Characterisation of an Algebraic Algorithm for Probabilistic Automata

Nathanaël Fijalkow

University of Oxford, United Kingdom; and
LIAFA, Paris 7, France; and
University of Warsaw, Poland
nathanael.fijalkow@cs.ox.ac.uk

Abstract

We consider the value 1 problem for probabilistic automata over finite words: it asks whether a given probabilistic automaton accepts words with probability arbitrarily close to 1. This problem is known to be undecidable. However, different algorithms have been proposed to partially solve it; it has been recently shown that the Markov Monoid algorithm, based on algebra, is the most correct algorithm so far. The first contribution of this paper is to give a characterisation of the Markov Monoid algorithm.

The second contribution is to develop a profinite theory for probabilistic automata, called the prostochastic theory. This new framework gives a topological account of the value 1 problem, which in this context is cast as an emptiness problem. The above characterisation is reformulated using the prostochastic theory, allowing to give a modular proof.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases Probabilistic Automata, Value 1 Problem, Markov Monoid Algorithm, Algebraic Algorithm, Profinite Theory, Topology in Computer Science

Digital Object Identifier 10.4230/LIPIcs.STACS.2016.34

1 Introduction

Rabin [9] introduced the notion of probabilistic automata, which are finite automata with randomised transitions. This powerful model has been widely studied since then and has applications, for instance in image processing, computational biology and speech processing. This paper follows a long line of work that studies the algorithmic properties of probabilistic automata. We consider the value 1 problem: it asks, given a probabilistic automaton, whether there exist words accepted with probability arbitrarily close to 1.

This problem has been shown undecidable [7]. Different approaches led to construct subclasses of probabilistic automata for which the value 1 problem is decidable; the first class was ♯-acylic automata [7], then concurrently simple automata [2] and leaktight automata [4]. It has been shown in [3] that the so-called Markov Monoid algorithm introduced in [4] is the most correct algorithm of the three algorithms. Indeed, both ♯-acylic and simple automata are strictly subsumed by leaktight automata, for which the Markov Monoid algorithm correctly solves the value 1 problem.

Yet we were missing a good understanding of the computations realised by the Markov Monoid algorithm. The aim of this paper is to provide such an insight by giving a characterisation of this algebraic algorithm. We show the existence of *convergence speeds* phenomena, which can be polynomial or exponential. Our main technical contribution is to prove that the Markov Monoid algorithm captures exactly *polynomial behaviours*.

33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016).
Editors: Nicolas Ollinger and Heribert Vollmer; Article No. 34; pp. 34:1–34:13

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany


Proving this characterisation amounts to give precise bounds on convergences of non-homogeneous Markov chains. Our second contribution is to define a new framework allowing to rephrase this characterisation and to give a modular proof for it, using techniques from topological and linear algebra. We develop a profinite approach for probabilistic automata, called prostochastic theory. This is inspired by the profinite approach for (classical) automata [1, 8, 6], and for automata with counters [10].

Section 3 is devoted to defining the Markov Monoid algorithm and stating the characterisation: it answers "YES" if, and only if, the probabilistic automaton accepts some polynomial sequence.

In Section 4, we introduce a new framework, the prostochastic theory, which is used to restate and prove the above characterisation. We first construct a space called the free prostochastic monoid, whose elements are called prostochastic words. We define the acceptance of a prostochastic word by a probabilistic automaton, and show that the value 1 problem can be reformulated as the emptiness problem for probabilistic automata over prostochastic words. We then explain how to construct non-trivial prostochastic words, by defining a limit operator ω, leading to the definition of polynomial prostochastic words. The above characterisation above reads in the realm of prostochastic theory as follows: the Markov Monoid algorithm answers "YES" if, and only if, the probabilistic automaton accepts some polynomial prostochastic word.

Section 5 concludes by showing how this characterisation, combined with an improved undecidability result, supports the claim that the Markov Monoid algorithm is *in some sense* optimal.

2 Probabilistic Automata and the Value 1 Problem

Let Q be a finite set of states.

A distribution over Q is a function $\delta : Q \to [0, 1]$ such that $\sum_{q \in Q} \delta(q) = 1$. We denote $\mathcal{D}(Q)$ the set of distributions over Q, which we often consider as vectors indexed by Q.

For $E \subseteq \mathbb{R}$, we denote $\mathcal{M}_{Q\times Q}(E)$ the set of (square) matrices indexed by Q over E. The space $\mathcal{M}_{Q\times Q}(\mathbb{R})$ is equipped with the norm $||\cdot||$ defined by

$$||M|| = \max_{s \in Q} \sum_{t \in Q} |M(s, t)|.$$

This induces the standard Euclidean topology on $\mathcal{M}_{Q\times Q}(\mathbb{R})$. We denote I the identity matrix. A matrix $M \in \mathcal{M}_{Q\times Q}(\mathbb{R})$ is stochastic if each line is a distribution over Q; the restriction to stochastic matrices is denoted $\mathcal{S}_{Q\times Q}(\mathbb{R})$. The following classical properties will be useful:

▶ **Fact 1** (Topology of the stochastic matrices).

- *For every matrix $M \in \mathcal{S}_{Q\times Q}(\mathbb{R})$, we have $||M|| = 1$,*
- *For every matrices $M, M' \in \mathcal{M}_{Q\times Q}(\mathbb{R})$, we have $||M \cdot M'|| \leq ||M|| \cdot ||M'||$,*
- *The monoid $\mathcal{S}_{Q\times Q}(\mathbb{R})$ is compact.*

▶ **Definition 2** (Probabilistic automaton). A *probabilistic automaton* $\mathcal{A}$ is given by a finite set of states Q, a transition function $\phi : A \to \mathcal{S}_{Q\times Q}(\mathbb{Q})$, an initial state $q_0 \in Q$ and a set of final states $F \subseteq Q$.

