Probabilistic Weighted Automata

Krishnendu Chatterjee[1], Laurent Doyen[2], and Thomas A. Henzinger[3]

[1] University of California, Santa Cruz
[2] Université Libre de Bruxelles, Belgium
[3] EPFL, Lausanne, Switzerland

Abstract. Nondeterministic weighted automata are finite automata with numerical weights on transitions. They define quantitative languages L that assign to each word w a real number $L(w)$. The value of an infinite word w is computed as the maximal value of all runs over w, and the value of a run as the maximum, limsup, liminf, limit average, or discounted sum of the transition weights. We introduce probabilistic weighted automata, in which the transitions are chosen in a randomized (rather than nondeterministic) fashion. Under almost-sure semantics (resp. positive semantics), the value of a word w is the largest real v such that the runs over w have value at least v with probability 1 (resp. positive probability).
We study the classical questions of automata theory for probabilistic weighted automata: emptiness and universality, expressiveness, and closure under various operations on languages. For quantitative languages, emptiness and universality are defined as whether the value of some (resp. every) word exceeds a given threshold. We prove some of these questions to be decidable, and others undecidable. Regarding expressive power, we show that probabilities allow us to define a wide variety of new classes of quantitative languages, except for discounted-sum automata, where probabilistic choice is no more expressive than nondeterminism. Finally, we give an almost complete picture of the closure of various classes of probabilistic weighted automata for the following pointwise operations on quantitative languages: max, min, sum, and numerical complement.

1 Introduction

In formal design, specifications describe the set of correct behaviours of a system. An implementation satisfies a specification if all its behaviours are correct. If we view a behaviour as a word, then a specification is a language, i.e., a set of words. Languages can be specified using finite automata, for which a large number of results and techniques are known; see [11, 13]. We call them *boolean languages* because a given behaviour is either good or bad according to the specification. Boolean languages are useful to specify functional requirements.

In a generalization of this approach, we consider *quantitative languages*, where each word is assigned a real number. The value of a word can be interpreted as the amount of some resource (e.g., memory or power) needed to produce it, or as a quality measurement for the corresponding behaviour [4, 5]. Therefore, quantitative languages are useful to specify non-functional requirements such as resource constraints, reliability properties, or levels of quality (such as quality of service).

Quantitative languages can be defined using (nondeterministic) weighted automata, i.e., finite automata with numerical weights on transitions [9, 10]. In [7], we studied quantitative languages of infinite words and defined the value of an infinite word w as the maximal value of all runs of an automaton over w (if the automaton is nondeterministic, then there may be many runs over w). The

value of a run r is a function of the infinite sequence of weights that appear along r. There are several natural functions to consider, such as Sup, LimSup, LimInf, limit average, and discounted sum of weights. For example, peak power consumption can be modeled as the maximum of a sequence of weights representing power usage; energy use, as a discounted sum; average response time, as a limit average [3, 4].

In this paper, we consider *probabilistic* weighted automata as generator of quantitative languages. In such automata, nondeterministic choice is replaced by probability distributions on successor states. The value of an infinite word w is defined to be the maximal value v such that the set of runs over w with value at least v has either positive probability (*positive semantics*), or probability 1 (*almost-sure semantics*). The probabilistic Büchi and coBüchi automata of [2] are a special case of probabilistic weighted automata with weights 0 and 1 only (and the value of an infinite run computed as LimSup or LimInf, respectively). We are not aware of any other model combining probabilities and weights for the specification of quantitative languages.

We study fundamental questions about the expressive power, closure properties, and the emptiness and universality problems for probabilistic weighted automata.

First, we compare the expressiveness of the various classes of probabilistic and nondeterministic weighted automata over infinite words. For LimSup, LimInf, and limit average, we show that a wide variety of new classes of quantitative languages can be defined using probabilities, which are not expressible using nondeterminism. Our results rely on reachability properties of closed recurrent sets in Markov chains. For discounted sum, we show that probabilistic weighted automata under the positive semantics have the same expressive power as nondeterministic weighted automata, while under the almost-sure semantics, they have the same expressive power as weighted automata with universal branching, where the value of a word is the minimal (instead of maximal) value of all runs. The question of whether the positive semantics of weighted limit-average automata is more expressive than nondeterminism, remains open.

