Non-deterministic Weighted Automata on Random Words

Jakub Michaliszyn
University of Wrocław

Jan Otop
University of Wrocław

Abstract

We present the first study of non-deterministic weighted automata under probabilistic semantics. In this semantics words are random events, generated by a Markov chain, and functions computed by weighted automata are random variables. We consider the probabilistic questions of computing the expected value and the cumulative distribution for such random variables.

The exact answers to the probabilistic questions for non-deterministic automata can be irrational and are uncomputable in general. To overcome this limitation, we propose an approximation algorithm for the probabilistic questions, which works in exponential time in the automaton and polynomial time in the Markov chain. We apply this result to show that non-deterministic automata can be effectively determinised with respect to the standard deviation metric.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects, Theory of computation → Quantitative automata, Mathematics of computing → Markov networks

Keywords and phrases quantitative verification, weighted automata, expected value

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2018.10

Acknowledgements This work was supported by the National Science Centre (NCN), Poland under grant 2014/15/D/ST6/04543. We would like to thank anonymous reviewers for their valuable comments on this paper. Our special thanks go to Günter Rote who pointed out a blooper in an earlier version of our running example.

1 Introduction

Weighted automata are finite automata in which transitions carry weights [13]. We study here weighted automata (on finite and infinite words) whose semantics is given by *value functions* (such as sum or average) [8]. In such a weighted automaton transitions are labeled with rational numbers and hence every run yields a sequence of rationals, which the value function aggregates into a single (real) number. This number is the value of the run, and the value of a word is the infimum over values of all accepting runs on that word.

The value function approach has been introduced to express quantitative system properties (performance, energy consumption, etc.) and it serves as a foundation for *quantitative verification* [8, 18]. Basic decision questions for weighted automata are quantitative counterparts of the emptiness and universality questions obtained by imposing a threshold on the values of words.

Probabilistic semantics. The emptiness and the universality problems correspond to the best-case and the worst-case analysis. For the average-case analysis, weighted automata are considered under probabilistic semantics, in which words are random events generated by a

Markov chain [7, 9]. In this setting, functions from words to reals computed by deterministic weighted automata are measurable and hence can be considered as random variables. The fundamental probabilistic questions are to compute *the expected value* and *the cumulative distribution* for a given automaton and a Markov chain.

The deterministic case. Weighted automata under probabilistic semantics have been studied only in the deterministic case. In [7], a close relationship between weighted automata under probabilistic semantics and weighted Markov chains has been established. For a weighted automaton $\mathcal{A}$ and a Markov chain $\mathcal{M}$ representing the distribution over words, the probabilistic problems for $\mathcal{A}$ and $\mathcal{M}$ coincide with the probabilistic problem of the weighted Markov chain $\mathcal{A} \times \mathcal{M}$. Weighted Markov chains have been intensively studied with single and multiple quantitative objectives [3, 15, 24, 10]. The above reduction does not extend to non-deterministic weighted automata [9, Example 30].

Significance of nondeterminism. Non-deterministic weighted automata are provably more expressive than their deterministic counterpart [8]. Many important system properties can be expressed with weighted automata only in the nondeterministic setting. This includes minimal response time, minimal number of errors and the edit distance problem [18], which serves as the foundation for the *specification repair* framework from [5].

Non-determinism can also arise as a result of abstraction. The exact systems are often too large and complex to operate on and hence they are approximated with smaller non-deterministic models [11]. The abstraction is especially important for multi-threaded programs, where the explicit model grows exponentially with the number of threads [17].

Our contributions

We study non-deterministic weighted automata under probabilistic semantics. We work with weighted automata as defined in [8], where a value function f is used to aggregate weights along a run, and the value of the word is the infimum over values of all runs. (The infimum can be changed to supremum as both definitions are dual). We primarily focus on the two most interesting value functions: the sum of weights over finite runs, and the limit average over infinite runs. The main results presented in this paper are as follows.

- We show that the answers to the probabilistic questions for weighted automata with the sum and limit-average value functions can be irrational (Theorem 5) and cannot be computed by any effective representation (Theorem 6).
- We establish approximation algorithms for the probabilistic questions for weighted automata with the sum and limit-average value functions. The approximation is #P-complete for (total) weighted automata with the sum value function (Theorem 10), and it is PSpace-hard and solvable in exponential time for weighted automata with the limit-average value function (Theorem 16).
- We show that weighted automata with the limit-average value function can be approximately determinised (Theorem 18). Given an automaton $\mathcal{A}$ and $\epsilon > 0$, we show how to compute a deterministic automaton $\mathcal{A}_D$ such that the expected difference between the values returned by both automata is at most ϵ.

Applications

We briefly discuss applications of our contributions in quantitative verification.

- The expected-value question corresponds to the average-case analysis in quantitative verification [7, 9]. Using results from this paper, we can perform the average-case analysis with respect to quantitative specifications given by non-deterministic weighted automata.
- The universality problem for non-deterministic automata, which asks whether all words have the value below a given threshold, forms a basis for some quantitative-model-checking frameworks [8]. Unfortunately, the universality problem is undecidable for weighted automata with the sum or the limit average values functions. The distribution question can be considered as a computationally-attractive variant of universality, i.e., we ask whether almost-all words have value below some given threshold. We show that if the threshold can be approximated, the distribution question can be computed effectively.
- Weighted automata have been used to formally study online algorithms [2]. Online algorithms have been modeled by deterministic weighted automata, which make choices based solely on the past, while offline algorithms have been modeled by non-deterministic weighted automata. Relating deterministic and non-deterministic models allowed for formal verification of the worst-case competitiveness ratio of online algorithms. Using the result from our paper, we can extend the analysis from [2] to the average-case competitiveness.

Related work

Probabilistic verification of qualitative properties. Probabilistic verification asks for the probability of the sets of traces satisfying a given property. For non-weighted automata, it has been extensively studied [26, 12, 3] and implemented [22, 19]. The prevalent approach in this area is to work with deterministic automata, and apply determinisation as needed. To obtain better complexity bounds, the probabilistic verification problem has been directly studied for unambiguous Büchi automata in [4]; the authors explain there the potential pitfalls in the probabilistic analysis of non-deterministic automata.

Weighted automata under probabilistic semantics. Probabilistic verification of weighted automata and their extensions has been studied in [9]. All automata considered there are deterministic.

Markov Decision Processes (MDPs). MDPs are a classical extension of Markov chains, which allow to model control in a stochastic environment [3, 15]. In MDPs probabilistic and non-deterministic transitions are interleaved. Intuitively, the non-determinism in MDPs is resolved based on the past, i.e., already generated events. In our setting, non-deterministic weighted automata work over completely generated words and hence non-determinism may be resolved based on following letters, considered as future events.

Approximation determinisation. As weighted automata are not determinisable, Boker and Henzinger [6] studied *approximate* determinisation defined as follows. The distance $d_{\sup}$ between weighted automata $\mathcal{A}_1, \mathcal{A}_2$ is defined as $d_{\sup}(\mathcal{A}_1, \mathcal{A}_2) = \sup_w |\mathcal{A}_1(w) - \mathcal{A}_2(w)|$. A nondeterministic weighted automaton $\mathcal{A}$ can be *approximately* determinised if for every $\epsilon > 0$ there exists a deterministic automaton $\mathcal{A}_D$ such that $d_{\sup}(\mathcal{A}, \mathcal{A}_D) \leq \epsilon$. Unfortunately, weighted automata with the limit average value function cannot be approximately determinised [6]. In this work we show that the approximate determinisation is possible for the standard deviation metric d_{std} defined as $d_{\text{std}}(\mathcal{A}_1, \mathcal{A}_2) = \mathbb{E}(|\mathcal{A}_1(w) - \mathcal{A}_2(w)|)$.