Observe that it generalises the definition for classical deterministic automata, in which transitions functions are $\phi : A \to \mathcal{S}_{Q\times Q}(\{0, 1\})$.

A transition function $\phi : A \to \mathcal{S}_{Q\times Q}(\mathbb{Q})$ naturally induces a morphism $\phi : A^* \to \mathcal{S}_{Q\times Q}(\mathbb{Q})$[1].

We denote $P_{\mathcal{A}}(s \xrightarrow{w} t)$ the probability to go from state s to state t reading w on the automaton $\mathcal{A}$, *i.e.* $\phi(w)(s,t)$. We extend the notation: for a subset T of the set of states, $P_{\mathcal{A}}(s \xrightarrow{w} T)$ is defined by $\phi(w)(s,T) = \sum_{t\in T} \phi(w)(s,t)$.

The *acceptance probability* of a word $w \in A^*$ by $\mathcal{A}$ is $P_{\mathcal{A}}(q_0 \xrightarrow{w} F)$, denoted $P_{\mathcal{A}}(w)$. In words, it is the probability that a run starting from the initial state q_0 ends in a final state (*i.e.* a state in F).

The *value* of a probabilistic automaton $\mathcal{A}$ is $\mathrm{val}(\mathcal{A}) = \sup\{P_{\mathcal{A}}(w) \mid w \in A^*\}$, the supremum over all words of the acceptance probability.

▶ **Definition 3** (Value 1 Problem)**.** The value 1 problem is the following decision problem: given a probabilistic automaton $\mathcal{A}$ as input, determine whether $\mathrm{val}(\mathcal{A}) = 1$, *i.e.* whether there exist words whose acceptance probability is arbitrarily close to 1.

Equivalently, the value 1 problem asks for the existence of a sequence of words $(u_n)_{n\in\mathbb{N}}$ such that $\lim_n P_{\mathcal{A}}(u_n) = 1$.

3 Characterisation of the Markov Monoid Algorithm

3.1 Definition

The Markov Monoid algorithm was introduced in [4], we give here a different yet equivalent presentation. Consider $\mathcal{A}$ a probabilistic automaton, the Markov Monoid algorithm consists in computing, by a saturation process, the Markov Monoid of $\mathcal{A}$. It is a monoid of Boolean matrices: all numerical values are projected away to Boolean values.

We give a few definitions and conventions. Throughout the paper, M denotes a matrix in $\mathcal{S}_{Q\times Q}(\mathbb{R})$, and m a Boolean matrix. The following definitions mimick the notions of recurrent and transient states from Markov chain theory.

▶ **Definition 4** (Idempotent Boolean matrix, recurrent and transient state)**.** Consider a matrix $M \in \mathcal{S}_{Q\times Q}(\mathbb{R})$, its Boolean projection $\pi(M)$ is the Boolean matrix such that $\pi(M)(s,t) = 1$ if $M(s,t) > 0$, and $\pi(M)(s,t) = 0$ otherwise.

Let m be a Boolean matrix. It is idempotent if $m \cdot m = m$.

Assume m is idempotent. We say that:

- the state $s \in Q$ is m-recurrent if for all $t \in Q$, if $m(s,t) = 1$, then $m(t,s) = 1$,
- the m-recurrent states $s, t \in Q$ belong to the same recurrence class if $m(s,t) = 1$,
- the state $s \in Q$ is m-transient if it is not m-recurrent.

Since the Markov Monoid only considers Boolean values, one can consider the underlying non-deterministic automaton $\pi(\mathcal{A})$ instead of the probabilistic automaton $\mathcal{A}$. Formally, $\pi(A)$ is defined as $\mathcal{A}$, except that its transitions are given by $\pi(\phi(a))$ for the letter $a \in A$.

The Markov Monoid of $\pi(\mathcal{A})$ contains the transition monoid of $\pi(\mathcal{A})$, which is the monoid generated by $\{\pi(\phi(a)) \mid a \in A\}$ and closed under (Boolean matrix) products. Informally speaking, the transition monoid accounts for the Boolean action of every finite word. Formally, for a word $w \in A^*$, the element $\langle w\rangle$ of the transition monoid of $\pi(\mathcal{A})$ satisfies the following: $\langle w\rangle(s,t) = 1$ if, and only if, there exists a run from s to t reading w on $\pi(\mathcal{A})$.

[1] Note that we use "morphism" for "monoid homomorphism" throughout the paper.

The Markov Monoid extends the transition monoid by introducing a new operator, the stabilisation. On the intuitive level first: let $M \in \mathcal{S}_{Q\times Q}(\mathbb{R})$, it can be interpreted as a Markov chain; its Boolean projection $\pi(M)$ give the structural properties of this Markov chain. The stabilisation $\pi(M)^\sharp$ accounts for $\lim_n M^n$, *i.e.* the behaviour of the Markov chain M in the limit. The formal definition of the stabilisation operator is as follows:

▶ **Definition 5** (Stabilisation)**.** Let m be a Boolean idempotent matrix.
The stabilisation of m is denoted $m^\sharp$ and defined by:

$$m^\sharp(s,t) = \begin{cases} 1 & \text{if } m(s,t) = 1 \text{ and } t \text{ is } m\text{-recurrent,} \\ 0 & \text{otherwise.} \end{cases}$$

The definition of the stabilisation matches the intuition that in the Markov chain $\lim_n M^n$, the probability to be in non-recurrent states converges to 0.