Second, we give an almost complete picture of the closure of probabilistic weighted automata under the pointwise operations of maximum, minimum, and sum for quantitative languages. Note that closure under max trivially holds for the positive semantics, and closure under min for the almost-sure semantics. We also define the *complement* L^c of a quantitative language L by $L^c(w) = 1 - L(w)$ for all words w. Only LimSup-automata under positive semantics and LimInf-automata under almost-sure semantics are closed under all four operations; these results extend corresponding results for the boolean (i.e., non-quantitative) case [1]. To establish the closure properties of limit-average automata, we characterize the expected limit-average reward of Markov chains. Our characterization answers all closure questions except for the language sum in the case of positive semantics, which we leave open. Note that expressiveness results and closure properties are tightly connected. For instance, because they are closed under max, the LimInf-automata with positive semantics can be reduced to LimInf-automata with almost-sure semantics and to LimSup-automata with positive semantics; and because they are not closed under complement, the LimSup-automata with almost-sure semantics and LimInf-automata with positive semantics have incomparable expressive powers.

Third, we investigate the emptiness and universality problems for probabilistic weighted automata, which ask to decide if some (resp. all) words have a value above a given threshold. Using our expressiveness results, as well as [1, 6], we establish some decidability and undecidability results for Sup, LimSup, and LimInf automata; in particular, emptiness and universality are undecidable for LimSup-automata with positive semantics and for LimInf-automata with almost-sure semantics, while the question is open for the emptiness of LimInf-automata with positive semantics and for

the universality of LimSup-automata with almost-sure semantics. We also prove the decidability of emptiness for probabilistic discounted-sum automata with positive semantics, while the universality problem is as hard as for the nondeterministic discounted-sum automata, for which no decidability result is known. We leave open the case of limit average.

2 Definitions

A quantitative language over a finite alphabet Σ is a function $L : \Sigma^\omega \to \mathbb{R}$. A boolean language (or a set of infinite words) is a special case where $L(w) \in \{0, 1\}$ for all words $w \in \Sigma^\omega$. Nondeterministic weighted automata define the value of a word as the maximal value of a run [7]. In this paper, we study probabilistic weighted automata as generator of quantitative languages.

Value functions. We consider the following value functions $\mathsf{Val} : \mathbb{Q}^\omega \to \mathbb{R}$ to define quantitative languages. Given an infinite sequence $v = v_0 v_1 \dots$ of rational numbers, define

- $\mathsf{Sup}(v) = \sup\{v_n \mid n \geq 0\}$;
- $\mathsf{LimSup}(v) = \limsup_{n\to\infty} v_n = \lim_{n\to\infty} \sup\{v_i \mid i \geq n\}$;
- $\mathsf{LimInf}(v) = \liminf_{n\to\infty} v_n = \lim_{n\to\infty} \inf\{v_i \mid i \geq n\}$;
- $\mathsf{LimAvg}(v) = \liminf_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} v_i$;
- For $0 < \lambda < 1$, $\mathsf{Disc}_\lambda(v) = \sum_{i=0}^{\infty} \lambda^i \cdot v_i$;

Given a finite set S, a *probabilistic distribution* over S is a function $f : S \to [0, 1]$ such that $\sum_{s\in S} f(s) = 1$. We denote by $\mathcal{D}(S)$ the set of all probabilistic distributions over S.

Probabilistic weighted automata. A *probabilistic weighted automaton* is a tuple $A = \langle Q, \rho_I, \Sigma, \delta, \gamma\rangle$ where:

- Q is a finite set of states;
- $\rho_I \in \mathcal{D}(Q)$ is the initial distribution;
- Σ is a finite alphabet;
- $\delta : Q \times \Sigma \to \mathcal{D}(Q)$ is a probabilistic transition function;
- $\gamma : Q \times \Sigma \times Q \to \mathbb{Q}$ is a weight function.

We can define a *non-probabilistic* automaton from A by ignoring the probability values, and saying that q is initial if $\rho_I(q) > 0$, and (q, σ, q') is an edge of A if $\delta(q, \sigma)(q') > 0$. The automaton A is *deterministic* if $\rho_I(q_I) = 1$ for some $q_I \in Q$, and for all $q \in Q$ and $\sigma \in \Sigma$, there exists $q' \in Q$ such that $\delta(q, \sigma)(q') = 1$.