2 Preliminaries

Given a finite alphabet Σ of letters, a *word* w is a finite or infinite sequence of letters. We denote the set of all finite words over Σ by Σ^*, and the set of all infinite words over Σ by Σ^ω. For a word w, we define $w[i]$ as the i-th letter of w, and we define $w[i,j]$ as the subword $w[i]w[i+1]\dots w[j]$ of w. We use the same notation for other sequences defined later on. By $|w|$ we denote the length of w.

A *(non-deterministic) finite automaton* (NFA) is a tuple $(\Sigma, Q, Q_0, F, \delta)$ consisting of an input alphabet Σ , a finite set of states Q, a set of initial states $Q_0 \subseteq Q$, a set of final states F, and a finite transition relation $\delta \subseteq Q \times \Sigma \times Q$.

We denote by $\delta(q,a)$ the set of states $\{q' \mid \delta(q,a,q')\}$ and by $\delta(S,a)$ the set of states $\bigcup_{q\in S} \delta(q,a)$. We extend this to $\widehat{\delta}\colon 2^Q \times \Sigma^* \to 2^Q$ in the following way: $\widehat{\delta}(S,\epsilon) = S$ (where ϵ is the empty word) and $\widehat{\delta}(S,aw) = \widehat{\delta}(\delta(S,a),w)$, i.e., $\widehat{\delta}(S,w)$ is the set of states reachable from S via δ over the word w.

Weighted automata. A *weighted automaton* is a finite automaton whose transitions are labeled by rational numbers called *weights*. Formally, a weighted automaton is a tuple $(\Sigma, Q, Q_0, F, \delta, C)$, where the first five elements are as in the finite automata, and $C\colon \delta \to \mathbb{Q}$ is a function that defines *weights* of transitions. An example of a weighted automaton is depicted in Figure 1.

The size of a weighted automaton $\mathcal{A}$, denoted by $|\mathcal{A}|$, is $|Q| + |\delta| + \sum_{q,q',a} \mathrm{len}(C(q,a,q'))$, where len is the sum of the lengths of the binary representations of the numerator and the denominator of a given rational number.

A *run* π of an automaton $\mathcal{A}$ on a word w is a sequence of states $\pi[0]\pi[1]\dots$ such that $\pi[0]$ is an initial state and for each i we have $(\pi[i-1], w[i], \pi[i]) \in \delta$. A finite run π of length k is *accepting* if and only if the last state $\pi[k]$ belongs to the set of accepting states F. As in [8], we do not consider ω-accepting conditions and assume that all infinite runs are accepting. Every run π of an automaton $\mathcal{A}$ on a (finite or infinite) word w defines a sequence of weights of successive transitions of $\mathcal{A}$ as follows. Let $(C(\pi))[i]$ be the weight of the i-th transition, i.e., $C(\pi[i-1], w[i], \pi[i])$. Then, $C(\pi) = (C(\pi)[i])_{1\le i\le |w|}$. A *value functions* f is a function that assigns real numbers to sequences of rational numbers. The value $f(\pi)$ of the run π is defined as $f(C(\pi))$.

The value of a (non-empty) word w assigned by the automaton $\mathcal{A}$, denoted by $\mathcal{L}_\mathcal{A}(w)$, is the infimum of the set of values of all accepting runs on w. The value of a word that has no (accepting) runs is infinite. To indicate a particular value function f that defines the semantics, we will call a weighted automaton $\mathcal{A}$ an f-automaton.

Value functions. We consider the following value functions. For finite runs, functions Min and Max are defined in the usual manner, and the function Sum is defined as

$$\textsc{Sum}(\pi) = \sum_{i=1}^{|C(\pi)|} (C(\pi))[i]$$

For infinite runs we consider the supremum Sup and infimum Inf functions (defined like Max and Min but on infinite runs) and the limit average function LimAvg defined as

$$\textsc{LimAvg}(\pi) = \limsup_{k\to\infty} \textsc{Avg}(\pi[0,k])$$

where for finite runs π we have $\textsc{Avg}(\pi) = \textsc{Sum}(\pi)/|C(\pi)|$.

2.1 Probabilistic semantics

A (finite-state discrete-time) *Markov chain* is a tuple $\langle \Sigma, S, s_0, E \rangle$, where Σ is the alphabet of letters, S is a finite set of states, s_0 is an initial state, $E \colon S \times \Sigma \times S \mapsto [0,1]$ is an edge probability function, which for every $s \in S$ satisfies that $\sum_{a \in \Sigma, s' \in S} E(s,a,s') = 1$. By $|\mathcal{M}| = |S| + |E| + \sum_{q,q',a} \mathrm{len}(E(q,a,q'))$ we denote the size of the Markov chain $\mathcal{M}$. An example of a single-state Markov chain is depicted in Figure 1.

The probability of a finite word u w.r.t. a Markov chain $\mathcal{M}$, denoted $\mathbb{P}_{\mathcal{M}}(u)$, is the sum of probabilities of paths from s_0 labeled by u, where the probability of a path is the product of probabilities of its edges. For basic open sets $u \cdot \Sigma^\omega = \{uw \mid w \in \Sigma^\omega\}$, we have $\mathbb{P}_{\mathcal{M}}(u \cdot \Sigma^\omega) = \mathbb{P}_{\mathcal{M}}(u)$, and then the probability measure over infinite words defined by $\mathcal{M}$ is the unique extension of the above measure (by Carathéodory's extension theorem [14]). We will denote the unique probability measure defined by $\mathcal{M}$ as $\mathbb{P}_{\mathcal{M}}$, and the associated expectation measure as $\mathbb{E}_{\mathcal{M}}$. For example, for the Markov chain $\mathcal{M}$ presented in Figure 1, we have that $\mathbb{P}_{\mathcal{M}}(ab) = \frac{1}{4}$, and so $\mathbb{P}_{\mathcal{M}}(\{w \in \{a,b\}^\omega \mid w[0,1] = ab\}) = \frac{1}{4}$, whereas $\mathbb{P}_{\mathcal{M}}(X) = 0$ for any finite set of infinite words X.

A *terminating* Markov chain $\mathcal{M}^T$ is a tuple $\langle \Sigma, S, s_0, E, T \rangle$, where Σ, S and s_0 are as usual, $E \colon S \times (\Sigma \cup \{\epsilon\}) \times S \mapsto [0,1]$ is the edge probability function, such that if $E(s,a,t)$, then $a = \epsilon$ if and only if $t \in T$, and for every $s \in S$ we have $\sum_{a \in \Sigma \cup \{\epsilon\}, s' \in S} E(s,a,s') = 1$, and T is a set of terminating states such that the probability of reaching a terminating state from any state s is positive. Notice that the only ϵ-transitions in a terminating Markov chain are those that lead to a terminating state.

The probability of a finite word u w.r.t. $\mathcal{M}$, denoted $\mathbb{P}_{\mathcal{M}^T}(u)$, is the sum of probabilities of paths from s_0 labeled by u such that the only terminating state on this path is the last one. Notice that $\mathbb{P}_{\mathcal{M}^T}$ is a probability distribution on finite words whereas $\mathbb{P}_{\mathcal{M}}$ is not (because the sum of probabilities may exceed 1).

