}^{\frac{\ell}{3}-1} 2^{-r} \leq m 2^{n+1} \sum_{\ell=3}^{n-m+1} \binom{n-\ell}{m-1}$$

Finally, since $\sum_{\ell=3}^{n-m+1} \binom{n-\ell}{m-1} = \binom{n-2}{m-2}$ and $|\mathcal{S}_{n,m}| = \binom{n-1}{m-1} 2^n$, we obtain that $B'_{1,2} = \mathcal{O}(n |\mathcal{S}_{n,m}|)$, concluding the proof.

4 Average time complexity of the determinization

The state complexity of a language recognized by a non-deterministic automaton with n states is, in the worst case, equal to 2^n . Therefore the lower bound of the worst-case time complexity of the determinization is $\Omega(2^n)$. In such cases, it is interesting to measure the time complexity according to the size of the output of the algorithm and to try to design algorithms whose efficiency is a function of the size of the result instead of the one of the input. In particular they should be fast when the output is small, even if it is not possible to prevent the output from being of exponential size in the worst case.

The complexity of the subset construction basically depends upon the encoding and the storage of the set of states. At each step, for a given set of states P and a letter $a \in A$, the algorithm computes the set $P \cdot a$ of states of the initial automaton that can be reached from a state of P by a transition labelled by a . Then it tests whether this set has already been computed before or not.

For general non-deterministic automata, the choice of an appropriate data structure for the determinization is not easy. The use of a hashtable may not be an efficient strategy: it is hard to choose the size of the table and the time complexity grows when the table has to be resized and new hashvalues have to be computed for every subset.

Here the automata \mathcal{A}_X to be determinized are specific: for any state u and any letter a , the set $u \cdot a$ can only be \emptyset , $\{a\}$, $\{ua\}$ or $\{a, ua\}$. The sets of states of \mathcal{A}_X can be encoded with lists ordered according to the suffix order, *i.e.* $v \leq_{suff} u$ if and only if $v \in \text{Suff}(u) \cup \{\varepsilon\}$. By Lemma 1, it is a total order over the set of states of \mathcal{D}_X . Hence for any state P of \mathcal{D}_X , which is also a set of states of \mathcal{A}_X , and any letter $a \in A$, the set $P \cdot a$ can be computed in $\mathcal{O}(|P|)$ operations using these data structures. Moreover as the lists are sorted, the comparison of two sets of states P and P' can be done, in the worst case, with $\mathcal{O}(\min\{|P|, |P'|\})$ operations. To store the sets of states of \mathcal{A}_X we use $n+1$ balanced trees $\mathcal{T}_0, \dots, \mathcal{T}_n$ where each tree \mathcal{T}_i contains only subsets of size i . When a new set of states P is computed, it is inserted in the tree $\mathcal{T}_{|P|}$. To check whether the set of states P has already been created it is sufficient to search P in the tree $\mathcal{T}_{|P|}$. These operations can be done with $\mathcal{O}(\log |\mathcal{T}_{|P|}|)$ set comparisons, therefore their time complexity is $\mathcal{O}(|P| \log |\mathcal{T}_{|P|}|)$. As there are at most $\binom{n}{|P|} \leq n^{|P|}$ elements in $\mathcal{T}_{|P|}$, the insertion or the search of a set of states P can be done in $\mathcal{O}(|P|^2 \log n)$ arithmetic operations.

Using this data representation, we can prove the following result whose proof, similar to the proof of Theorem 1, is omitted.

Theorem 2. *For an alphabet of size at least 3, the average time complexity, for the uniform distribution over the sets X of $\text{Set}_{n,m}$, of the construction of the accessible and deterministic automaton \mathcal{D}_X is $\mathcal{O}(n \log n)$.*

The estimation of the time complexity of the determinization of \mathcal{A}_X remains an open problem in the case of a two-letters alphabet.

5 Minimal automata

In Section 3 we have proved that the average number of states of \mathcal{D}_X , for X in $\mathcal{S}et_{n,m}$, is linear in the length of X . The same result holds for the average state complexity of X^* since, for each X in $\mathcal{S}et_{n,m}$, the size of the minimal automaton \mathcal{M}_X of X^* is smaller or equal to the size of \mathcal{D}_X . Moreover, we prove that the average state complexity of X is $\Omega(n)$.