▶ **Definition 6** (Markov Monoid)**.** The Markov Monoid of an automaton $\mathcal{A}$ is the smallest set of Boolean matrices containing $\{\pi(\phi(a)) \mid a \in A\}$ and closed under product and stabilisation of idempotents.

Algorithm 1: The Markov Monoid algorithm.

Data: A probabilistic automaton.
$\mathcal{M} \leftarrow \{\pi(\phi(a)) \mid a \in A\} \cup \{I\}$.
repeat
  if *there is* $m, m' \in \mathcal{M}$ *such that* $m \cdot m' \notin \mathcal{M}$ **then**
    add $m \cdot m'$ to $\mathcal{M}$
  end
  if *there is* $m \in \mathcal{M}$ *such that* m *is idempotent and* $m^\sharp \notin \mathcal{M}$ **then**
    add $m^\sharp$ to $\mathcal{M}$
  end
until *there is nothing to add*;
if *there is a value* 1 *witness in* $\mathcal{M}$ **then**
  return YES;
else
  return NO;
end

On an intuitive level, a Boolean matrix in the Markov Monoid reflects the asymptotic effect of a sequence of finite words.

The Markov Monoid algorithm computes the Markov Monoid, and looks for *value* 1 *witnesses*:

▶ **Definition 7** (Value 1 witness)**.** Let $\mathcal{A}$ be a probabilistic automaton.
A Boolean matrix m is a value 1 witness if: for all states $t \in Q$, if $m(q_0, t) = 1$, then $t \in F$.

The Markov Monoid algorithm answers "YES" if there exists a value 1 witness in the Markov Monoid, and "NO" otherwise.

3.2 An Example

We apply the Markov Monoid algorithm on an example. As explained, the Markov Monoid does not take into account the numerical values of the probabilistic transition, so as input

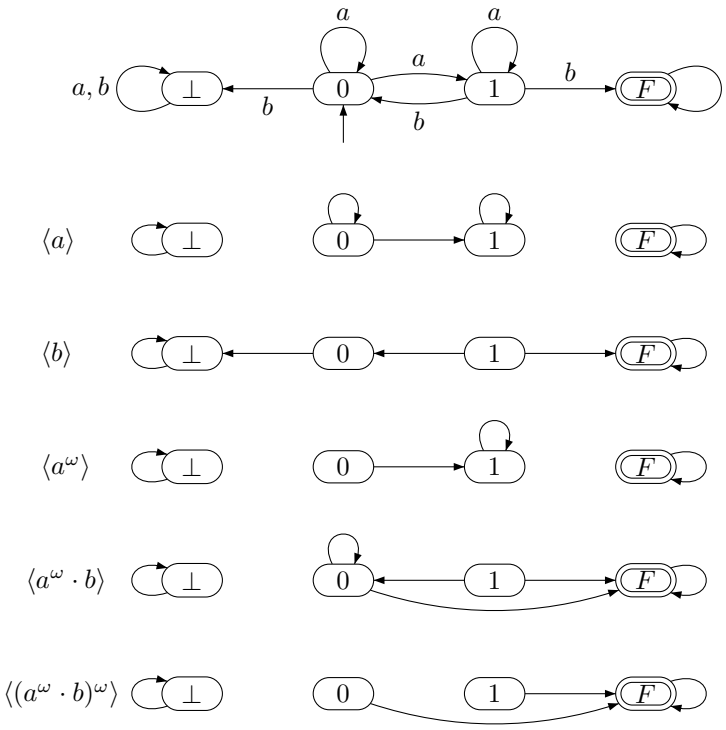

Figure 1 A non-deterministic automaton and part of its Markov Monoid.

we can consider the underlying non-deterministic automaton of a probabilistic automaton. This is what we do in figure 1: the non-deterministic automaton is represented at the top.

We do not represent here all elements of its Markov Monoid; the tool ACME computed it [5], it contains 42 elements. We chose to represent here only 5 elements:

- The first two correspond to the letters a and b.
- The third one is $\langle a^\omega \rangle$, its represents the behaviour of $(a^n)_{n\in\mathbb{N}}$. Indeed, when reading a from the state 0, two events happen with positive probability: looping around the state 0 and going to state 1. So, reading the word a^n from 0 gives a probability to remain in the state 0 converging to 0 when n goes to infinity. Formally, this is reflected by the fact that the state 0 is not $\langle a \rangle$-recurrent.
- The fourth one is $\langle a^\omega \cdot b \rangle$, it illustrates the concatenation between $\langle a^\omega \rangle$ and $\langle b \rangle$. Note that it is not a value 1 witness: from the only initial state 0, we have $\langle a^\omega \cdot b \rangle(0,0) = 1$, and 0 is not final.
- The fifth one is $\langle (a^\omega \cdot b)^\omega \rangle$, it illustrates that stabilisation can be nested. Observe that it is a value 1 witness. This matches the calculation showing that $\lim_n P_{\mathcal{A}}((a^n \cdot b)^n) = 1$, implying that $\mathcal{A}$ has value 1.

3.3 Characterisation

Our main technical result is the following theorem, which is a characterisation of the Markov Monoid algorithm. It relies on the notion of *polynomial sequences of words*.

