# Counting and Randomising in Automata Theory

Nathanaël Fijalkow

This document is a non-technical reading guide for my PhD thesis, which is a contribution to the study of quantitative models of automata, and more specifically of automata with counters and probabilistic automata.
The thesis manuscript consists of two chapters of equal length, the first about *finite-memory determinacy for boundedness games*, and the second about *the value 1 problem for probabilistic automata*. Although the two chapters are technically independent, some of the results are obtained by transferring techniques and ideas from one model to the other, revealing some similarities between them.

The first chapter deals with boundedness games, which are games with counters equipped with conditions requiring that the values of the counters are bounded. The problem we tackle here is the finite-memory determinacy for such games:

"Do there exist finite-memory winning strategies for boundedness games?"

The second chapter deals with probabilistic automata over finite words, and focuses on a decision problem called the value $1$ problem, which asks whether for a given probabilistic automaton, there exists a sequence of words accepted with arbitrarily high probability. This problem has been shown undecidable in 2010 by Gimbert and Oualhadj [GO10]; the objective of this chapter is to address the following question:

"To what extent is the value $1$ problem decidable?"

# Finite-Memory Determinacy for Boundedness Games

The first chapter investigates finite-memory determinacy for boundedness games. This study is motivated by, and belongs to, a research program launched ten years ago by Bojańczyk and Colcombet, aiming at understanding boundedness logics: the so-called monadic second-order logic augmented with the bounding quantifier (MSO + U) and cost-monadic second-order logic (cost-MSO).

## Context and motivations

### Boundedness questions

Boundedness questions have been introduced in the 80s, when Hashiguchi, and then later Leung, Simon and Kirsten solved the notoriously difficult star-height problem by reducing it to a boundedness question [Has90; Leu91; Sim94; Kir05]. A typical example of a boundedness question is: given a regular language $L \subseteq \{a,b\}^*$, does there exist a bound $\mathbf{N} \in \mathbb{N}$ such that all words from $L$ contain at most $\mathbf{N}$ occurrences of $a$? Since then, several problems from logics and automata have been formulated as boundedness questions, which motivated the introduction of logical formalisms expressing such questions: the logic MSO + U by Bojańczyk [Boj04] and its restriction cost-MSO by Colcombet [Col09].

The study of these two logics over the past ten years led to the definitions of natural and robust quantitative notions extending the theory of regular languages. This toolbox has been used in several contexts, two of which we further discuss now, the Mostowski index problem and a quantitative extension of the synthesis problem.

### The Mostowski index problem

The most important problem that has been reduced to a boundedness question is to decide the Mostowski index. This hierarchy, named after Mostowski [Mos84], is naturally cast as a hierarchy inside modal $\mu$-calculus, which is a powerful and well-studied logic introduced by Scott and de Bakker, and further developed by Kozen into the version most used nowadays [Koz83]. It enriches modal logics by two dual operators which compute least (denoted $\mu$) and greatest (denoted $\nu$) fixpoints. The Mostowski hierarchy quantifies the number of alternations between these two operators; for instance a $\mu\nu$ formula uses an arbitrary number of operators $\mu$, and then an arbitrary number of operators $\nu$. This hierarchy has been shown to be infinite and to have important properties, in particular with respect to computational complexity: for instance the complexity of the model-checking problem is naturally parameterised by the index in this hierarchy.

The Mostowski index problem is the following decision problem: given a property expressible in modal $\mu$-calculus and a level of the Mostowski hierarchy, can this property be formulated using a formula in this level? This problem has been open for over 30 years; it reformulates as a similar problem for parity automata over infinite trees, where the hierarchy is the number of priorities used in the parity condition. In 2008, Colcombet and Löding cast this problem as a boundedness question, namely boundedness of cost automata over infinite trees [CL08]. The latter problem is not known to be decidable either; however, it can be expressed in cost-MSO over infinite trees, motivating the study of this logic. This approach led to the solution of a special case of the Mostowski index problem, which is the weak definability for Büchi automata [CKLV13].

The Mostowski index problem motivated the investigations reported in the first chapter of this PhD. As we shall see, our results imply the decidability of the Mostowski index problem in the special case of thin trees.