A *run* of A over a finite (resp. infinite) word $w = \sigma_1\sigma_2 \dots$ is a finite (resp. infinite) sequence $r = q_0\sigma_1 q_1\sigma_2 \dots$ of states and letters such that (i) $\rho_I(q_0) > 0$, and (ii) $\delta(q_i, \sigma_{i+1}, q_{i+1}) > 0$ for all $0 \leq i < |w|$. We denote by $\gamma(r) = v_0 v_1 \dots$ the sequence of weights that occur in r where $v_i = \gamma(q_i, \sigma_{i+1}, q_{i+1})$ for all $0 \leq i < |w|$.

The probability of a finite run $r = q_0\sigma_1 q_1\sigma_2 \dots \sigma_k q_k$ over a finite word $w = \sigma_1 \dots \sigma_k$ is $\mathbb{P}^A(r) = \rho_I(q_0).\prod_{i=1}^{k} \delta(q_{i-1}, \sigma_i)(q_i)$. For each $w \in \Sigma^\omega$, the function $\mathbb{P}^A(\cdot)$ defines a unique probability measure over Borel sets of (infinite) runs of A over w.

Given a value function $\mathsf{Val} : \mathbb{Q}^\omega \to \mathbb{R}$, we say that the probabilistic Val-automaton A generates the quantitative languages defined for all words $w \in \Sigma^\omega$ by $L_A^{=1}(w) = \sup\{\eta \mid \mathbb{P}^A(\{r \in \mathsf{Run}^A(w) \mid \mathsf{Val}(\gamma(r)) \geq \eta\}) = 1\}$ under the almost-sure semantics, and $L_A^{>0}(w) = \sup\{\eta \mid \mathbb{P}^A(\{r \in \mathsf{Run}^A(w) \mid \mathsf{Val}(\gamma(r)) \geq \eta\}) > 0\}$ under the positive semantics. For non-probabilistic automata, the value of a word is either the maximal value of the runs (i.e., $L_A^{\max}(w) = \sup\{\mathsf{Val}(\gamma(r)) \mid r \in \mathsf{Run}^A(w)\}$ for all $w \in \Sigma^\omega$) and the automaton is then called *nondeterministic*, or the minimal value of the runs, and the automaton is then called *universal*.

Note that Büchi and coBüchi automata ([2]) are special cases of respectively LimSup- and LimInf-automata, where all weights are either 0 or 1.

Notations. The first letter in acronyms for classes of automata can be N(ondeterministic), D(eterministic), U(niversal), POS for the language in the positive semantics, or AS for the language in the almost-sure semantics. We use the notations $^{\mathrm{D}}_{\mathrm{N}}$ to denote the classes of automata whose deterministic version has the same expressiveness as their nondeterministic version. When the type of an automaton A is clear from the context, we often denote its language simply by $L_A(\cdot)$ or even $A(\cdot)$, instead of $L_A^{=1}$, $L_A^{\max}$, etc.

Reducibility. A class $\mathcal{C}$ of weighted automata is *reducible* to a class $\mathcal{C}'$ of weighted automata if for every $A \in \mathcal{C}$ there exists $A' \in \mathcal{C}'$ such that $L_A = L_{A'}$, i.e. $L_A(w) = L_{A'}(w)$ for all words w. Reducibility relationships for (non)deterministic weighted automata are given in [7].

Composition. Given two quantitative languages $L, L' : \Sigma^\omega \to \mathbb{R}$, we denote by $\max(L, L')$ (resp. $\min(L, L')$ and $L + L'$) the quantitative language that assigns $\max\{L(w), L'(w)\}$ (resp. $\min\{L(w), L'(w)\}$ and $L(w) + L'(w)$) to each word $w \in \Sigma^\omega$. The language $1 - L$ is called the *complement* of L. The $\max$, $\min$ and complement operators for quantitative languages generalize respectively the union, intersection and complement operator for boolean languages. The closure properties of (non)deterministic weighted automata are given in [6].