We define two operations for sequences of words, mimicking the operations of the Markov Monoid:

- the first is concatenation: given $(u_n)_{n\in\mathbb{N}}$ and $(v_n)_{n\in\mathbb{N}}$, the concatenation is the sequence $(u_n \cdot v_n)_{n\in\mathbb{N}}$,
- the second is iteration: given $(u_n)_{n\in\mathbb{N}}$, its iteration is the sequence $(u_n^n)_{n\in\mathbb{N}}$; the n^{th} word is repeated n times.

▶ **Definition 8** (Polynomial sequence)**.** The class of polynomial sequences is the smallest class of sequences containing the constant sequences $(\varepsilon)_{n\in\mathbb{N}}$ and $(a)_{n\in\mathbb{N}}$ for $a \in A$, and closed under concatenation and iteration.

A typical example of a polynomial sequence is $((a^n b)^n)_{n\in\mathbb{N}}$, and a typical example of a sequence which is not polynomial is $\left((a^n b)^{2^n}\right)_{n\in\mathbb{N}}$.

We proceed to our main result:

▶ **Theorem 9** (Characterisation of the Markov Monoid algorithm)**.** *The Markov Monoid algorithm answers "YES" on input $\mathcal{A}$ if, and only if, there exists a polynomial sequence $(u_n)_{n\in\mathbb{N}}$ such that $\lim_n P_{\mathcal{A}}(u_n) = 1$.*

This result could be proved directly, without appealing to the prostochastic theory developed in the next section. The proof relies on technically intricate calculations over non-homogeneous Markov chains; the prostochastic theory allows to simplify its presentation, making it more modular. We will give the proof of Theorem 9 in Subsection 4.4, after restating it using the prostochastic theory.

A second advantage of using the prostochastic theory is to give a more natural and robust definition of polynomial sequences, which in the prostochastic theory correspond to polynomial prostochastic words.

A direct corollary of Theorem 9 is the absence of false negatives:

▶ **Corollary 10** (No false negatives for the Markov Monoid algorithm)**.** *If the Markov Monoid algorithm answers "YES" on input $\mathcal{A}$, then $\mathcal{A}$ has value 1.*

4 The Prostochastic Theory

In this section, we introduce the prostochastic theory, which draws from profinite theory to give a topological account of probabilistic automata. We construct the free prostochastic monoid in Subsection 4.1.

The aim of this theory is to give a topological account of the value 1 problem; we show in Subsection 4.2 that the value 1 problem can be reformulated as an emptiness problem for prostochastic words.

In Subsection 4.3 we define the notion of polynomial prostochastic words.

The characterisation given in Section 3 is stated and proved in this new framework in Subsection 4.4.

4.1 The Free Prostochastic Monoid

The purpose of the prostochastic theory is to construct a compact monoid $\mathcal{P}A^*$ together with a continuous injective morphism $\iota : A^* \to \mathcal{P}A^*$, called the free prostochastic monoid, satisfying the following universal property:

"Every morphism $\phi : A^* \to \mathcal{S}_{Q\times Q}(\mathbb{R})$ extends uniquely
to a continuous morphism $\widehat{\phi} : \mathcal{P}A^* \to \mathcal{S}_{Q\times Q}(\mathbb{R})$."

Here, by "$\widehat{\phi}$ extends ϕ" we mean $\phi = \widehat{\phi} \circ \iota$.

We give two statements about $\mathcal{P}A^*$, the first will be weaker but enough for our purposes in this paper, and the second more precise, and justifying the name "free prostochastic monoid". The reason for giving two statements is that the first avoids a number of technical points that will not play any further role, so the reader interested in the applications to the Markov Monoid algorithm may skip this second statement.

▶ **Theorem 11** (Existence of the free prostochastic monoid – weaker statement)**.** *For every finite alphabet A, there exists a compact monoid $\mathcal{P}A^*$ and a continuous injective morphism $\iota : A^* \to \mathcal{P}A^*$ such that every morphism $\phi : A^* \to \mathcal{S}_{Q\times Q}(\mathbb{R})$ extends uniquely to a continuous morphism $\widehat{\phi} : \mathcal{P}A^* \to \mathcal{S}_{Q\times Q}(\mathbb{R})$.*

We construct $\mathcal{P}A^*$ and ι. Consider $X = \prod_{\phi:A^*\to\mathcal{S}_{Q\times Q}(\mathbb{R})} \mathcal{S}_{Q\times Q}(\mathbb{R})$, the product of several copies of $\mathcal{S}_{Q\times Q}(\mathbb{R})$, one for each morphism $\phi : A^* \to \mathcal{S}_{Q\times Q}(\mathbb{R})$. An element m of X is denoted $(m(\phi))_{\phi:A^*\to\mathcal{S}_{Q\times Q}(\mathbb{R})}$: it is given by an element $m(\phi)$ of $\mathcal{S}_{Q\times Q}(\mathbb{R})$ for each morphism $\phi : A^* \to \mathcal{S}_{Q\times Q}(\mathbb{R})$. Thanks to Tychonoff's theorem, the monoid X equipped with the product topology is compact[2].

Consider the map $\iota : A \to X$ defined by $\iota(a) = (\phi(a))_{\phi:A\to\mathcal{P}}$, it induces a continuous injective morphism $\iota : A^* \to X$. To simplify notations, we sometimes assume that $A \subseteq X$ and denote a for $\iota(a)$.