## A quantitative extension of the synthesis problem

The synthesis problem, also known as Church's problem [Chu57], asks whether given a specification one can automatically construct a system satisfying this specification. A key conceptual idea, due to McNaughton [McN65], is to cast the synthesis problem as a game, in which two players have antagonistic goals. The game is played over a graph, called an arena, which represents the system; the first player, we call her Eve, represents the controller of the system, and her opponent, we call him Adam, represents the environment. The specification gives rise to a winning condition that Eve tries to ensure. Thus, the Church problem is equivalent to determining whether Eve has a winning strategy in the game, and to construct such a winning strategy. We give an example of such a game in the next section.

The celebrated Büchi-Landweber theorem [BL69] solves the synthesis problem for specifications given in MSO. Several extensions have been investigated, aiming at solving the synthesis problem for richer specifications. The boundedness logics appear naturally in this context, as specifying bounds is a missing feature of regular properties. This observation was made in the verification and synthesis community, motivating the introduction of formalisms expressing quantitative specifications, and in particular to further specify the existence of a time bound between requests and their grants. Historically, the first to be introduced are the finitary conditions introduced by Alur and Henzinger [AH98], then parametric linear temporal logic, defined by Alur, Etessami, La Torre and Peled [AELP01], and its fragment prompt linear temporal logic, introduced by Kupferman, Piterman and Vardi [KPV09].

These three approaches are subsumed by the logics MSO + U and cost-MSO, motivating studying the synthesis problem for boundedness logics. As we shall see, our results imply general decidability results for these questions.

## The LoCo conjecture and the theory of regular cost functions

The aim of the theory of regular cost functions is to provide tools for solving the boundedness questions formulated in cost-MSO. It has been successfully developed over finite words [Col09; Col13] and finite trees [CL10], yielding notions of regular expressions, automata and monoids that all have the same expressive power, and that extend the standard notions. This led to decidability results for boundedness questions over finite words and trees.

The decidability of cost-MSO over infinite trees is the main open question; it would yield a non-trivial extension of the celebrated Rabin's theorem stating the decidability of MSO over infinite trees [Rab69], and would imply the decidability of the Mostowski index problem mentioned above.
A cost-MSO formula $\varphi(N)$ is an MSO formula using the predicate $\mathsf{Card}(X) \leq N$, where we assume that each occurrence of $\mathsf{Card}(X) \leq N$ appears positively, *i.e.* under the scope of an even number of negations. This ensures a monotonic behaviour: if the formula holds for some $\mathbf{N} \in \mathbb{N}$, then it holds for larger values as well. The decision problem we are looking at is the boundedness problem:

$$\exists \mathbf{N} \in \mathbb{N}, \quad \forall t \in \mathsf{Trees}, \quad t, \mathbf{N} \models \varphi(N).$$

Colcombet and Löding attacked this problem in 2008, and showed that its solution requires a finite-memory determinacy result for boundedness games. We sketch the underlying ideas of their approach relying on cost-automata. Such automata are automata with counters that come in two dual forms: B-automata and S-automata. The former aim at minimising the values of the counters, they are meant to represent cost-MSO formulae, while the latter aim at maximising them, to represent the negations of cost-MSO formulae. Indeed, the crucial difficulty in proving the decidability of cost-MSO is to handle the negation, giving rise to these dual models.
There are actually four variants of cost-automata: in one direction, B- and S-, and in the other direction,

non-deterministic and alternating. These four models appear naturally, as for each construction in cost-MSO, one of the four variants is easily shown to be closed under this construction. For instance, the existential quantifications are easily implemented for non-deterministic automata, while the negation is not, but it is a simple operation dualising B- and S-alternating automata.

Hence to solve cost-MSO, one needs to prove that these four models are equivalent. To this end, one needs to generalise the famous construction of Muller and Schupp [MS87; MS95], which from an alternating automaton constructs an equivalent non-deterministic one. The heart of this construction is the positionality result for parity games, which states the existence of positional winning strategies in such games. The LoCo conjecture, stated by Colcombet and Löding, extends the positionality result for parity games to boundedness games, which are games with B- and S-conditions. We show an example of a boundedness game in the next section.