Automata as random variables. A weighted automaton defines the function $\mathcal{L}_{\mathcal{A}}(w) \colon \Sigma^\omega \mapsto \mathbb{R}$ that assigns values to words. This function is measurable for all the automata types we consider in this paper (see Remark 2 below). Thus, this function can be interpreted as random variables w.r.t. the probabilistic space we consider. Hence, for a given automaton $\mathcal{A}$ and a Markov chain $\mathcal{M}$, we consider the following quantities:

$\mathbb{E}_{\mathcal{M}}(\mathcal{A})$ – the expected value of the random variable defined by $\mathcal{A}$ w.r.t. the probability measure defined by $\mathcal{M}$.
$\mathbb{D}_{\mathcal{M},\mathcal{A}}(\lambda) = \mathbb{P}_{\mathcal{M}}(\{w \mid \mathcal{L}_{\mathcal{A}}(w) \leq \lambda\})$ – the (cumulative) distribution function of the random variable defined by $\mathcal{A}$ w.r.t. the probability measure defined by $\mathcal{M}$.

In the finite words case, the expected value $\mathbb{E}_{\mathcal{M}^T}$ and the distribution $\mathbb{D}_{\mathcal{M}^T,\mathcal{A}}$ are defined in the same manner.

▶ **Remark 1** (Bounds on the expected value and the distribution). *Both quantities can be easily bounded: the value of the distribution function is always between* 0 *and* 1. *For a* LimAvg*-automaton* $\mathcal{A}$*, we have* $\mathbb{E}_{\mathcal{M}}(\mathcal{A}) \in [\min_{\mathcal{A}}, \max_{\mathcal{A}}] \cup \{\infty\}$*, where* $\min_{\mathcal{A}}$ *and* $\max_{\mathcal{A}}$ *denote the minimal and the maximal weight of* $\mathcal{A}$*. For a* Sum*-automaton* $\mathcal{A}$*, we have* $\mathbb{E}_{\mathcal{M}}(\mathcal{A}) \in [L_{\mathcal{M}} \cdot \min_{\mathcal{A}}, L_{\mathcal{M}} \cdot \max_{\mathcal{A}}] \cup \{\infty\}$*, where* $L_{\mathcal{M}}$ *is the expected length of a word generated by* $\mathcal{M}$ *(it can be computed in a standard way [16, Section 11.2]). In both cases,* $\mathbb{E}_{\mathcal{M}}(\mathcal{A}) = \infty$ *if and only if the probability of the set of words with no accepting runs in* $\mathcal{A}$ *is positive. Note that we consider no* ω*-accepting conditions, and hence all infinite runs of* Sum*-automata are accepting. Still there can be infinite words, on which a given* Sum*-automaton has no infinite runs. We show in Section 3.2 that the distribution and expected value may be irrational, even for integer weights and uniform distributions.*

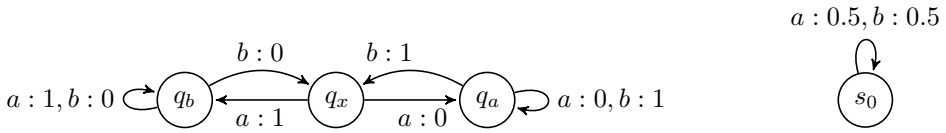

Figure 1 The automaton $\mathcal{A} = \{\{a,b\}, \{q_x, q_a, q_b\}, \{q_a, q_b\}, \emptyset, \delta, C\}$, where $\delta = \{(q_a, a, q_a), (q_a, b, q_a), (q_a, b, q_x), (q_x, a, q_a), (q_x, a, q_b), (q_b, a, q_x), (q_b, a, q_b), (q_b, b, q_b)\}$ and C such that $C(q_a, b, q_a) = C(q_b, a, q_b) = C(q_a, b, q_x) = C(q_x, a, q_b) = 1$ and for all other inputs the value of C is 0 (left) and the Markov chain $\mathcal{M} = \{\{a,b\}, \{s_0\}, \{s_0\}, E\}$ where E always returns 0.5 (right).

▶ **Remark 2** (Measurability of functions represented by automata). *For automata on finite words,* Inf*-automata and* Sup*-automata, measurability of $\mathcal{L}_\mathcal{A}$ is straightforward. To show that $\mathcal{L}_\mathcal{A}(w)\colon \Sigma^\omega \mapsto \mathbb{R}$ is measurable for any non-deterministic* LimAvg*-automaton $\mathcal{A}$, it suffices to show that for every $x \in \mathbb{R}$, the preimage $\mathcal{L}_\mathcal{A}^{-1}(-\infty, x]$ is measurable. Let Q be the set of states of $\mathcal{A}$. Consider the set $\Sigma^\omega \times \mathbb{Q}^\omega$. We can define a subset $A_x \subseteq \Sigma^\omega \times \mathbb{Q}^\omega$ of the pairs, the word and the run on it, where the value of the run is less than or equal to x. The set A_x can be presented as a countable intersection of open sets, and hence it is Borel. Observe that $\mathcal{L}_\mathcal{A}^{-1}(-\infty, x]$ is the projection of A_x on the first component Σ^ω. The projection of a Borel set is* analytic*, which is measurable [20]. Thus, $\mathcal{L}_\mathcal{A}$ defined by a non-deterministic* LimAvg*-automaton is measurable.*

The above proof of measurability requires some knowledge of descriptive set theory. We will give a direct proof of measurability of $\mathcal{L}_\mathcal{A}$ in the paper (Theorem 16).

2.2 Computational questions

We consider the following basic computational questions:

The expected value question: Given an f-automaton $\mathcal{A}$ and a (terminating) Markov chain $\mathcal{M}$, compute $\mathbb{E}_\mathcal{M}(\mathcal{A})$.
The distribution question: Given an f-automaton $\mathcal{A}$, a (terminating) Markov chain $\mathcal{M}$ and a threshold λ, compute $\mathbb{D}_{\mathcal{M},\mathcal{A}}(\lambda)$.

Each of the above questions have its decision variant (useful for lower bounds), where instead of computing the value we ask whether the value is less than a given threshold t.

The above questions have their approximate variants:

The approximate expected value question: Given an f-automaton $\mathcal{A}$, a (terminating) Markov chain $\mathcal{M}$, $\epsilon > 0$, compute a number η such that $|\eta - \mathbb{E}_\mathcal{M}(\mathcal{A})| \leq \epsilon$.
The approximate distribution question: Given an f-automaton $\mathcal{A}$, a (terminating) Markov chain $\mathcal{M}$, a threshold λ and $\epsilon > 0$ compute a number $\eta \in [\mathbb{D}_{\mathcal{M},\mathcal{A}}(\lambda - \epsilon), \mathbb{D}_{\mathcal{M},\mathcal{A}}(\lambda + \epsilon)]$.

In the later case, we use the Skorokhod's notion. One could expect there "$\eta \in [\mathbb{D}_{\mathcal{M},\mathcal{A}}(\lambda) - \epsilon, \mathbb{D}_{\mathcal{M},\mathcal{A}}(\lambda) + \epsilon]$" instead. However, this would lead to undecidable approximation in the LimAvg case (cf. Theorem 6).

3 Basic properties

Consider an f-automaton $\mathcal{A}$, a Markov chain $\mathcal{M}$ and a set of words X. We denote by $\mathbb{E}_\mathcal{M}(\mathcal{A} \mid X)$ the expected value of $\mathcal{A}$ w.r.t. $\mathcal{M}$ restricted only to words in the set X (see [14]).

The following says that we can disregard a set of words with probability 0 (e.g. containing only some of the letters under uniform distribution) while computing the expected value.

▶ **Fact 3.** *If* $\mathbb{P}(X) = 1$ *then* $\mathbb{E}_{\mathcal{M}}(\mathcal{A}) = \mathbb{E}_{\mathcal{M}}(\mathcal{A} \mid X)$.

The proof is rather straightforward; the only interesting case is when there are some words not in X with infinite values. But for all the functions we consider, one can show that in this case there is a set of words with infinite value that has a non-zero probability, and therefore $\mathbb{E}_{\mathcal{M}}(\mathcal{A}) = \mathbb{E}_{\mathcal{M}}(\mathcal{A} \mid X) = \infty$.