Denote $\mathcal{P}A^* = \overline{A^*}$, the closure of $A^* \subseteq X$. Note that it is a compact monoid: the compactness follows from the fact that it is closed in X. By definition, an element $\overline{u}$ of $\mathcal{P}A^*$, called a *prostochastic word*, is obtained as the limit in $\mathcal{P}A^*$ of a sequence $\mathbf{u}$ of finite words. In this case we write $\lim \mathbf{u} = \overline{u}$ and say that $\mathbf{u}$ induces $\overline{u}$.

Note that by definition of the product topology on X, a sequence of finite words $\mathbf{u}$ converges in X if, and only if, for every morphism $\phi : A^* \to \mathcal{S}_{Q\times Q}(\mathbb{R})$, the sequence of stochastic matrices $\phi(\mathbf{u})$ converges.

We say that two converging sequences of finite words $\mathbf{u}$ and $\mathbf{v}$ are equivalent if they induce the same prostochastic word, *i.e.* if $\lim \mathbf{u} = \lim \mathbf{v}$. Equivalently, two converging sequences of finite words $\mathbf{u}$ and $\mathbf{v}$ are equivalent if, and only if, for every morphism $\phi : A^* \to \mathcal{S}_{Q\times Q}(\mathbb{R})$, we have $\lim \phi(\mathbf{u}) = \lim \phi(\mathbf{v})$.

We give a second, stronger statement about $\mathcal{P}A^*$, which in particular justifies the name "free prostochastic monoid".

From now on, by "monoid" we mean "compact topological monoids". The term topological means that the product function is continuous:

$$\begin{cases} \mathcal{P}\times\mathcal{P} & \to & \mathcal{P} \\ (s,t) & \mapsto & s\cdot t \end{cases}$$

A monoid is profinite if any two elements can be distinguished by a morphism into a finite monoid, *i.e.* by a finite automaton. (Formally speaking, this is the definition of residually finite monoids, which coincide with profinite monoids for compact monoids, see [1].)

To define prostochastic monoids, we use a stronger distinguishing feature, namely probabilistic automata. Probabilistic automata correspond to stochastic matrices over the rationals; here we use stochastic matrices over the reals, since $\mathcal{S}_{Q\times Q}(\mathbb{R})$ is compact, while $\mathcal{S}_{Q\times Q}(\mathbb{Q})$ is not.

[2] Note that here by compact we mean Hausdorff compact: distinct points have disjoint neighbourhoods.

▶ **Definition 12** (Prostochastic monoid)**.** A monoid $\mathcal{P}$ is prostochastic if for every elements $s \neq t$ in $\mathcal{P}$, there exists a continuous morphism $\psi : \mathcal{P} \to \mathcal{S}_{Q\times Q}(\mathbb{R})$ such that $\psi(s) \neq \psi(t)$.

There are many more prostochastic monoids than profinite monoids. Indeed, $\mathcal{S}_{Q\times Q}(\mathbb{R})$ is prostochastic, but not profinite in general.

The following theorem extends Theorem 11. The statement is the same as in the profinite theory, replacing "profinite monoid" by "prostochastic monoid".

▶ **Theorem 13** (Existence of the free prostochastic monoid – stronger statement)**.** *For every finite alphabet A,*

1. *There exists a prostochastic monoid $\mathcal{P}A^*$ and a continuous injective morphism $\iota : A^* \to \mathcal{P}A^*$ such that every morphism $\phi : A^* \to \mathcal{P}$, where $\mathcal{P}$ is a prostochastic monoid, extends uniquely to a continuous morphism $\widehat{\phi} : \mathcal{P}A^* \to \mathcal{P}$.*
2. *All prostochastic monoids satisfying this universal property are homeomorphic.*

The unique prostochastic monoid satisfying the universal property stated in item 1. is called the free prostochastic monoid, and denoted $\mathcal{P}A^*$.

▶ Remark. The free prostochastic monoid $\mathcal{P}A^*$ contains the free profinite monoid $\widehat{A^*}$. To see this, we start by recalling some properties of $\widehat{A^*}$, which is the set of *converging* sequences up to *equivalence*, where:

- a sequence of finite words $\mathbf{u}$ is converging if, and only if, for every deterministic automaton $\mathcal{A}$, the sequence is either ultimately accepted by $\mathcal{A}$ or ultimately rejected by $\mathcal{A}$, *i.e.* there exists $N \in \mathbb{N}$ such that either for all $n \geq N$, the word u_n is accepted by $\mathcal{A}$, or for all $n \geq N$, the word u_n is rejected by $\mathcal{A}$,
- two sequences of finite words $\mathbf{u}$ and $\mathbf{v}$ are equivalent if for every deterministic automaton $\mathcal{A}$, either both sequences are ultimately accepted by $\mathcal{A}$, or both sequences are ultimately rejected by $\mathcal{A}$.

Clearly:

- if a sequence of finite words is converging with respect to $\mathcal{P}A^*$, then it is converging with respect to $\widehat{A^*}$, as deterministic automata form a subclass of probabilistic automata,
- if two sequences of finite words are equivalent with respect to $\mathcal{P}A^*$, then they are equivalent with respect to $\widehat{A^*}$.

Every profinite word induces at least one prostochastic word: by compactness of $\mathcal{P}A^*$, every sequence of finite words $\mathbf{u}$ contains a converging subsequence with respect to $\mathcal{P}A^*$. This defines an injection from $\widehat{A^*}$ into $\mathcal{P}A^*$. In particular, this implies that $\mathcal{P}A^*$ is uncountable.