In summary, Colcombet and Löding proved the following implications:

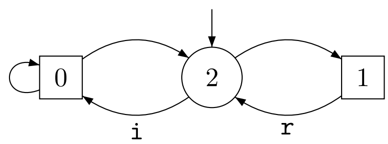
LoCo conjecture
$\Longrightarrow$ B- and S- alternating and non-deterministic automata are equivalent
$\Longrightarrow$ decidability of cost-MSO
$\Longrightarrow$ decidability of the Mostowski index problem

The LoCo conjecture resisted both proofs and refutations, and the only non-trivial positive case known was due to Vanden Boom [Van11], which implied the decidability of the weak variant of cost-MSO over infinite trees.

The LoCo conjecture is the main concern of the first chapter of this PhD. We show a number of examples witnessing interesting phenomena, and develop some original techniques to tackle it. As we shall see, our main results are to prove that it does not hold in general, but does hold in the special case of thin trees.

## A motivating example

We give an example of a boundedness game, which is the main object studied throughout this chapter.




A game is played over a finite or infinite graph. The vertices are represented either by circles and controlled by Eve, or by squares and controlled by Adam. A token is initially placed on the initial vertex, which is the central vertex in our example (it has an ingoing arrow). The player who controls the vertex the token is in chooses an edge and pushes the token along this edge. This interaction goes on forever, describing an infinite path in the graph. To declare the winner, we equip the game with a winning condition, which is a set of plays winning for Eve.

The winning condition of a boundedness game is the conjunction of a parity condition and a condition on the counters, requiring that their values remain bounded along the play. The parity condition looks at the sequence of priorities seen along the play; in our example the priorities label the vertices. The parity condition is satisfied if the smallest priority that appears infinitely often is even. In our example there is only one counter, and its value is updated by the actions labelling the edges. If an edge has no label, traversing it has no influence over the value of the counter. The action `i` increments the value by $1$, and the action `r` resets it to $0$.

Let us analyse the example, which witnesses the interaction between the parity condition and the counters. For Eve, going to the left is good for the parity condition, as she visits a vertex of color $0$, but bad for the counter, as it gets incremented. On the other hand, going to the right is good for the counter as it gets reset, but bad for the parity condition. It follows that in order to win, Eve has to alternate between going to the left and to the right.

The following three sections state the main results about boundedness games presented in the first chapter of this PhD.

## Collapse result for pushdown boundedness games

The sentence "the values of the counters remain bounded along the play" is ambiguous; it can be formalised in two different ways, which highlights the difference between MSO + U and cost-MSO. The non-uniform way, as in MSO+U, reads as follows:

Eve wins if she has a strategy such that for all plays,
there exists $\mathbf{N} \in \mathbb{N}$ such that all counters are bounded by $\mathbf{N}$ and the parity condition is satisfied.

The uniform way, as in cost-MSO, reads as follows:

Eve wins if there exist $\mathbf{N} \in \mathbb{N}$ and a strategy, such that for all plays,
all counters are bounded by $\mathbf{N}$ and the parity condition is satisfied.

The first question we address is: when are the two variants equivalent? We prove that this is (essentially) the case for pushdown games, but not for graphs of higher order.
This collapse result allows us to obtain the following theorem.

**Theorem.** *Solving pushdown $\omega$B-regular games is decidable.*

This theorem generalises a number of results about the synthesis problem for various logics subsumed by $\omega$B-regular languages.

## The frontier for finite-memory determinacy of boundedness games

In order to attack the LoCo conjecture, we develop a set of tools and techniques to prove finite-memory determinacy. We obtain two sets of results, the first about topologically closed conditions, and the second about boundedness conditions.

A condition is topologically closed if it can be described by a set of finite forbidden prefixes: a play is won if none of its prefixes are forbidden. Such conditions are very common, and the premium example is the boundedness condition itself: requiring that all counters remain bounded by a value $\mathbf{N} \in \mathbb{N}$ is equivalent to not having a prefix ending with $\mathbf{N}+1$ increments without a reset.