One important corollary of Fact 3 is that if $\mathcal{M}$ is, for example, uniform, then because the set Y of ultimately-periodic words (i.e., words of the form vw^ω) has probability 0, we have $\mathbb{E}_{\mathcal{M}}(\mathcal{A}) = \mathbb{E}_{\mathcal{M}}(\mathcal{A} \mid \Sigma^\omega \setminus Y)$. This suggests a possibility that the expected value may not be realised by any ultimately periodic word. We exemplify this in Remark 13.

3.1 Example of computing expected value by hand

Consider a LimAvg-automaton $\mathcal{A}$ and a Markov chain $\mathcal{M}$ depicted in Figure 1. We encourage the reader to take a moment to study this automaton and try to figure out its expected value.

The idea behind $\mathcal{A}$ is as follows. Assume that $\mathcal{A}$ is in a state q_l for some $l \in \{a, b\}$. Then, it reads a word up to the first occurrence a subword ba, where it has a possibility to go to q_x and then to non-deterministically choose q_a or q_b as the next state. Since going to q_x and back to q_l costs the same as staying in q_l, we will assume that the automaton always goes to q_x in such a case. When an automaton is in the state q_x and has to read a word $w = a^j b^k$, then average cost of a run on w is $\frac{j}{j+k}$ if the run goes to q_b and $\frac{k}{j+k}$ otherwise. So the run with the lowest value is the one that goes to q_a if $j > k$ and q_b otherwise.

To compute the expected value of the automaton, we focus on the set X of words w such that for each positive $n \in \mathbb{N}$ there are only finitely many prefixes of w of the form $w'a^j b^k$ such that $\frac{j+k}{|w'|+j+k} \geq \frac{1}{n}$. Notice that this means that w contains infinitely many a and infinitely many b. It can be proved in a standard manner that $\mathbb{P}_{\mathcal{M}}(X) = 1$.

Let $w \in X$ be a random event, which is a word generated by $\mathcal{M}$. Since w contains infinitely many letters a and b, it can be partitioned in the following way. Let $w = w_1 w_2 w_3 \dots$ be a partition of w such that each w_i for $i > 0$ is of the form $a^j b^k$ for $j \geq 0, k > 0$, and for $i > 1$ we also have $j > 0$. For example, the partition of $w = baaabbbaabbbaba\dots$ is such that $w_1 = b$, $w_2 = aaabbb$, $w_3 = aabbb$, $w_4 = ab$, $\dots$. Let $s_i = |w_1 w_2 \dots w_i|$.

We now define a run π_w on w as follows:

$$q_1^w \dots q_1^w q_x q_2^w \dots q_2^w q_x q_3^w \dots q_3^w q_x q_4^w \dots$$

where the length of each block of q_i is $|w_i| - 1$, $q_0^w = q_a$ and $q_i^w = q_a$ if $w_i = a^j b^k$ for some $j > k$ and $q_i^w = q_b$ otherwise. It can be shown by a careful consideration of all possible runs that this run's value is the infimum of values of all the runs on this word.

▶ **Lemma 4.** $\mathcal{L}_{\mathcal{A}}(w) = \text{LimAvg}(\pi_w)$.

By Fact 3 and Lemma 4, it remains to compute the expected value of $\text{LimAvg}(\{\pi_w \mid w \in X\})$. As the expected value of the sum is the sum of expected values, we can state that

$$\mathbb{E}_{\mathcal{M}}(\text{LimAvg}(\{\pi_w \mid w \in X\})) = \limsup_{s\to\infty} \frac{1}{s} \cdot \sum_{i=1}^{s} \mathbb{E}_{\mathcal{M}}\left(\{(C(\pi_w))[i] \mid w \in X\}\right)$$

It remains to compute $\mathbb{E}_{\mathcal{M}}((C(\pi_w)[i])$. If i is large enough (and since the expected value does not depend on a finite number of values, we assume that it is), the letter $\pi_w[i]$ is in some block $w_s = a^j b^k$. There are $j + k$ possible letters in this block, and the probability that

the letter $\pi_w[i]$ is an ith letter in such a block is $2^{-(j+k+2)}$ ("+2", because the block has to be maximal, so we need to include the letters before the block and after the block). So the probability that a letter is in a block $a^j b^k$ is $\frac{j+k}{2^{j+k+2}}$. The average cost of a such a letter is $\frac{\min(j,k)}{j+k}$, as there are $j+k$ letters in this block and the block contributes $\min(j,k)$ to the sum.

It can be analytically checked that

$$\sum_{j=1}^{\infty}\sum_{k=1}^{\infty}\frac{j+k}{2^{j+k+2}}\cdot\frac{\min(j,k)}{j+k}=\sum_{j=1}^{\infty}\sum_{k=1}^{\infty}\frac{\min(j,k)}{2^{j+k+2}}=\frac{1}{3}$$

We can conclude that $\mathbb{E}_{\mathcal{M}}(\textsc{LimAvg}(\pi_w))=\frac{1}{3}$ and, by Lemma 4, $\mathbb{E}_{\mathcal{M}}(\mathcal{A})=\frac{1}{3}$.

The bottom line is that even for such a simple automaton with only one strongly connected component consisting of three states (and two of them being symmetrical), the analysis is complicated. On the other hand, we conducted a simple Monte Carlo experiment in which we computed the value of this automaton on 10000 random words of length 2^{22} generated by $\mathcal{M}$, and observed that the obtained values are in the interval $[0.3283, 0.3382]$, with the average of 0.33336, which is a good approximation of the expected value 0.(3). This foreshadows our results for LimAvg-automata: we show that computing the expected value is, in general, impossible, but it is possible to approximate it with arbitrary precision. Furthermore, the small variation of the results is not accidental – we show that for strongly-connected LimAvg-automata, almost all words have the same value (which is equal to the expected value).

3.2 Irrationality of the distribution and the expected value

We argue that the exact values of $\mathbb{E}_{\mathcal{M}}(\mathcal{A})$ and $\mathbb{D}_{\mathcal{M},\mathcal{A}}(\lambda)$ for Sum-automata and LimAvg-automata may be irrational.

For the rest of this section we assume that the distribution of words is uniform. In the infinite case, this means that the Markov chain contains a single state where it loops over any letter with probability $\frac{1}{|\Sigma|}$, where Σ is the alphabet. In the finite case, this amounts to a terminating Markov chain with one regular state and one terminating state; it loops over any letter in the non-terminating state with probability $\frac{1}{|\Sigma|+1}$ or go to the terminating state over ϵ with probability $\frac{1}{|\Sigma|+1}$. Below we omit the Markov chain as it is fixed (for a given alphabet).

We define a Sum-automaton $\mathcal{A}$ (Figure 2) over the alphabet $\Sigma=\{a,\#\}$ such that $\mathcal{A}(w)=0$ if $w=a\#a^2\#\ldots\#a^{2^n}$ and $\mathcal{A}(w)\leq -1$ otherwise. Such an automaton basically picks a block with an inconsistency and verifies it. For example, if w contains a block $\#a^i\#a^j\#$, the automaton $\mathcal{A}$ first assigns -2 to each letter a and upon $\#$ it switches to the mode in which it assigns 1 to each letter a. Then, $\mathcal{A}$ returns the value $j-2\cdot i$. Similarly, we can encode the run that returns the value $2\cdot i-j$. Therefore, all the runs return 0 if and only if each block of a's is twice as long as the previous block. Finally, $\mathcal{A}$ checks whether the first block of a's has length 1 and returns -1 otherwise.