4.2 Reformulation of the Value 1 Problem

The aim of this subsection is to show that the value 1 problem, which talks about sequences of finite words, can be reformualted as an emptiness problem over prostochastic words.

▶ **Definition 14** (Prostochastic language of a probabilistic automaton)**.** Let $\mathcal{A}$ be a probabilistic automaton. The prostochastic language of $\mathcal{A}$ is:

$$L(\mathcal{A}) = \{\overline{u} \mid \widehat{\phi}(\overline{u})(q_0, F) = 1\}.$$

We say that $\mathcal{A}$ accepts a prostochastic word $\overline{u}$ if $\overline{u} \in L(\mathcal{A})$.

▶ **Theorem 15** (Reformulation of the value 1 problem)**.** *Let $\mathcal{A}$ be a probabilistic automaton. The following are equivalent:*

- $\mathrm{val}(\mathcal{A}) = 1$,
- *$L(\mathcal{A})$ is non-empty.*

Proof. Assume $\mathrm{val}(\mathcal{A}) = 1$, then there exists a sequence of words $\mathbf{u}$ such that $\lim P_{\mathcal{A}}(\mathbf{u}) = 1$. We see $\mathbf{u}$ as a sequence of prostochastic words. By compactness of $\mathcal{P}A^*$ it contains a converging subsequence. The prostochastic word induced by this subsequence belongs to $L(\mathcal{A})$.

Conversely, let $\overline{u}$ in $L(\mathcal{A})$, *i.e.* such that $\widehat{\phi}(\overline{u})(q_0, F) = 1$. Consider a sequence of finite words $\mathbf{u}$ inducing $\overline{u}$. By definition, we have $\lim \phi(\mathbf{u})(q_0, F) = 1$, *i.e.* $\lim P_{\mathcal{A}}(\mathbf{u}) = 1$, implying that $\mathrm{val}(\mathcal{A}) = 1$. ◀

4.3 The Limit Operator, Fast and Polynomial Prostochastic Words

We show in this subsection how to construct non-trivial prostochastic words, and in particular the polynomial prostochastic words. To this end, we need to better understand *convergence speeds phenomena*: different limit behaviours can occur, depending on how fast the underlying Markov chains converge.

We define a limit operator ω. Consider the function $f : \mathbb{N} \to \mathbb{N}$ defined by $f(n) = k!$, where k is maximal such that $k! \leq n$. The function f grows linearly: roughly, $f(n) \sim n$. The choice of n is arbitrary; one could replace n by any polynomial, or even by any subexponential function, see Remark 4.3.

The operator ω takes as input a sequence of finite words, and outputs a sequence of finite words. Formally, let $\mathbf{u}$ be a sequence of finite words, define:

$$\mathbf{u}^\omega = (u_n^{f(n)})_{n \in \mathbb{N}}.$$

It is not true in general that if $\mathbf{u}$ converges, then $\mathbf{u}^\omega$ converges. We will show that a sufficient condition is that $\mathbf{u}$ is fast.

We say that a sequence $(M_n)_{n \in \mathbb{N}}$ converges exponentially fast to M if there exists a constant $C > 1$ such that for all n large enough, $||M_n - M|| \leq C^{-n}$.

▶ **Definition 16** (Fast sequence)**.** A sequence of finite words $\mathbf{u}$ is fast if it converges (we denote $\overline{u}$ the prostochastic word it induces), and for every morphism $\phi : A^* \to \mathcal{S}_{Q \times Q}(\mathbb{R})$, the sequence $(\phi(u_n))_{n \in \mathbb{N}}$ converges exponentially fast.

A prostochastic word is *fast* if it is induced by *some* fast sequence. We denote by $\mathcal{P}A^*_f$ the set of fast prostochastic words. Note that a priori, not all prostochastic words are induced by some fast sequence.

We first prove that $\mathcal{P}A^*_f$ is a submonoid of $\mathcal{P}A^*$.

▶ **Lemma 17** (The concatenation of two fast sequences is fast)**.** *Let* $\mathbf{u}, \mathbf{v}$ *be two fast sequences. The sequence* $\mathbf{u} \cdot \mathbf{v} = (u_n \cdot v_n)_{n \in \mathbb{N}}$ *is fast.*

Let $\overline{u}$ and $\overline{v}$ be two fast prostochastic words, thanks to Lemma 17, the prostochastic word $\overline{u} \cdot \overline{v}$ is fast.

The remainder of this subsection is devoted to proving that ω is an operator $\mathcal{P}A^*_f \to \mathcal{P}A^*_f$. This is the key technical point of our characterisation. Indeed, we will define polynomial prostochastic words using concatenation and the operator ω, mimicking the definition of polynomial sequences of finite words. The fact that ω preserves the fast property of prostochastic words allows to obtain a perfect correspondence between polynomial sequences of finite words and polynomial prostochastic words.

Recall that M denotes a matrix in $\mathcal{S}_{Q \times Q}(\mathbb{R})$, and m a Boolean matrix. Note that when considering stochastic matrices we compute in the real semiring, and when considering

Boolean matrices, we compute products in the Boolean semiring, leading to two distinct notions of idempotent matrices.

The main technical tool is the following theorem, stating the exponentially fast convergence of the powers of a stochastic matrix.

▶ **Theorem 18** (Powers of a stochastic matrix). *Let $M \in \mathcal{S}_{Q\times Q}(\mathbb{R})$. Denote $P = M^{|Q|!}$. Then the sequence $(P^n)_{n\in\mathbb{N}}$ converges exponentially fast to a matrix M^ω, satisfying:*

$$\pi(M^\omega)(s,t) = \begin{cases} 1 & \text{if } \pi(P)(s,t) = 1 \text{ and } t \text{ is } \pi(P)\text{-recurrent,} \\ 0 & \text{otherwise.} \end{cases}$$

The following lemma shows that the ω operator is well defined by fast sequences. The second item shows that ω commutes with morphisms.