**Theorem.** *The amount of memory required to win a topologically closed condition is the size of the largest antichain in the lattice of left quotients of the condition.*

This resut does not make any assumption on the condition besides being topologically closed, which allows one to apply it to various settings, and in particular to boundedness conditions.

We obtain several positive and negative results about finite-memory determinacy for boundedness games. The most surprising result is negative, and shows that boundedness games are not in general finite-memory determined.

**Theorem.** *The LoCo conjecture does not hold over finite arenas.*

We provide a set of positive results showing that the subclass of boundedness conditions called finitary parity conditions is finite-memory determined. In light of the above counter-example, these positive results are maximal, drawing a frontier of finite-memory determinacy for boundedness conditions.

## The theory of regular cost functions over thin trees

In order to prove the decidability of cost-MSO over infinite trees, it is enough to prove that the LoCo conjecture holds for games played over tree arenas. Unfortunately, we did not manage to prove or disprove this restricted statement. We consider the special case of thin trees, which are the trees containing countably many branches. Our most technical achievement in this chapter is the following theorem.

**Theorem.** *The LoCo conjecture holds over thin trees arenas.*

We obtain as a corollary two decidability results, which are the most general results obtained so far for these two problems.

**Corollary.** *Cost-MSO is decidable over thin trees.*
*The Mostowski index problem for parity automata over thin trees is decidable.*

## In a nutshell

We study the finite-memory determinacy of boundedness games. We obtain positive and negative results, which allowed us to construct algorithms for the synthesis problem of pushdown boundedness games, as well as decidability results for cost-MSO and the Mostowski index problem over thin trees.

# The Value 1 Problem for Probabilistic Automata

The second chapter investigates the value $1$ problem for probabilistic automata. This study is part of a general program aiming at solving the synthesis problem for partially observable systems. These models are very general, implying that such problems become quickly undecidable. The objective of this chapter is an in-depth study of the decidable cases for the value $1$ problem.

## Context and motivations

### The synthesis problem for partially observable systems.................................................

A partially observable Markov decision process (POMDP) is a stochastic system whose evolution depends on the actions of a controller having only a partial observation on the evolution. This model appears in various fields such as operational research, artificial intelligence and motion planning in robotics. Developing formal methods to analyse such systems is a major challenge, which attracted a lot of attention in the past ten years. More specifically, the main objective is the synthesis problem, which asks for a given POMDP and given specification whether one can construct a controller for this POMDP ensuring the specification with high probability.

### Undecidability.................................................................................................

Unfortunately, the synthesis problem for POMDP is in general undecidable, motivating the quest for decidable cases. The starting point of this chapter is the result of Gimbert and Oualhadj from 2010 [GO10], which states that the synthesis problem remains undecidable under two restrictions:

- the controller is blind, *i.e.* does not observe anything about the evolution of the system,
- the objective is to satisfy the specification with probability arbitrarily close to $1$.

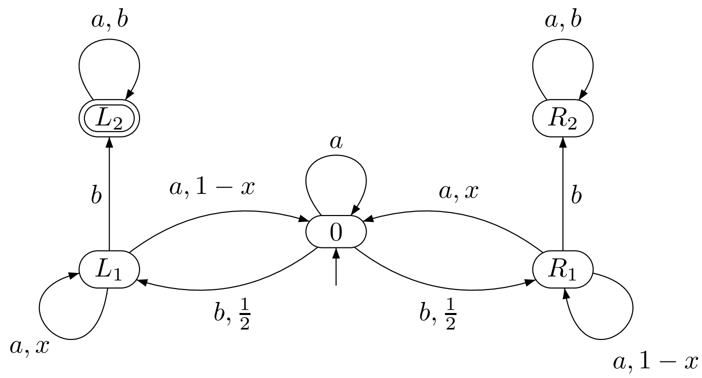
The setting of blind controllers corresponds to the model of probabilistic automata. We show an example of a probabilistic automaton in the next section.