A word of the form $a\#a^2\#\ldots\#a^{2^n}$ has length $2^{n+1}+n-1$ and its probability is $3^{-(2^{n+1}+n)}$ (as the probability of any given word with n letters over a two-letters alphabet is $3^{-(n+1)}$). Therefore, the probability γ that a word is of the form $a\#a^2\#\ldots\#a^{2^n}$ is equal to $\sum_{n=0}^{\infty}3^{-(2^{n+1}+n)}$. Observe that γ written in base 3 has arbitrary long sequences of 0's and hence its representation is acyclic. Thus, γ is irrational. Observe that $\gamma=1-\mathbb{D}_{\mathcal{A}}(-1)$. Therefore, $\mathbb{D}_{\mathcal{A}}(-1)$ is irrational.

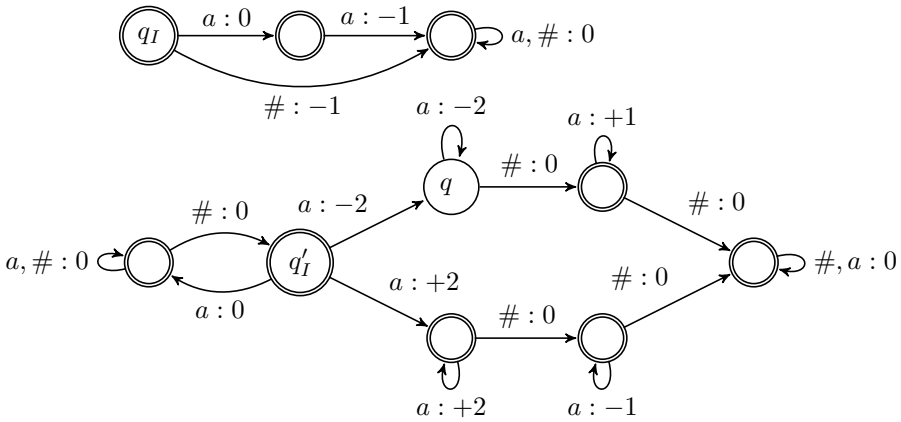

Figure 2 The automaton $\mathcal{A}$ from Section 3.2. States q_I and q'_I are initial and states but q are accepting. Any word that starts with $\#$ or aa has the value at most -1 because of a run that starts in q_I. For all other words, the runs starting in q_I have value -1. The accepting runs starting in q'_I have negative value only if the input word contains a (maximal) subword $a^i\#a^j$ such that $j \neq 2i$.

For the expected value, we construct $\mathcal{A}'$ such that for every word w we have $\mathcal{L}_{\mathcal{A}'}(w) = min(\mathcal{L}_{\mathcal{A}}(w), -1)$. This can be done by adding to $\mathcal{A}$ an additional initial state q_0, which starts an automaton that assigns to all words value -1. Observe that $\mathcal{A}$ and $\mathcal{A}'$ differ only on words w of the form $a\#a^2\#\ldots\#a^{2^n}$, where $\mathcal{A}(w) = 0$ and $\mathcal{A}'(w) = -1$. On all other words, both automata return the same values. Therefore, $\mathbb{E}(\mathcal{A}) - \mathbb{E}(\mathcal{A}') = \gamma$. It follows that at least one of the values $\mathbb{E}(\mathcal{A})$, $\mathbb{E}(\mathcal{A}')$ is irrational.

The same construction works for LimAvg. We take $\mathcal{A}$ (resp., $\mathcal{A}'$) and we convert it to a LimAvg-automaton $\mathcal{A}_\infty$ (resp., $\mathcal{A}'_\infty$) over $\Sigma' = \Sigma \cup \{\$\}$. The new letter \$ resets the automaton, i.e., $\mathcal{A}_\infty$ (resp., $\mathcal{A}'_\infty$) has transitions from the final states of $\mathcal{A}$ (resp., $\mathcal{A}'$) to its initial states labeled with \$. We can show that $1 - \mathbb{D}_{\mathcal{A}_\infty}(-1)$ and $\mathbb{E}(\mathcal{A}_\infty) - \mathbb{E}(\mathcal{A}'_\infty)$ over the uniform distribution are equal to γ and hence $\mathbb{D}_{\mathcal{A}_\infty}(-1)$ is irrational and one of the values $\mathbb{E}(\mathcal{A}_\infty)$, $\mathbb{E}(\mathcal{A}'_\infty)$ is irrational.

▶ **Theorem 5.** *There exist a* Sum*-automaton and a* LimAvg*-automaton whose distributions and expected values w.r.t. the uniform distribution are irrational.*

4 The exact value problems

In this section we consider the probabilistic questions for non-deterministic Sum-automata and LimAvg-automata, i.e., the problems of computing the exact values of the expected value $\mathbb{E}_{\mathcal{M}}(\mathcal{A})$ and the distribution $\mathbb{D}_{\mathcal{M},\mathcal{A}}(\lambda)$ w.r.t. a Markov chain $\mathcal{M}$ and an f-automaton $\mathcal{A}$. We showed that these values may be irrational. But one can perhaps argue that there might be some representation of irrational numbers that can be employed to avoid this problem. We prove that this is not the case by showing that computing the exact value to any representation with decidable equality of two numbers is not possible. The proof is by a (Turing) reduction from the quantitative universality problem for Sum-automata:

The quantitative universality problem for Sum*-automata*: Given a Sum-automaton with weights -1, 0 and 1, decide whether for all words w we have $\mathcal{L}_{\mathcal{A}}(w) \leq 0$.

The quantitative universality problem for Sum-automata is undecidable [21, 1].

We first discuss reductions to the probabilistic problems for Sum-automata. Consider an instance of the quantitative universality problem, which is a Sum-automaton $\mathcal{A}$. If there is a word w with the value greater than 0, then $\mathbb{P}(w) > 0$, and thus $\mathbb{D}_{\mathcal{A}}(0) < 1$. Otherwise, clearly $\mathbb{D}_{\mathcal{A}}(0) = 1$. Therefore, solving the universality problem amounts to computing whether the $\mathbb{D}_{\mathcal{A}}(0) = 1$, and thus the latter problem is undecidable. For the expected value, we construct a Sum-automaton $\mathcal{A}'$ such that for every word w we have $\mathcal{L}_{\mathcal{A}'}(w) = min(\mathcal{L}_{\mathcal{A}}(w), 0)$. Observe that $\mathbb{E}(\mathcal{A}) = \mathbb{E}(\mathcal{A}')$ if and only if for every word w we have $\mathcal{L}_{\mathcal{A}}(w) \leq 0$, i.e., the answer to the universality problem is YES. Therefore, there is no Turing machine, which given a Sum-automaton $\mathcal{A}$ computes $\mathbb{E}(\mathcal{A})$.

For LimAvg case, we construct a LimAvg-automaton $\mathcal{A}_\infty$ from the Sum-automaton $\mathcal{A}$, by connecting all accepting states (of $\mathcal{A}$) with all initial states by transitions of weight 0 labeled by an auxiliary letter #. For the expected value we construct $\mathcal{A}'_\infty$ from $\mathcal{A}'$ in the same way. Again, the distribution $\mathbb{D}_{\mathcal{A}_\infty}(0) = 1$ if and only if for all words we have $\mathcal{L}_{\mathcal{A}}(w) \leq 0$. Then, observe that $\mathbb{E}(\mathcal{A}_\infty) = \mathbb{E}(\mathcal{A}'_\infty)$ if and only if for every word w we have $\mathcal{L}_{\mathcal{A}}(w) \leq 0$. Therefore, there is no Turing machine computing the expected value or the distribution of a given LimAvg-automaton.