Remark. We sometimes use automata with weight functions $\gamma : Q \to \mathbb{Q}$ that assign a weight to states instead of transitions. This is a convenient notation for weighted automata in which from each state, all outgoing transitions have the same weight. In pictorial descriptions of probabilistic weighted automata, the transitions are labeled with probabilities, and states with weights.

3 Expressive Power of Probabilistic Weighted Automata

We complete the picture given in [7] about reducibility for nondeterministic weighted automata, by adding the relations with probabilistic automata. The results for LimInf, LimSup, and LimAvg are summarized in Fig. 1s, and for Sup- and Disc-automata in Theorems 1 and 5.

3.1 Probabilistic Sup-automata

Like for probabilistic automata over finite words, the quantitative languages definable by probabilistic and (non)deterministic Sup-automata coincide.

Theorem 1. POSSUP *and* ASSUP *are reducible to* DSUP.

Proof. It is easy to see that POSSUP-automata define the same language when interpreted as NSUP-automata, and the same holds for ASSUP and USUP. The result then follows from [7, Theorem 9]. ■

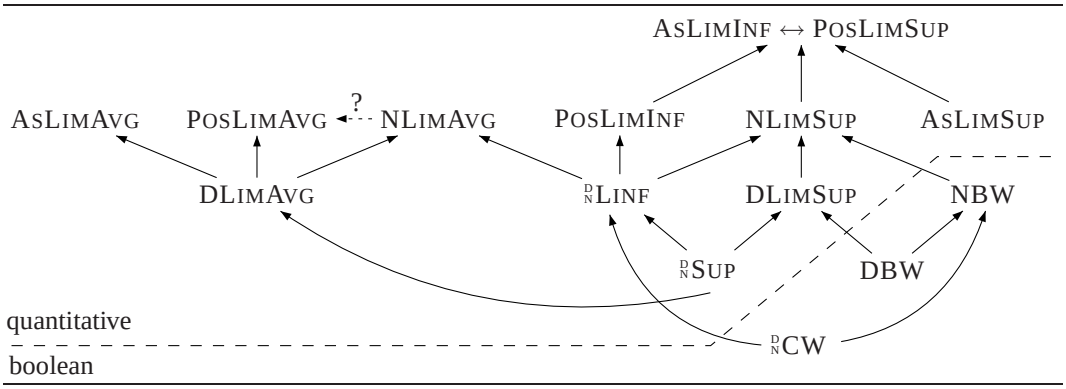

Fig. 1. Reducibility relation. $\mathcal{C}$ is reducible to $\mathcal{C}'$ if $\mathcal{C} \to \mathcal{C}'$. Classes that are not connected by an arrow are incomparable. Reducibility for the dashed arrow is open.

3.2 Probabilistic LimAvg-automata

Many of our results would consider *Markov chains* and *closed recurrent states* in Markov chains. A *Markov chain* $M = (S, E, \delta)$ consists of a finite set S of states, a set E of edges, and a probabilistic transition function $\delta : S \to \mathcal{D}(S)$. For all $s, t \in S$, there is an edge $(s, t) \in E$ iff $\delta(s)(t) > 0$. A *closed recurrent set* C of states in M is a bottom strongly connected set of states in the graph (S, E). We will use the following two key properties of closed recurrent states.

1. *Property 1.* Given a Markov chain M, and a start state s, with probability 1, the set of closed recurrent states is reached from s in finite time. Hence for any $\epsilon > 0$, there exists k_0 such that for all $k > k_0$, for all starting state s, the set of closed recurrent states are reached with probability at least $1 - \epsilon$ in k steps.
2. *Property 2.* If a closed recurrent set C is reached, and the limit of the expectation of the average weights of C is α, then for all $\epsilon > 0$, there exists a k_0 such that for all $k > k_0$ the expectation of the average weights for k steps is at least $\alpha - \epsilon$.

The above properties are the basic properties of finite state Markov chains and closed recurrent states [12].