The value of a POMDP is the supremum over all strategies of the probability to satisfy the specification, hence satisfying a specification with probability arbitrarily close to $1$ is equivalent to determining whether the value is $1$ or not. In other words, Gimbert and Oualhadj proved that the value $1$ problem is undecidable for probabilistic automata. This problem appears to be a natural first step towards understanding the synthesis problem for POMDP. Indeed, it is restricted yet already undecidable and witnesses complex phenomena.

The results presented in this chapter aim at analysing the value $1$ problem for probabilistic automata to draw a precise decidability frontier.

## A motivating example

We give an example of a probabilistic automaton, which is the main object studied throughout this chapter.




Probabilistic automata were introduced by Rabin [Rab63] as a generalisation of non-deterministic automata over finite words. Indeed, a probabilistic automaton is a non-deterministic automaton equipped with a probabilistic transition function assigning to non-deterministic transitions a probability. It follows that a probabilistic automaton $\mathcal{A}$ assigns to every finite word $u$ a value in $[0, 1]$, which is the probability that a run ends up in an accepting state, denoted $\mathcal{P}_A(u)$.

The value 1 problem takes as input a probabilistic automaton $\mathcal{A}$ and asks whether there exists a sequence of words $(u_n)_{n\in\mathbb{N}}$ such that $\lim_n \mathcal{P}_{\mathcal{A}}(u_n) = 1$. Equivalently, is it true that for all $\varepsilon > 0$, there exists a word $u$ such that $\mathcal{P}_{\mathcal{A}}(u) \geq 1 - \varepsilon$?

In the probabilistic automaton represented above, $x$ is a value in $(0, 1)$. The initial state is the central state $0$ and the unique final state is $L_2$. We claim that $\mathcal{A}$ has value 1 if, and only if, $x > \frac{1}{2}$.

After reading one $b$, the distribution is uniform over $L_1, R_1$. To reach $L_2$, one needs to read a $b$ from the state $L_1$, but on the right-hand side this leads to the non-accepting absorbing state $R_2$. In order to maximise the probability to reach $L_2$, one tries to "tip the scales" to the left.

If $x \leq \frac{1}{2}$, there is no hope to achieve this: reading a letter $a$ gives more chance to stay in $R_1$ than in $L_1$ thus all words are accepted with probability at most $\frac{1}{2}$.

However, if $x > \frac{1}{2}$ then one can show that $\mathcal{A}$ has value 1. More precisely, one can show that the probability of the sequence of words $((ba^n)^{2^n})_{n\in\mathbb{N}}$ converges to 1. Interestingly, repeating only $n$ times is not enough: the sequence $((ba^n)^n)_{n\in\mathbb{N}}$ does not witness that $\mathcal{A}$ has value 1.

The following three sections state the main results about the value 1 problem presented in the second chapter of this PhD.

## First step: constructing the Markov monoid algorithm

Our first contribution is to construct the Markov monoid algorithm, which is based on the algebraic notion of stabilisation monoids. We introduce the class of probabilistic leaktight automata and prove that the Markov monoid algorithm solves the value 1 problem for this subclass.

### Stabilisation monoids

The notion of stabilisation monoids has been introduced by Colcombet in the study of regular cost functions [Col09]. A stabilisation monoid is a set equipped with an associative product and a unary operation $\sharp$ called stabilisation. The intuitive meaning of $e^\sharp$ for an element $e$ is $\lim_n e^n$: the stabilisation monoids have a built-in notion of limits.

The Markov monoid of a probabilistic automaton generalises the transition monoid for non-deterministic automata by adding the stabilisation operation. Intuitively, each element of the Markov monoid represents the action of a sequence of words. The Markov monoid algorithm takes as input a probabilistic automaton, computes the Markov monoid of this automaton and looks for value $1$ witnesses.

Since the value $1$ problem is undecidable, the Markov monoid algorithm is incomplete, meaning that it does not abstract all behaviours of the probabilistic automaton. We prove that one direction holds: every behaviour predicted by the Markov monoid can be reflected on the probabilistic automaton, as formalised by the following theorem.

**Theorem.** *If the Markov monoid algorithm answers "YES", then the probabilistic automaton has value* $1$.

### The class of probabilistic leaktight automata

We introduce the class of probabilistic leaktight automata, and prove the following theorem, using the factorisation forest theorem of Simon [Sim94].