▶ **Theorem 6.** *The expected value and the distribution of (non-deterministic)* Sum*-automata (resp.,* LimAvg*-automata) are uncomputable even for the uniform distribution.*

4.1 Extrema automata

We discuss the distribution problem for Min-, Max-, Inf- and Sup-automata, where Min and Max return the minimal and the maximal (resp.) element of a finite sequence, and Inf and Sup return the minimal and the maximal (resp.) element of an infinite sequence. The expected value of an automaton can be easily computed based on the distribution as there are only finitely many possible values of a run (each possible value is a label of some transition).

▶ **Theorem 7.** *For* Min-, Max-, Inf- *and* Sup*-automata $\mathcal{A}$ and a Markov chain $\mathcal{M}$, the distribution problem can be solved in exponential time in $|\mathcal{A}|$ and polynomial time in $|\mathcal{M}|$.*

Proof. We discuss the case of $f =$ Inf as the other cases are similar. Consider an Inf-automaton $\mathcal{A}$. For each weight x of $\mathcal{A}$, we can construct a (non-deterministic) ω-automaton $\mathcal{A}_x$ that accepts only words of value greater than x – we take $\mathcal{A}$, remove the transitions of weight at most x, and drop all the weights. Therefore, the set of words with the value greater than x is regular, and hence it is measurable. We can compute its probability p_x w.r.t. $\mathcal{M}$ in exponential time in $|\mathcal{A}|$ and polynomial in $|\mathcal{M}|$ [3]. Note that $p_x = 1 - \mathbb{D}_{\mathcal{M},\mathcal{A}}(x)$. ◀

5 The approximation problems

We start the discussion on the approximation problems by showing a hardness result that holds for a wide range of value functions. We say that a function is 0-preserving if its value is 0 whenever the input consists only of 0s. Notice that functions such as Sum, LimAvg, Min, Max, Inf, Sup and virtually all the functions from the literature [8] are 0-preserving. The hardness results follow from the fact that accepted words have finite values, which we can force to be 0, while words without accepting runs have infinite values.

The answers in the approximation problems are numbers and to study the lower bounds, we consider their decision variants, called the *separation problems*. In these variants, the input is enriched with numbers a, b such that $b - a > 2\epsilon$ and the instance is such that

$\mathbb{E}_{\mathcal{M}}(\mathcal{A}) \notin [a, b]$ (resp. $\mathbb{D}_{\mathcal{M}}(\mathcal{A}) \notin [a, b]$), and the question is whether $\mathbb{E}_{\mathcal{M}}(\mathcal{A}) < a$ (resp. $\mathbb{D}_{\mathcal{M}}(\mathcal{A}) < a$). Note that having an algorithm computing one of the approximate problems (for the distribution or the expected value), we can use it to decide the separation question. Conversely, using the separation problem as an oracle, we can perform binary search on the domain to compute solve the corresponding approximation problem in polynomial time.

▶ **Theorem 8.** *For a 0-preserving function f, the separation problems for non-deterministic f-automata are* PSpace*-hard.*

Total automata. Theorem 8 gives us a general hardness result, which is due to accepting conditions rather than values returned by weighted automata. In the following, we focus on weights and we assume that weighted automata are *total*, i.e., they accept all the words (resp., almost all the words in the infinite-word case). For Sum-automata under the totality assumption, the approximate probabilistic questions become #P-complete.

▶ **Theorem 9.** *The approximate expected value and the approximate distribution questions for non-deterministic total* Sum*-automata are* #P*-complete.*

The lower bound can be obtained by a reduction from the problem of counting the number of satisfying assignment of a given propositional formula φ in Conjunctive Normal Form (CNF) [25, 23], which is #P-complete. We construct an automaton that, for a formula with n variables, accepts only words of length n that encode valuations that make the formula satisfied. It follows that the expected value of the automaton equals $3^{-(n+1)} \cdot C$, where $3^{-(n+1)}$ is the probability of generating a word of length n under uniform distribution and C is the number of variable assignments satisfying φ.

The upper bound follows the idea that the probability that $\mathcal{M}^T$ emits a word of length greater than n decreases exponentially with n. This means that there is N of polynomial size such that the distribution (resp., the expected value) of $\mathcal{A}$ and the distribution (resp., the expected value) of $\mathcal{A}$ over words up to length N differ by less than ϵ. Based on this, we can build a Turing machine that imitates the distribution of $\mathcal{M}^T$ over words up to length N with its non-deterministic computations.

We show that the approximation problem for LimAvg-automata is PSpace-hard over the class of total automata.

▶ **Theorem 10.** *The separation problems for non-deterministic total* LimAvg*-automata are* PSpace*-hard.*

Proof. Given a non-deterministic finite-word automaton $\mathcal{A}$, we construct an infinite-word LimAvg-automaton $\mathcal{A}_\infty$ from $\mathcal{A}$ in the following way. We introduce an auxiliary symbol # and we add transitions labeled by # between any final state of $\mathcal{A}$ and any initial state of $\mathcal{A}$. Then, we label all transitions of $\mathcal{A}_\infty$ with 0. Finally, we connect all non-accepting states of $\mathcal{A}$ with an auxiliary state q_{sink}, which is a sink state with all transitions of weight 1. The automaton $\mathcal{A}_\infty$ is total.

Observe that if $\mathcal{A}$ is universal, then $\mathcal{A}_\infty$ has a run of value 0 on every word. Otherwise, if $\mathcal{A}$ rejects a word w, then upon reading a subword $\#w\#$, the automaton $\mathcal{A}_\infty$ reaches q_{sink}, i.e., the value of the whole word is 1. Almost all words contain an infix $\#w\#$ and hence almost all words have value 1. ◀

6 Approximating LimAvg-automata in exponential time

The case of LimAvg is significantly more complex than the other cases. First, we restrict our attention to *recurrent* LimAvg-automata and the uniform distribution over infinite words. Then, we comment on the extension to all distributions given by Markov chains. Finally, we show the proof for all LimAvg-automata over probability measures given by Markov chains.

6.1 Recurrent automata

A non-deterministic LimAvg-automaton $\mathcal{A} = (\Sigma, Q, Q_0, \delta)$ is recurrent if and only if for every set $S \subseteq Q$ such that $|S| = 1$ or $\widehat{\delta}(Q_0, w) = S$ for some word w, there is a finite word u such that $\widehat{\delta}(S, u) = Q_0$.

For every $\mathcal{A}$, which is strongly connected as a graph, there exists a set of initial states T with which it becomes recurrent. Moreover, the probability of words accepted by $\mathcal{A}$ is either 0 or 1. Indeed, consider $\mathcal{A}$ as an unweighted ω-automaton and construct a deterministic ω-automaton $\mathcal{A}^D$ through the power-set construction applied to $\mathcal{A}$. Note that $\mathcal{A}^D$ has a single bottom strongly-connected component (BSCC) and Q_0 belongs to that component. Conversely, for any strongly connected automaton $\mathcal{A}$, if Q_0 belongs to the BSCC of $\mathcal{A}^D$, then $\mathcal{A}$ is recurrent. Moreover, since $\mathcal{A}^D$ has a single BSCC, for almost all words, all runs end up in that BSCC and hence the probability of the set of words having any infinite run in $\mathcal{A}$ is either 0 or 1.

6.2 Nearly-deterministic approximations

While words are generated by a Markov chain letter by letter, the run on that word can be defined only when the complete word is generated. This precludes application of standard techniques for probabilistic verification, which relies on the fact that the word and the run on it are generated simultaneously [26, 12, 3].