Lemma 1. *Let A be a probabilistic weighted automata with alphabet $\Sigma = \{a, b\}$. Consider the Markov chain arising of A on input b^ω (we refer to this as the b-Markov chain) and we use similar notation for the a-Markov chain. The following assertions hold:*

1. *If for all closed recurrent sets C in the b-Markov chain, the (expected) limit-average value (in probabilistic sense) is at least 1, then there exists j such that for all closed recurrent sets arising of A on input $(b^j \cdot a)^\omega$ the expected limit-average reward is positive.*
2. *If for all closed recurrent sets C in the b-Markov chain, the (expected) limit-average value (in probabilistic sense) is at most 0, then there exists j such that for all closed recurrent sets arising of A on input $(b^j \cdot a)^\omega$ the expected limit-average reward is strictly less than 1.*

3. *If for all closed recurrent sets C in the b-Markov chain, the (expected) limit-average value (in probabilistic sense) is at most 0, and if for all closed recurrent sets C in the a-Markov chain, the (expected) limit-average value (in probabilistic sense) is at most 0, then there exists j such that for all closed recurrent sets arising of A on input $(b^j \cdot a^j)^\omega$ the expected limit-average reward is strictly less than 1/2.*

Proof. We present the proof in three parts.

1. Let β be the maximum absolute value of the weights of A. From any state $s \in A$, there is a path of length at most n to a closed recurrent set C in the b-Markov chain, where n is the number of states of A. Hence if we choose $j > n$, then any closed recurrent set in the Markov chain arising on the input $(b^j \cdot a)^\omega$ contains closed recurrent sets of the b-Markov chain. For $\epsilon > 0$, there exists k_ϵ such that from any state $s \in A$, for all $k > k_\epsilon$, on input b^k from s, the closed recurrent sets of the b-Markov chain is reached with probability at least $1 - \epsilon$ (by property 1 for Markov chains). If all closed recurrent sets in the b-Markov chain have expected limit-average value at least 1, then (by property 2 for Markov chains) for all $\epsilon > 0$, there exists l_ϵ such that for all $l > l_\epsilon$, from all states s of a closed recurrent set on the input b^l the expected average of the weights is at least $1 - \epsilon$, (i.e., expected sum of the weights is $l - l \cdot \epsilon$). Consider $0 < \epsilon \leq \min\{1/4, 1/(20 \cdot \beta)\}$, we choose $j = k + l$, where $k = k_\epsilon > 0$ and $l > \max\{l_\epsilon, k\}$. Observe that by our choice $j + 1 \leq 2l$. Consider a closed recurrent set in the Markov chain on $(b^j \cdot a)^\omega$ and we obtain a lower bound on the expected average reward as follows: with probability $1 - \epsilon$ the closed recurrent set of the b-Markov chain is reached within k steps, and then in the next l steps at the expected sum of the weights is at least $l - l \cdot \epsilon$, and since the worst case weight is $-\beta$ we obtain the following bound on the expected sum of the rewards

$$(1 - \epsilon) \cdot (l - l \cdot \epsilon) - \epsilon \cdot \beta \cdot (j + 1) \geq \frac{l}{2} - \frac{l}{10} = \frac{2l}{5}$$

   Hence the expected average reward is at least $1/5$ and hence positive.
2. The proof is similar to the previous result.
3. The proof is also similar to the first result. The only difference is that we use a long enough sequence of b^j such that with high probability a closed recurrent set in the b-Markov chain is reached and then stay long enough in the closed recurrent set to approach the expected sum of rewards to 0, and then present a long enough sequence of a^j such that with high probability a closed recurrent set in the a-Markov chain is reached and then stay long enough in the closed recurrent set to approach the expected sum of rewards to 0. The calculation is similar to the first part of the proof.

Thus we obtain the desired result. ■

We consider the alphabet Σ consisting of letters a and b, i.e., $\Sigma = \{a, b\}$. We define the language L_F of finitely many a's, i.e., for an infinite word w if w consists of infinitely many a's, then $L_F(w) = 0$, otherwise $L_F(w) = 1$. We also consider the language L_I of words with infinitely many a's (it is the complement of L_F).

Lemma 2. *Consider the language L_F of finitely many a's. The following assertions hold.*

1. *The language can be expressed as a* NLIMAVG.

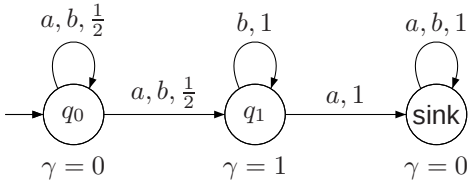

Fig. 2. A POSLIMAVG for Lemma 2.