**Theorem.** *If the Markov monoid algorithm answers "NO" and the automaton is leaktight, then it does not have value* $1$. *Consequently, the value* $1$ *problem is decidable for probabilistic leaktight automata, and more precisely PSPACE-complete.*

Over the past years, different restrictions on the class of probabilistic automata have been introduced in order to obtain decidability results. The first subclass which was introduced specifically to decide the value 1 problem are the $\sharp$-acyclic automata by Gimbert and Oualhadj [GO10]. Later on, and at the same time as we introduced leaktight automata, Chatterjee and Tracol [CT12] introduced structurally simple automata, which are probabilistic automata satisfying a structural property (related to the decomposition-separation theorem from probability theory), and proved that the value 1 problem is decidable for structurally simple automata. Chadha, Sistla and Viswanathan introduced the subclass of hierarchical automata [CSV11], and showed that this restriction allows one to recover decidability results.
The next question we have to answer is to clarify the relations between all these subclasses. We prove the following theorem.

**Theorem.** *The classes of $\sharp$-acyclic, structurally simple and hierarchical probabilistic automata are strictly subsumed by the class of probabilistic leaktight automata.*

Consequently, the value 1 problem is decidable for hierarchical automata.

## Second step: understanding the Markov monoid algorithm

Our second contribution is to get a better understanding of what the Markov monoid algorithm computes, and what it does not. In particular, what behaviours make the value $1$ problem undecidable that are not captured by the Markov monoid algorithm? A first answer is given by looking at the numberless value $1$ problems, and a second by considering the notion of convergence speeds.

### The numberless value 1 problems

The Markov monoid does not depend on the numerical values of the probabilistic transitions, but only on the structure of the underlying non-deterministic automaton. Furthermore, the undecidability proof of the value $1$ problem given by Gimbert and Oualhadj [GO10] involves precise calculations, which depend on the numerical values.

Hence one may wonder whether abstracting away these values makes the value $1$ problem easier. We show that it does not. To this end, we provide a construction as stated in the following theorem.

**Theorem.** *There exists an effective construction which takes as input a probabilistic automaton $\mathcal{A}$ and constructs a non-deterministic automaton $\mathcal{M}$ such that the following are equivalent:*

1. *$\mathcal{A}$ has value $1$,*
2. *there exists a probabilistic transition function $\Delta$ such that the probabilistic automaton $\mathcal{M}[\Delta]$ has value $1$,*
3. *for all probabilistic transition functions $\Delta$, the probabilistic automaton $\mathcal{M}[\Delta]$ has value $1$.*

The properties 2. and 3. are called the existential (respectively universal) numberless value $1$ problem. It follows from this construction that both problems are undecidable.

**Theorem.** *The existential and universal value $1$ problems are recursively inseparable.*

The situation is even worse: the above construction shows that determining whether $\mathcal{M}[\Delta]$ has value $1$ does not depend at all on $\Delta$, which means that not specifying the numerical values of the probabilistic transitions does not make the value $1$ problem any easier.

### Convergence speeds and characterisation

Analysing the undecidability proof of the value $1$ problem reveals that the constructed probabilistic automata create two competing converging speeds making the combined behaviour hard to describe. To observe this behaviour, one needs to input sequences of words of the form $((u^n v)^{2^n})_{n \in \mathbb{N}}$, where $u$ and $v$ are two finite words. We show that this is actually enough to obtain the undecidability result, as stated in the following theorem.

**Theorem.** *The following problem is undecidable: given a probabilistic automaton $\mathcal{A}$, determine whether there exist two finite words $u, v$ such that $\lim_n \mathcal{P}_{\mathcal{A}}((u^n v)^{2^n}) = 1$.*

This refines the undecidability of the value $1$ problem, as it restricts to very simple sequences of words.