Theorem 3. *For the uniform distribution over the sets X of $\mathcal{S}et_{n,m}$ the average state complexity of X^* is $\Theta(n)$.*

Proof. (sketch) Let $\mathcal{S}_{log} \subset \mathcal{S}_{n,m}$ be the subset of sequences $S = (u_1, \dots, u_m)$ such that for $i \in \{1, \dots, m\}$, $|u_i| > 2 \lfloor \log n \rfloor$ and the prefixes (resp. suffixes) of length $\lfloor \log n \rfloor$ of words in S are pairwise distinct.

For any $S = (u_1, \dots, u_m) \in \mathcal{S}_{log}$, the set $\{u_1, \dots, u_m\}$ is a prefix code. Therefore, making use of a usual construction of the minimal automaton \mathcal{M}_S from the literal automaton of $\{u_1, \dots, u_m\}$ [1, Prop. 2.4], we prove that \mathcal{M}_S has at least $n - 2m \log n$ states.

Next, using asymptotic estimations, we show that the cardinalities of \mathcal{S}_{log} and $\mathcal{S}_{n,m}$ are asymptotically close: $|\mathcal{S}_{n,m}| = |\mathcal{S}_{log}|(1 + o(1))$. Moreover, as $\mathcal{S}_{log} \subset \mathcal{S}_{n,m}^\neq$, we have:

$$\frac{1}{|\mathcal{S}et_{n,m}|} \sum_{X \in \mathcal{S}et_{n,m}} \#\mathcal{M}_X \geq \frac{1}{m!|\mathcal{S}et_{n,m}|} \sum_{S \in \mathcal{S}_{log}} \#\mathcal{M}_S \geq \frac{|\mathcal{S}_{log}|(n - 2m \log n)}{m!|\mathcal{S}et_{n,m}|}$$

Finally we conclude the proof using Proposition 1. \square

Corollary 1. *For the uniform distribution over the sets X of $\mathcal{S}et_{n,m}$, the average number of states of \mathcal{D}_X is $\Theta(n)$.*

Now we study the case $m = 2$ of sets of two non-empty words:

Theorem 4. *For the uniform distribution over the sets X of $\mathcal{S}et_{n,2}$, the average state complexity of X^* is asymptotically equivalent to n .*

Proof. First the proof of Theorem 3 leads to a lower bound asymptotically equivalent to n . Second Kao, Shallit and Xu recently proved [10] that

$$\begin{cases} \#\mathcal{M}_{\{u,v\}} \leq |u| + |v| & \text{if } u, v \in A^+ \text{ are not powers of the same word} \\ \#\mathcal{M}_{\{u,v\}} \leq (|u| + |v|)^2 & \text{otherwise.} \end{cases}$$

Let P_n be the subset of $\mathcal{S}_{n,2}$ containing all sequences (u, v) such that u and v are powers of a same word. For any non-empty word u of size $|u| \leq \frac{n}{2}$ there is at most one word v in A^+ such that $(u, v) \in P_n$. Therefore

$$\sum_{(u,v) \in P_n} \#\mathcal{M}_{\{u,v\}} \leq 2 \sum_{u \in A^+, |u| \leq \frac{n}{2}} n^2 \leq 2n^2 \sum_{i=1}^{\lfloor n/2 \rfloor} |A|^i = \mathcal{O}\left(n^2 |A|^{n/2}\right).$$

Consequently, as $|\mathcal{S}_{n,2}| \sim n|A|^n$ when n tends towards infinity, the contribution of P_n to the average is negligible. And since, for $(u, v) \in \mathcal{S}_{n,2} \setminus P_n$, the size of $\mathcal{M}_{\{u,v\}}$ is lower or equal to n , the average state complexity of X^* is equivalent to n . \square

6 Random generation and experimental results

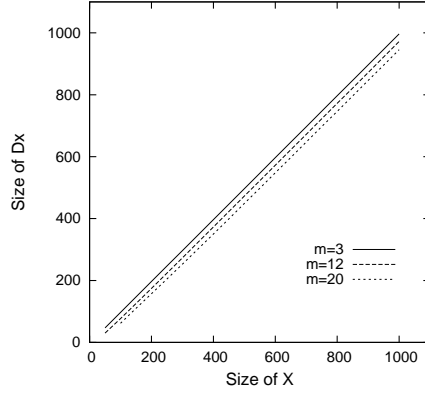


Fig. 2. The average number of states of \mathcal{D}_X for random sets of words $X \in \text{Set}_{n,m}$ on a 3-letters alphabet. For each value of m , 20 points have been computed using 1000 random draws each time.

In the following we explain how to build a random generator for the uniform distribution over the set $\text{Set}_{n,m}$. Recall that each element of $\text{Set}_{n,m}$ corresponds to exactly $m!$ elements of $\mathcal{S}_{n,m}^\neq$. Therefore a uniform random generator for $\mathcal{S}_{n,m}^\neq$ provides a uniform generator for $\text{Set}_{n,m}$.