Key ideas. Our main idea is to change the non-determinism to *bounded look-ahead*. This must be inaccurate, as the expected value of a deterministic automaton with bounded look-ahead is always rational, whereas Theorem 5 shows that the values of non-deterministic automata may be irrational. Nevertheless, we show that bounded look-ahead is sufficient to *approximate* the probabilistic questions for recurrent automata (Lemma 11). Furthermore, the approximation can be done effectively (Lemma 14), which in turn gives us an exponential-time approximation algorithm for recurrent automata (Lemma 15).

Jumping runs. Let $k > 0$. A *k-jumping run* ξ of $\mathcal{A}$ on a word w is an infinite sequence of states such that for every i there is a run π of $\mathcal{A}$ on w such that $\xi[ki, k(i+1) - 1] = \pi[ki, k(i+1) - 1]$.

A *block* of a k-jumping run is a sequence $\xi[ki, k(i+1) - 1]$ for some i; positions $k, 2k, \ldots$ are *jumps*, where the sequence ξ need not obey the transition relation of $\mathcal{A}$.

The cost C of a transition of a k-jumping run ξ within a block is defined as usual, while the cost of a jump is defined as the minimal weight of $\mathcal{A}$. The value of a k-jumping run ξ is defined as the limit average computed for such costs.

Optimal and block-deterministic jumping runs. We say that a k-jumping run ξ on a word w is *optimal* if its value is the infimum over values of all k-jumping runs on w. We show that optimal k-jumping runs can be constructed nearly deterministically, i.e., only looking ahead to see the whole current block.

For every $S \subseteq Q$ and $u \in \Sigma^k$ we fix a run $\xi_{S,u}$ on u starting in one of states of S, which has the minimal average weight. Then, given a word $w \in \Sigma^\omega$, we define a k-jumping run ξ as follows. We divide w into k-letter blocks $u_1, u_2, \ldots$ and we put $\xi = \xi_{S_0,u_1}\xi_{S_1,u_2}\ldots$, where $S_0 = \{q_0\}$ and for $i > 0$, S_i is the set of states reachable from q_0 on the word $u_1 \ldots u_i$. The run ξ_o is a k-jumping run and it is indeed optimal. We call such runs *block-deterministic* – they can be constructed based on finite memory – the set of reachable states S_i and the current block of the input word.

Since all runs of $\mathcal{A}$ are in particular k-jumping runs, the value of (any) optimal k-jumping run on w is less or equal to $\mathcal{A}(w)$. We show that for recurrent LimAvg-automata, the values of k-jumping runs on w converge to $\mathcal{A}(w)$ as k tends to infinity. To achieve this, we construct a run of $\mathcal{A}$ which tries to "follow" a given jumping run, i.e., after most all of the jumps it is able to synchronize with the jumping run quickly.

▶ **Lemma 11.** *Let $\mathcal{A}$ be a recurrent* LimAvg*-automaton. For every $\epsilon > 0$, there exists k such that for almost all words w, the value $\mathcal{A}(w)$ and the value an optimal k-jumping run on w differ by at most ϵ. The value k is doubly-exponential in $|\mathcal{A}|$ and polynomial in $\frac{1}{\epsilon}$.*

6.3 Random variables

Given a recurrent LimAvg-automaton $\mathcal{A}$ and $k > 0$, we define a function $g[k] : \Sigma^\omega \to \mathbb{R}$ such that $g[k](w)$ is the value of some optimal k-jumping run ξ_o on w. We can pick ξ_o to be block-deterministic and hence $g[k]$ corresponds to a Markov chain $M[k]$. More precisely, we define $M[k]$ labeled by Σ^k such that for every word w, the limit average of the path in $M[k]$ labeled by blocks of w (i.e., blocks $w[1,k]w[k+1,2k]\ldots$) equals $g[k](w)$. Moreover, the distribution of blocks Σ^k is uniform and hence $M[k]$ corresponds to $g[k]$ over the uniform distribution over Σ. The Markov chain $M[k]$ is a labeled weighted Markov chain [15], such that its states are all subsets of Q, the set of states of $\mathcal{A}$. For each state $S \subseteq Q$ and $u \in \Sigma^k$, the Markov chain $\mathcal{M}$ has an edge $(S, \widehat{\delta}(S,u))$ of probability $\frac{1}{|\Sigma|^k}$. The weight of an edge (S,S') labeled by u is the minimal average of weights of any run from some state of S to some state of S' over the word w.

We have the following:

▶ **Lemma 12.** *Let $\mathcal{A}$ be a recurrent* LimAvg*-automaton and $k > 0$. (1) The functions $g[k]$ and $\mathcal{L}_\mathcal{A}$ are random variables. (2) For almost all words w we have $g[k](w) = \mathbb{E}(g[k])$ and $\mathcal{L}_\mathcal{A}(w) = \mathbb{E}(\mathcal{L}_\mathcal{A})$.*

Proof. Since $\mathcal{A}$ is recurrent, $M[k]$ has a single BSCC and hence $M[k]$ and $g[k]$ return the same value for almost all words [15]. This implies that the preimage through $g[k]$ of each set has measure 0 or 1, and hence $g[k]$ is measurable [14]. Lemma 11 implies that (measurable functions) $g[k]$ converge to $\mathcal{L}_\mathcal{A}$ with probability 1, and hence $\mathcal{L}_\mathcal{A}$ is measurable [14]. As the limit of $g[k]$, $\mathcal{L}_\mathcal{A}$ also has the same value for almost all words. ◀

▶ **Remark 13.** *The automaton $\mathcal{A}$ from the proof of Theorem 5 is recurrent (it resets after each \$), so the value of $\mathcal{A}$ on almost all words is irrational. Yet, for every ultimately periodic word vw^ω, the value of $\mathcal{A}$ is rational. This means that while the expected value is realised by almost all words, it is not realised by any ultimately periodic word.*

6.4 Approximation algorithms

We show that the expected value of $g[k]$ can be efficiently approximated. The approximation is exponential in the size of $\mathcal{A}$, but only logarithmic in k (which is doubly-exponential due to Lemma 11).

▶ **Lemma 14.** *Given a recurrent* LimAvg*-automaton $\mathcal{A}$, $k = 2^l$ and $\epsilon > 0$, the expected value $\mathbb{E}(g[k])$ can be approximated up to ϵ in exponential time in $|\mathcal{A}|$, logarithmic time in k and polynomial time in $\frac{1}{\epsilon}$.*

Lemma 11 and Lemma 14 imply the following:

▶ **Lemma 15.** *Given a recurrent* LimAvg*-automaton $\mathcal{A}$, Markov chain $\mathcal{M}$, $\epsilon > 0$ and $\lambda \in \mathbb{Q}$, we can compute ϵ-approximations of the distribution $\mathbb{D}_{\mathcal{M},\mathcal{A}}(\lambda)$ and the expected value $\mathbb{E}_{\mathcal{M}}(\mathcal{A})$ in exponential time in $|\mathcal{A}|$ and polynomial time in $|\mathcal{M}|$ and $\frac{1}{\epsilon}$.*

Proof. For uniform distributions, by Lemma 11, for every $\epsilon > 0$, there exists k such that $|\mathbb{E}(\mathcal{A}) - \mathbb{E}(g[k])| \leq \frac{\epsilon}{2}$. The value k is doubly-exponential in $|\mathcal{A}|$ and polynomial in $\frac{1}{\epsilon}$. Then, Lemma 14, we can compute γ such that $|\gamma - \mathbb{E}(g[k])| \leq \frac{\epsilon}{2}$ in exponential time in $|\mathcal{A}|$ and polynomial in $\frac{1}{\epsilon}$. Thus, γ differs from $\mathbb{E}(\mathcal{A})$ by at most ϵ. Since almost all words have the same value, we can approximate $\mathbb{D}_{\mathcal{A}}(\lambda)$ by comparing λ with γ, i.e., 1 is an ϵ-approximation of $\mathbb{D}_{\mathcal{A}}(\lambda)$ if $\lambda \leq \gamma$, and otherwise 0 is an ϵ-approximation of $\mathbb{D}_{\mathcal{A}}(\lambda)$.