The behaviours described above are not taken into account by the Markov monoid algorithm. More precisely, these behaviours only exist when considering non-polynomial sequences. Intuitively, the sequence $((a^n b)^n)_{n \in \mathbb{N}}$ is polynomial, while the sequence $((a^n b)^{2^n})_{n \in \mathbb{N}}$ is not, since $n$ and $2^n$ are not polynomial related. We prove that the Markov monoid algorithm captures exactly all polynomial sequences, which is formalised in the following theorem.

**Theorem.** *Let $\mathcal{A}$ be a probabilistic automaton. The following are equivalent:*

- *the Markov monoid algorithm answers "YES",*
- *there exists a polynomial sequence $(u_n)_{n \in \mathbb{N}}$ such that $\lim_n \mathcal{P}_{\mathcal{A}}(u_n) = 1$.*

This theorem precisely characterises the computations of the Markov monoid algorithm. Combined with the two undecidability results mentioned above, this suggests that the Markov monoid algorithm is in some sense optimal.

## Third step: developing the prostochastic theory

Our third contribution is to introduce the prostochastic theory.

The profinite theory is a deep mathematical theory originating from topology. It has been developed in automata theory by Almeida, Pin, Weil and others, see for instance [Pin09]. In this context, it consists in constructing the topological completion of the set of finite words. In other words, it allows one to define the notion of converging sequences of finite words and their limits.

The prostochastic theory follows the same approach, generalising it to probabilistic automata. In particular, we construct the topological completion of the set of finite words, with respect to all probabilistic automata. We prove the following result, which reformulates the value $1$ problem over finite words as the emptiness problem over prostochastic words.

**Theorem.** *Let $\mathcal{A}$ be a probabilistic automaton. The following are equivalent:*
- *$\mathcal{A}$ has value $1$, i.e. there exists a sequence of words $(u_n)_{n \in \mathbb{N}}$ such that $\lim_n \mathcal{P}_{\mathcal{A}}(u_n) = 1$,*
- *there exists a prostochastic word accepted by $\mathcal{A}$.*

Our motivations for introducing the prostochastic theory is to formalise the proof of the characterisation of the Markov monoid algorithm given above. Indeed, it is rather technical as it requires to obtain precise bounds on convergence phenomena of sequences of Markov chains. The prostochastic theory is a language to formalise this proof, meaning that it provides a set of definitions and notions that are used to formulate the statements and the proofs of this theorem.

In particular, the main appeal of the prostochastic theory is that it provides a simple and natural definition for polynomial sequences of words. Indeed, the notion of convergence speeds is intrinsic to the prostochastic theory, which allows us to define polynomial prostochastic words from their convergence properties. Formally, we define an $\omega$ operator echoing the profinite theory for classical automata, and obtain the set of polynomial prostochastic words as the set of $\omega$-terms. We prove the following characterisation theorem, equivalent to the previous one.

**Theorem.** *Let $\mathcal{A}$ be a probabilistic automaton. The following are equivalent:*
- *the Markov monoid algorithm answers "YES",*
- *there exists a polynomial prostochastic word accepted by $\mathcal{A}$.*

## In a nutshell

We study the decidability frontier for the value $1$ problem for probabilistic automata. We introduce the Markov monoid algorithm and show that it subsumes all algorithms from the literature. We develop a profinite approach leading to a characterisation of this algorithm, which combined with refined undecidability results suggest that it is optimal.

## Bibliographic references


[AELP01] Rajeev Alur, Kousha Etessami, Salvatore La Torre, and Doron Peled. "Parametric Temporal Logic for Model Measuring". In: *ACM Transactions on Computational Logics* 2.3 (2001), pp. 388–407.

[AH98] Rajeev Alur and Thomas A. Henzinger. "Finitary Fairness". In: *ACM Transactions on Programming Languages and Systems* 20.6 (1998), pp. 1171–1194.

[Boj04] Mikołaj Bojańczyk. "A Bounding Quantifier". In: *CSL*. 2004, pp. 41–55.

[BL69] J. Richard Büchi and Lawrence H. Landweber. "Definability in the Monadic Second-Order Theory of Successor". In: *Journal of Symbolic Logic* 34.2 (1969), pp. 166–170.