2. *The language can be expressed as a* POSLIMAVG.
3. *The language cannot be expressed as* ASLIMAVG.

Proof. We present the three parts of the proof.

1. The result follows from the results of [7, Theorem 12] where the explicit construction of a NLIMAVG to express L_F is presented.
2. A POSLIMAVG automaton A to express L_F is as follows (see Fig. 2):
   (a) *States and weight function.* The set of states of the automaton is $\{q_0, q_1, sink\}$, with q_0 as the starting state. The weight function γ is as follows: $\gamma(q_0) = \gamma(sink) = 0$ and $\gamma(q_1) = 1$.
   (b) *Transition function.* The probabilistic transition function is as follows:
      (i) from q_0, given a or b, the next states are q_0, q_1, each with probability 1/2;
      (ii) from q_1 given b the next state is q_1 with probability 1, and from q_1 given a the next state is $sink$ with probability 1; and
      (iii) from $sink$ state the next state is $sink$ with probability 1 on both a and b. (it is an absorbing state).

   Given the automaton A consider any word w with infinitely many a's then, the automata reaches sink state in finite time with probability 1, and hence $A(w) = 0$. For a word w with finitely many a's, let k be the last position that an a appears. Then with probability $1/2^k$, after k steps, the automaton only visits the state q_1 and hence $A(w) = 1$. Hence there is a POSLIMAVG for L_F.
3. We show that L_F cannot be expressed as an ASLIMAVG. Consider an ASLIMAVG automaton A. Consider the Markov chain that arises from A if the input is only b (i.e., on b^ω), we refer to it as the b-Markov chain. If there is a closed recurrent set C that can be reached from the starting state (reached by any sequence of a and b's), then the limit-average reward (in probabilistic sense) in C must be at least 1 (otherwise, if there is a closed recurrent set C with limit-average reward less than 1, we can construct a finite word w that with positive probability will reach C, and then follow w by b^ω and we will have $A(w \cdot b^\omega) < 1$). Hence any closed recurrent set on the b-Markov chain has limit-average reward at least 1 and by Lemma 1 there exists j such that the $A((b^j \cdot a)^\omega) > 0$. Hence it follows that A cannot express L_F.

Hence the result follows. ■

Lemma 3. *Consider the language L_I of infinitely many a's. The following assertions hold.*

1. *The language cannot be expressed as an* NLIMAVG.

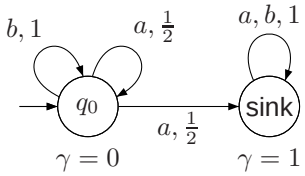

Fig. 3. An ASLIMAVG for Lemma 3.

2. *The language cannot be expressed as a* POSLIMAVG.
3. *The language can be expressed as* ASLIMAVG.

Proof. We present the three parts of the proof.

1. It was shown in the proof of [7, Theorem 13] that NLIMAVG cannot express L_I.
2. We show that L_I is not expressible by a POSLIMAVG. Consider a POSLIMAVG A and consider the b-Markov chain arising from A under the input b^ω. All closed recurrent sets C reachable from the starting state must have the limit-average value at most 0 (otherwise we can construct an word w with finitely many a's such that $A(w) > 0$). Since all closed recurrent set in the b-Markov chain has limit-average reward that is 0, using Lemma 1 we can construct a word $w = (b^j \cdot a)^\omega$, for a large enough j, such that $A(w) < 1$. Hence the result follows.
3. We now show that L_I is expressible as an ASLIMAVG. The automaton A is as follows (see Fig. 3):
   (a) *States and weight function.* The set of states are $\{q_0, sink\}$ with q_0 as the starting state. The weight function is as follows: $\gamma(q_0) = 0$ and $\gamma(sink) = 1$.
   (b) *Transition function.* The probabilistic transition function is as follows:
      (i) from q_0 given b the next state is q_0 with probability 1;
      (ii) at q_0 given a the next states are q_0 and $sink$ each with probability 1/2; (iii) the $sink$ state is an absorbing state.

   Consider a word w with infinitely many a's, then the probability of reaching the sink state is 1, and hence $A(w) = 1$. Consider a word w with finitely many a's, and let k be the number of a's, and then with probability $1/2^k$ the automaton always stay in q_0, and hence $A(w) = 0$.