The case of the nonuniform distributions can be solved similarly, by encoding the Markov chain in the automaton. ◀

We lift Lemma 15 from recurrent to all LimAvg-automata and formally show that $\mathcal{L}_{\mathcal{A}} : \Sigma^\omega \to \mathbb{R}$ is measurable. To do so, we take the product of a given automaton and Markov chain and observe that its BSCC correspond to recurrent automata. A careful analysis allows to compute the values for the whole automata based on the values for the BSCC.

▶ **Theorem 16.** *(1) For a non-deterministic* LimAvg*-automaton $\mathcal{A}$ the function $\mathcal{L}_{\mathcal{A}} : \Sigma^\omega \to \mathbb{R}$ is measurable. (2) Given a non-deterministic* LimAvg*-automaton $\mathcal{A}$, Markov chain $\mathcal{M}$, $\epsilon > 0$, and $\lambda \in \mathbb{Q}$, we can ϵ-approximate the distribution $\mathbb{D}_{\mathcal{M},\mathcal{A}}(\lambda)$ and the expected value $\mathbb{E}(\mathcal{A})$ in exponential time in $|\mathcal{A}|$ and polynomial time in $|\mathcal{M}|$ and $\frac{1}{\epsilon}$.*

7 Determinising and approximating LimAvg-automata

For technical simplicity, we assume that the distribution of words is uniform. However, the results presented here extend to all distributions given by Markov chains.

Recall that for the LimAvg automata, the value of almost all words, whose optimal runs end up in the same SSC, is the same. This means that there is a finite set of values (not greater than the number of SSCs of the automaton) such that almost all the words have their values in this set.

LimAvg-automata are not determinisable [8]. We say that a non-deterministic LimAvg-automaton $\mathcal{A}$ is *weakly determinisable* if there is a deterministic LimAvg-automaton B such that A and B have the same value over almost all the words. From [9] we know that deterministic automata return rational values for almost all the words, so not all LimAvg-automata are weakly determinisable. However, we can show the following.

▶ **Theorem 17.** *A* LimAvg*-automaton $\mathcal{A}$ is weakly determinisable if and only if it returns rational values for almost all words.*

Proof sketch. Assume an automaton $\mathcal{A}$ with SSCs $C_1, \ldots, C_m$. For each i let v_i be defined as the expected value of $\mathcal{A}$ when its set of initial states is C_i and the run is bounded to stay in C_i. If $\mathcal{A}$ has no such runs for some C_i, then $v_i = \infty$.

We now construct a deterministic automaton B with rational weights using the standard power-set construction. We define the cost function such that the cost of any transition from

a state Y is the minimal value v_i such that v_i is rational and Y contains a state from C_i. If there are no such v_i, then we set the cost to the maximal cost of $\mathcal{A}$. Roughly speaking, B tracks in which SSCs A can be and the weight corresponds to the SSC with the lowest value.

To see that B weakly determinises A observe that for almost all words w, a run with the lowest value over w ends in some SSC and its value then equals the expected value of this component, which is rational as the value of this word is rational. ◀

A straightforward corollary is that every non-deterministic LimAvg-automaton can be weakly determinised by an LimAvg-automaton with real weights.

Theorem 17 does not provide an implementable algorithm for weakly-determinisation, because of the hardness of computing the values v_i. It is possible, however, to approximate this automaton. We say that a deterministic LimAvg-automaton B *ϵ-approximates* $\mathcal{A}$ if for almost every word w we have that $\mathcal{L}_B(w) \in [\mathcal{L}_{\mathcal{A}}(w) - \epsilon, \mathcal{L}_{\mathcal{A}}(w) + \epsilon]$.

▶ **Theorem 18.** *For every $\epsilon > 0$ and a non-deterministic* LimAvg*-automaton $\mathcal{A}$, one can compute in exponential time a deterministic* LimAvg*-automaton that ϵ-approximates $\mathcal{A}$.*

The proof of this theorem is similar to the proof of Theorem 17, except now it is enough to approximate the values v_i, which can be done in exponential time.

References

1. Shaull Almagor, Udi Boker, and Orna Kupferman. What's decidable about weighted automata? In *ATVA*, pages 482–491. LNCS 6996, Springer, 2011.
2. Benjamin Aminof, Orna Kupferman, and Robby Lampert. Reasoning about online algorithms with weighted automata. *ACM Transactions on Algorithms (TALG)*, 6(2):28, 2010.
3. Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.
4. Christel Baier, Stefan Kiefer, Joachim Klein, Sascha Klüppelholz, David Müller, and James Worrell. Markov chains and unambiguous büchi automata. In *CAV 2016*, pages 23–42. Springer, 2016.
5. Michael Benedikt, Gabriele Puppis, and Cristian Riveros. Regular repair of specifications. In *LICS 2011*, pages 335–344, 2011.
6. Udi Boker and Thomas A. Henzinger. Approximate determinization of quantitative automata. In *FSTTCS 2012*, pages 362–373. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.
7. Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. A survey of stochastic games with limsup and liminf objectives. In *ICALP 2009*, pages 1–15, 2009.
8. Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Quantitative languages. *ACM TOCL*, 11(4):23, 2010.
9. Krishnendu Chatterjee, Thomas A. Henzinger, and Jan Otop. Quantitative automata under probabilistic semantics. In *LICS 2016*, pages 76–85. ACM, 2016.
10. Krishnendu Chatterjee, Mickael Randour, and Jean-François Raskin. Strategy synthesis for multi-dimensional quantitative objectives. In *CONCUR 2012*, pages 115–131, 2012.
11. Edmund Clarke, Thomas Henzinger, Helmut Veith, and Roderick. Bloem. *Handbook of Model Checking*. Springer International Publishing, 2016.
12. Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, 1995.
13. Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of Weighted Automata*. Springer, 1st edition, 2009.
14. William Feller. *An introduction to probability theory and its applications*. Wiley, 1971.

15 Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes*. Springer, 1996.

16 Charles Miller Grinstead and James Laurie Snell. *Introduction to probability*. American Mathematical Soc., 2012.

17 Ashutosh Gupta, Corneliu Popeea, and Andrey Rybalchenko. Predicate abstraction and refinement for verifying multi-threaded programs. In *POPL 2011*, pages 331–344, 2011.

18 Thomas A. Henzinger and Jan Otop. Model measuring for discrete and hybrid systems. *Nonlinear Analysis: Hybrid Systems*, 23:166–190, 2017.

19 Andrew Hinton, Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM: A tool for automatic verification of probabilistic systems. In *TACAS 2006*, pages 441–444, 2006.

20 Alexander Kechris. *Classical descriptive set theory*, volume 156. Springer Science & Business Media, 2012.

21 Daniel Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *Int. J. Algebr. Comput.*, 4(3):405–426, 1994. doi:10.1142/S0218196794000063.

22 Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Quantitative analysis with the probabilistic model checker PRISM. *Electr. Notes Theor. Comput. Sci.*, 153(2):5–31, 2006.

23 Christos H Papadimitriou. *Computational complexity*. Wiley, 2003.

24 Mickael Randour, Jean-François Raskin, and Ocan Sankur. Percentile queries in multi-dimensional markov decision processes. In *CAV 2015*, pages 123–139, 2015.

25 Leslie G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8(2):189–201, 1979.

26 Moshe Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *FOCS 1985*, pages 327–338. IEEE Computer Society, 1985.