[CSV11] Rohit Chadha, A. Prasad Sistla, and Mahesh Viswanathan. "Power of Randomization in Automata on Infinite Strings". In: *Logical Methods in Computer Science* 7.3 (2011).

[CT12] Krishnendu Chatterjee and Mathieu Tracol. "Decidable Problems for Probabilistic Automata on Infinite Words". In: *LICS*. 2012, pp. 185–194.

[Chu57] Alonzo Church. "Applications of recursive arithmetic to the problem of circuit synthesis". In: *Summaries of the Summer Institute of Symbolic Logic* 1 (1957), pp. 3–50.

[Col13] Thomas Colcombet. "Regular Cost Functions, Part I: Logic and Algebra over Words". In: *Logical Methods in Computer Science* 9.3 (2013).

[Col09] Thomas Colcombet. "The Theory of Stabilisation Monoids and Regular Cost Functions". In: *ICALP (2)*. 2009, pp. 139–150.

[CKLV13] Thomas Colcombet, Denis Kuperberg, Christof Löding, and Michael Vanden Boom. "Deciding the weak definability of Büchi definable tree languages". In: *CSL*. 2013, pp. 215–230.

[CL10] Thomas Colcombet and Christof Löding. "Regular Cost Functions over Finite Trees". In: *LICS*. 2010, pp. 70–79.

[CL08] Thomas Colcombet and Christof Löding. "The Non-deterministic Mostowski Hierarchy and Distance-Parity Automata". In: *ICALP (2)*. 2008, pp. 398–409.

[GO10] Hugo Gimbert and Youssouf Oualhadj. "Probabilistic Automata on Finite Words: Decidable and Undecidable Problems". In: *ICALP (2)*. 2010, pp. 527–538.

[Has90] Kosaburo Hashiguchi. "Improved Limitedness Theorems on Finite Automata with Distance Functions". In: *Theoretical Computer Science* 72.1 (1990), pp. 27–38.

[Kir05] Daniel Kirsten. "Distance Desert Automata and the Star-Height Problem". In: *Informatique Théorique et Applications* 39.3 (2005), pp. 455–509.

[Koz83] Dexter Kozen. "Results on the Propositional mu-Calculus". In: *Theoretical Computer Science* 27 (1983), pp. 333–354.

[KPV09] Orna Kupferman, Nir Piterman, and Moshe Y. Vardi. "From Liveness to Promptness". In: *Formal Methods in System Design* 34.2 (2009), pp. 83–103.

[Leu91] Hing Leung. "Limitedness Theorem on Finite Automata with Distance Functions: An Algebraic Proof". In: *Theoretical Computer Science* 81.1 (1991), pp. 137–145.

[McN65] Robert McNaughton. "Finite-state infinite games". In: *MAC Technical Report* (1965).

[Mos84] Andrzej W. Mostowski. "Regular expressions for infinite trees and a standard form of automata". In: *Computation Theory*. 1984, pp. 157–168.

[MS87] David E. Muller and Paul E. Schupp. "Alternating Automata on Infinite Trees". In: *Theoretical Computer Science* 54 (1987), pp. 267–276.

[MS95] David E. Muller and Paul E. Schupp. "Simulating Alternating Tree Automata by Nondeterministic Automata: New Results and New Proofs of the Theorems of Rabin, McNaughton and Safra". In: *Theoretical Computer Science* 141.1&2 (1995), pp. 69–107.

[Pin09] Jean-Éric Pin. "Profinite Methods in Automata Theory". In: *STACS*. 2009, pp. 31–50.

[Rab69] Michael O. Rabin. "Decidability of Second-Order Theories and Automata on Infinite Trees". In: *Transactions of the AMS* 141 (1969), pp. 1–23.

[Rab63] Michael O. Rabin. "Probabilistic Automata". In: *Information and Control* 6.3 (1963), pp. 230–245.

[Sim94] Imre Simon. "On Semigroups of Matrices over the Tropical Semiring". In: *Informatique Théorique et Applications* 28.3-4 (1994), pp. 277–294.

[Van11] Michael Vanden Boom. "Weak Cost Monadic Logic over Infinite Trees". In: *MFCS*. 2011, pp. 580–591.