Hence the result follows. ■

Lemma 4. *There exists a language that can be expressed by* POSLIMAVG, POSLIMSUP *and* POSLIMINF, *but not by* NLIMAVG, NLIMSUP *or* NLIMINF.

Proof. Consider an automaton A as follows (see Fig. 4):

1. *States and weight function.* The set of states are $\{q_0, q_1, sink\}$ with q_0 as the starting state. The weight function is as follows: $\gamma(q_0) = \gamma(q_1) = 1$ and $\gamma(sink) = 0$.
2. *Transition function.* The probabilistic transition is as follows:

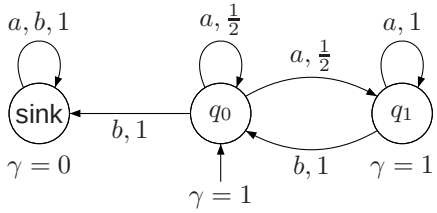

Fig. 4. A probabilistic weighted automaton (POSLIMAVG, POSLIMSUP, or POSLIMINF) for Lemma 4.

(i) from q_0 if the input letter is a, then the next states are q_0 and q_1 with probability 1/2;
(ii) from q_0 if the input letter is b, then the next state is $sink$ with probability 1;
(iii) from q_1, if the input letter is b, then the next state is q_0 with probability 1;
(iv) from q_1, if the input letter is a, then the next state is q_1 with probability 1; and
(v) the state $sink$ is an absorbing state.

If we consider the automaton A, and interpret it as a POSLIMAVG, POSLIMSUP, or POSLIMINF, then it accepts the following language:

$$L_z = \{a^{k_1}ba^{k_2}ba^{k_3}b\ldots \mid k_1, k_2, \cdots \in \mathbb{N}_{\geq 1} \cdot \prod_{i=1}^{\infty}(1 - \frac{1}{2^{k_i}}) > 0\};$$

i.e., $A(w) = 1$ if $w \in L_z$ and $A(w) = 0$ if $w \notin L_z$: the above claim follows from the argument following Lemma 5 of [2]. We now show that L_z cannot be expressed as NLIMAVG, NLIMSUP or NLIMINF. Consider a non-deterministic automaton A. Suppose there is a cycle C in A such that average of the rewards in C is positive. If no such cycle exists, then clearly A cannot express L_z as there exists word for which $L_z(w) = 1$. Consider a cycle C such that average of the rewards is positive, and let the cycle be formed by a finite word $w_C = a_0a_1\ldots a_n$ and there must exist at least one index $0 \leq i \leq n$ such that $a_i = b$. Hence the word can be expressed as $w_C = a^{j_1}ba^{j_2}b\ldots a^{j_k}b$, and hence there exists a finite word w_R (that reaches the cycle) such that $A(w_R \cdot w_C^\omega) > 0$. This contradicts that A is an automaton to express L_z as $L_z(w_R \cdot w_C^\omega) = 0$. Simply exchanging the average reward of the cycle by the maximum reward (resp. minimum reward) shows that L_z is not expressible by a NLIMSUP (resp. NLIMINF). ■

The next theorem summarizes the results for limit-average automata obtained in this section.

Theorem 2. ASLIMAVG *is incomparable in expressive power with* POSLIMAVG *and* NLIMAVG, *and* NLIMAVG *cannot express all languages expressible by* POSLIMAVG.

Open question. Whether NLIMAVG is reducible to POSLIMAVG or NLIMAVG is incomparable to POSLIMAVG (i.e., there is a language expressible by NLIMAVG but not by a POSLIMAVG) remains open.

3.3 Probabilistic LimInf-automata

Lemma 5. NLIMINF *is reducible to both* ASLIMINF *and* POSLIMINF.

Proof. It was shown in [7] that NLIMINF is reducible to DLIMINF. Since DLIMINF are special cases of ASLIMINF and POSLIMINF the result follows. ■

Lemma 6. *The language L_I is expressible by an* ASLIMINF*, but cannot be expressed as a* NLIMINF *or a* POSLIMINF.