▶ **Lemma 19** (Limit operator for fast sequences). *Let $\mathbf{u}, \mathbf{v}$ be two equivalent fast sequences, inducing the fast prostochastic word $\overline{u}$. Then the sequences $\mathbf{u}^\omega$ and $\mathbf{v}^\omega$ are fast and equivalent, inducing the fast prostochastic word denoted $\overline{u}^\omega$.*

Furthermore, for every morphism $\phi : A^* \to \mathcal{S}_{Q\times Q}(\mathbb{R})$, we have $\widehat{\phi}(\overline{u}^\omega) = \widehat{\phi}(\overline{u})^\omega$.

We can now define polynomial prostochastic words.

First, ω-expressions are described by the following grammar:

$$E \quad \longrightarrow \quad a \quad | \quad E \cdot E \quad | \quad E^\omega.$$

We define an interpretation $\overline{\cdot}$ of ω-expressions into fast prostochastic words:

- $\overline{a}$ is prostochastic word induced by the constant sequence of the one letter word a,
- $\overline{E_1 \cdot E_2} = \overline{E_1} \cdot \overline{E_2}$,
- $\overline{E^\omega} = \overline{E}^\omega$.

The following definition of polynomial prostochastic words is in one-to-one correspondence with the definition of polynomial sequences of finite words.

▶ **Definition 20** (Polynomial prostochastic word). The set of *polynomial prostochastic words* is $\{\overline{E} \mid E \text{ is an } \omega\text{-expression}\}$.

▶ Remark. Why the term polynomial? Consider an ω-expression E, say $(a^\omega b)^\omega$, and the prostochastic word $\overline{(a^\omega b)^\omega}$, which is induced by the sequence of finite words $((a^{f(n)}b)^{f(n)})_{n\in\mathbb{N}}$. Roughly speaking $f(n) \sim n$, so this sequence represents a polynomial behaviour. Furthermore, the proofs above yield the following robustness property: all converging sequences of finite words $((a^{g(n)}b)^{h(n)})_{n\in\mathbb{N}}$, where $g, h : \mathbb{N} \to \mathbb{N}$ are subexponential functions, are equivalent, so they induce the same polynomial prostochastic word $\overline{(a^\omega b)^\omega}$. We say that a function $g : \mathbb{N} \to \mathbb{N}$ is subexponential if for all constants $C > 1$ we have $\lim_n g(n) \cdot C^{-n} = 0$; all polynomial functions are subexponential.

This justifies the terminology; we say that the polynomial prostochastic words represent all polynomial behaviours.

4.4 Reformulating the Characterisation

For proof purposes, we give an equivalent presentation of the Markov Monoid through ω-expressions. Given a probabilistic automaton $\mathcal{A}$, we define an interpretation $\langle\cdot\rangle$ of ω-expressions into Boolean matrices:

- $\langle a \rangle$ is $\pi(\phi(a))$,
- $\langle E_1 \cdot E_2 \rangle$ is $\langle E_1 \rangle \cdot \langle E_2 \rangle$,
- $\langle E^\omega \rangle$ is $\langle E \rangle^\sharp$, only defined if $\langle E \rangle$ is idempotent.

Then the Markov Monoid of $\mathcal{A}$ is $\{\langle E \rangle \mid E \text{ an } \omega\text{-expression}\}$.

The following theorem is a reformulation of Theorem 9, using the prostochastic theory. It clearly implies Theorem 9: indeed, a polynomial prostochastic word induces a polynomial sequence, and vice-versa.

▶ **Theorem 21** (Characterisation of the Markov Monoid algorithm). *The Markov Monoid algorithm answers "YES" on input $\mathcal{A}$ if, and only if, there exists a polynomial prostochastic word accepted by $\mathcal{A}$.*

The proof relies on the notion of reification, used in the following proposition, from which follows Theorem 21.

▶ **Definition 22** ((Reification)). Let $\mathcal{A}$ be a probabilistic automaton.

A sequence $(u_n)_{n\in\mathbb{N}}$ of words reifies a Boolean matrix m if for all states $s, t \in Q$, the sequence $\left(P_{\mathcal{A}}(s \xrightarrow{u_n} t)\right)_{n\in\mathbb{N}}$ converges and:

$$m(s,t) = 1 \iff \lim_n P_{\mathcal{A}}(s \xrightarrow{u_n} t) > 0.$$

▶ **Proposition 23** (Characterisation of the Markov Monoid algorithm). *For every ω-expression E, for every $\phi : A \to \mathcal{S}_{Q\times Q}(\mathbb{R})$, we have*

$$\pi(\widehat{\phi}(\overline{E})) = \langle E \rangle.$$

Consequently, for every probabilistic automaton $\mathcal{A}$:

- *any sequence inducing the polynomial prostochastic word $\overline{E}$ reifies $\langle E \rangle$,*
- *the element $\langle E \rangle$ of the Markov Monoid is a value 1 witness if, and only if, the polynomial prostochastic word $\overline{E}$ is accepted by $\mathcal{A}$.*

5 Towards an Optimality Argument

It was shown in [3] that the Markov Monoid algorithm subsumes all previous known algorithms to solve the value 1 problem. Indeed, it was proved that it is correct for the subclass of leaktight automata, and that the class of leaktight automata strictly contains all subclasses for which the value 1 problem has been shown to be decidable.