We use a rejection algorithm to generate elements of $\mathcal{S}_{n,m}^\neq$: we repeatedly generate a random element of $\mathcal{S}_{n,m}$, reject it if it is not in $\mathcal{S}_{n,m}^\neq$, stop if it is in $\mathcal{S}_{n,m}^\neq$. One can show that the average number of elements to be generated is equal to $\frac{1}{p}$, where p is the probability for an element of $\mathcal{S}_{n,m}$ to be in $\mathcal{S}_{n,m}^\neq$, which is $\mathcal{O}(1)$ from Proposition 1.

To draw uniformly at random an element (u_1, \dots, u_m) of $\mathcal{S}_{n,m}^\neq$, we first generate the lengths of the u_i . More precisely a random composition of n into m parts is generated making use of the bijection (see Proposition 1) with the subsets of $\{1, \dots, n-1\}$ of size $m-1$, themselves seen as the $m-1$ first values of a random permutation of $\{1, \dots, n-1\}$. When the lengths of the words are known, each letter is drawn uniformly at random from the alphabet A .

Because of the rejection algorithm, this method may never end, but its average complexity is $\mathcal{O}(n)$. Indeed all algorithms are linear, testing whether the sequence is in $\mathcal{S}_{n,m}^\neq$ is also linear, and the average number of rejects is $\mathcal{O}(1)$. This algorithm has been used to obtain the results shown in Figure 2.

From these experimental results, the average number of states of the deterministic automaton \mathcal{D}_X recognizing X^* seems asymptotically of the form $n - c_m + o(1)$, where c_m is a positive number depending on m .

Acknowledgement: The first and third authors were supported by the ANR (project BLAN07-2_195422).

References

1. J. Berstel and D. Perrin. *Theory of Codes*. Academic Press, 1985.
2. C. Campeanu, K. Culik, K. Salomaa and S. Yu. State complexity of basic operations on finite languages. In *Automata Implementation: 4th International Workshop on Implementing automata (WIA'99)*, Vol. 2214 of *Lectures Notes in Computer Science*, 60–70, 2001.
3. C. Campeanu, K. Salomaa and S. Yu. State complexity of regular languages: finite versus infinite. In C. S. Calude and G. Paun, eds., *Finite Versus Infinite: Contributions to an Eternal Dilemma*, 53–73, Springer, 2000.
4. J. Clément, J.-P. Duval, G. Guaiana, D. Perrin, G. Rindone. Parsing with a finite dictionary. *Theoretical Computer Science*, 340:432–442, 2005.
5. K. Ellul, B. Krawetz, J. Shallit and M.-W. Wang. Regular expressions: new results and open problems. *J. Autom. Lang. Combin.*, 10:407–437, 2005.
6. P. Flajolet, R. Sedgewick. *Analytic combinatorics*, in preparation, (Version of January 2, 2008 is available at <http://www.algo.inria.fr/flajolet/publist.html>).
7. J.E. Hopcroft, J.D. Ullman *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley Publishing Company, 1979.
8. J. L. Ramírez-Alfonsín. Complexity of the Frobenius problem. *Combinatorica*, 16:143–147, 1996.
9. J. L. Ramírez-Alfonsín. *The Diophantine Frobenius Problem*. Oxford University Press, 2005.
10. Jui-Yi Kao, J. Shallit and Zhi Xu. The Frobenius problem in a free monoid. *Symposium on Theoretical Aspects of Computer Science 2008* (Bordeaux), 421–432, www.stacs-cong.org.
11. M. Lothaire. *Combinatorics on words*, Vol 17 of Encyclopedia of mathematics and its applications. Addison-Wesley, 1983.
12. M. Lothaire. *Algebraic combinatorics on words*, Vol 90 of Encyclopedia of mathematics and its applications. Cambridge University Press, 2002.
13. M. Lothaire. *Applied combinatorics on words*, Vol 104 of Encyclopedia of mathematics and its applications. Cambridge University Press, 2005.
14. A. N. Maslov. Estimates of the number of states of finite automata. *Dokl. Akad. Nauk. SSSR*, 194:1266–1268, 1970. (in Russian). English translation in. *Soviet. Math. Dokl.*, 11:1373–1375, 1970.
15. S. Yu, Q. Zhuang and K. Salomaa. The state complexities of some basic operations on regular languages. *Theoretical Computer Science*, 125:315–328, 1994.