At this point, the Markov Monoid algorithm is the best algorithm *so far*. But can we go further? If we cannot, then what is an optimality argument? It consists in constructing a maximal subclass of probabilistic automata for which the problem is decidable. We can reverse the point of view, and equivalently construct an optimal algorithm, *i.e.* an algorithm that correctly solves a subset of the instances, such that no algorithm correctly solves a superset of these instances. However, it is clear that no such strong statement holds, as one can always from any algorithm obtain a better algorithm by precomputing finitely many instances.

Since there is no strong optimality argument, we can only give a subjective argument. We argue that the combination of our characterisation from Section 3 and the undecidability of the following problem, called the two-tier value 1 problem, supports the claim that the Markov Monoid algorithm is *in some sense* optimal.

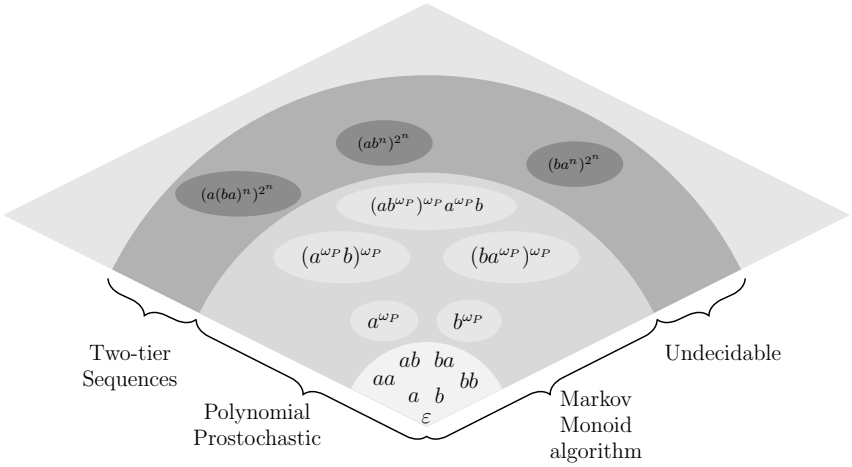

Figure 2 Optimality of the Markov Monoid algorithm.

▶ **Theorem 24** (Undecidability of the two-tier value 1 problem). *The following problem is undecidable: given a probabilistic automaton $\mathcal{A}$, determine whether there exist two finite words u, v such that $\lim_n P_{\mathcal{A}}((u \cdot v^n)^{2^n}) = 1$.*

Note that the two-tier value 1 problem is a priori much easier than the value 1 problem, as it restricts the set of sequences of finite words to very simple sequences. We call such sequences two-tier, because they exhibit two different behaviours: the word v is repeated a linear number of times, namely n, while the word $u \cdot v^n$ is repeated an exponential number of times, namely 2^n. The proof is obtained using the same reduction as for the undecidability of the value 1 problem, from [7], with a refined analysis.

To conclude:

- The characterisation says that the Markov Monoid algorithm captures exactly all polynomial behaviours.
- The undecidability result says that the undecidability of the value 1 problem arises when polynomial and exponential behaviours are combined.

So, the Markov Monoid algorithm is optimal in the sense that it captures a *large* set of behaviours, namely polynomial behaviours, and that no algorithm can capture both polynomial and exponential behaviours.

Acknowledgments. This paper and its author owe a lot to Szymon Toruńczyk's PhD thesis and its author, to Sam van Gool for his expertise on Profinite Theory, to Mikołaj Bojańczyk for his insightful remarks and to Jean-Éric Pin for his numerous questions and comments. The opportunity to present partial results on this topic in several scientific meetings has been a fruitful experience, and I thank everyone that took part in it.

References

1 Jorge Almeida. Profinite semigroups and applications. *Structural Theory of Automata, Semigroups, and Universal Algebra*, 207:1–45, 2005.

2 Krishnendu Chatterjee and Mathieu Tracol. Decidable problems for probabilistic automata on infinite words. In *LICS*, pages 185–194, 2012. doi:10.1109/LICS.2012.29.

3 Nathanaël Fijalkow, Hugo Gimbert, Edon Kelmendi, and Youssouf Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. *Logical Methods in Computer Science*, 11(1), 2015.

4 Nathanaël Fijalkow, Hugo Gimbert, and Youssouf Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. In *LICS*, pages 295–304, 2012. doi:10.1109/LICS.2012.40.

5 Nathanaël Fijalkow and Denis Kuperberg. ACME: automata with counters, monoids and equivalence. In *ATVA*, pages 163–167, 2014. doi:10.1007/978-3-319-11936-6_12.

6 Mai Gehrke, Serge Grigorieff, and Jean-Éric Pin. A topological approach to recognition. In *ICALP (2)*, pages 151–162, 2010. doi:10.1007/978-3-642-14162-1_13.

7 Hugo Gimbert and Youssouf Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In *ICALP (2)*, pages 527–538, 2010. doi:10.1007/978-3-642-14162-1_44.

8 Jean-Éric Pin. Profinite methods in automata theory. In *STACS*, pages 31–50, 2009. doi:10.4230/LIPIcs.STACS.2009.1856.

9 Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963. doi:10.1016/S0019-9958(63)90290-0.

10 Szymon Toruńczyk. *Languages of Profinite Words and the Limitedness Problem*. PhD thesis, University of Warsaw, 2011.