Proof. It was shown in [7] that the language L_I is not expressible by NLIMINF. If we consider the automaton A of Lemma 3 and interpret it as an ASLIMINF, then the automaton A expresses the language L_I. The proof of the fact that POSLIMINF cannot express L_I is similar to the the proof of Lemma 3 (part(2)) and instead of the average reward of the closed recurrent set C, we need to consider the minimum reward of the closed recurrent set C. ■

Lemma 7. POSLIMINF *is reducible to* ASLIMINF.

Proof. Let A be a POSLIMINF and we construct a ASLIMINF B such that B is equivalent to A. Let V be the set of weights that appear in A and let v_1 be the least value in V. For each weight $v \in V$, consider the POSCW A^v that is obtained from A by considering all states with weight at least v as accepting states. It follows from the results of [1] that POSCW is reducible to ASCW (it was shown in [1] that ASBW is reducible to POSBW and it follows easily that dually POSCW is reducible to ASCW). Let D^v be an ASCW that is equivalent to A^v. We construct a POSLIMINF B^v from D^v by assigning weights v to the accepting states of D^v and the minimum weight v_1 to all other states. Consider a word w, and we consider the following cases.

1. If $A(w) = v$, then for all $v' \in V$ such that $v' \leq v$ we have $D^{v'}(w) = 1$, (i.e., the POSCW $A^{v'}$ and the ASCW $D^{v'}$ accepts w).
2. For $v \in V$, if $D^v(w) = 1$, then $A(w) \geq v$

It follows from above that $A = \max_{v \in V} B^v$. We will show later that ASLIMINF is closed under $\max$ (Lemma 18) and hence we can construct an ASLIMINF B such that $B = \max_{v \in V} B^v$. Thus the result follows. ■

Theorem 3. *We have the following strict inclusion*

$$\text{NLIMINF} \subsetneq \text{POSLIMINF} \subsetneq \text{ASLIMINF}$$

Proof. The fact that NLIMINF is reducible to POSLIMINF follows from Lemma 5, and the fact the POSLIMINF is not reducible to NLIMINF follows from Lemma 4. The fact that POSLIMINF is reducible to ASLIMINF follows from Lemma 7 and the fact that ASLIMINF is not reducible to POSLIMINF follows from Lemma 6. ■

3.4 Probabilistic LimSup-automata

Lemma 8. NLIMSUP *and* POSLIMSUP *are not reducible to* ASLIMSUP.

Proof. The language L_F of finitely many a's can be expressed as a non-deterministic Büchi automata, and hence as a NLIMSUP. We will show that NLIMSUP is reducible to POSLIMSUP. It follows that L_F is expressible as NLIMSUP and POSLIMSUP. The proof of the fact that ASLIMSUP cannot express L_F is similar to the the proof of Lemma 2 (part(3)) and instead of the average reward of the closed recurrent set C, we need to consider the maximum reward of the closed recurrent set C. ■

Deterministic in limit NLIMSUP**.** Consider an automaton A that is a NLIMSUP. Let $v_1 < v_2 < \ldots < v_k$ be the weights that appear in A. We call the automaton A *deterministic in the limit* if for all states s with weight greater than v_1, all states t reachable from s are deterministic.

Lemma 9. *For every* NLIMSUP A*, there exists a* NLIMSUP B *that is deterministic in the limit and equivalent to* B.

Proof. From the results of [8] it follows that a NBW A can be reduced to an equivalent NBW B such that B is deterministic in the limit. Let A be a NLIMSUP, and let V be the set of weights that appear in A. and let $V = \{v_1, \ldots, v_k\}$ with $v_1 < v_2 < \cdots < v_k$. For each $v \in V$, consider the NBW A_v whose (boolean) language is the set of words w such that $L_A(w) \geq v$, by declaring to be accepting the states with weight at least v. Let B_v be the deterministic in the limit NBW that is equivalent to A_v. The automaton B that is deterministic in the limit and is equivalent to A is obtained as the automaton that by initial non-determinism chooses between the B_v's, for $v \in V$. ■

Lemma 10. NLIMSUP *is reducible to* POSLIMSUP.

Proof. Given a NLIMSUP A, consider the NLIMSUP B that is deterministic in the limit and equivalent to B. By assigning equal probabilities to all out-going transitions from a state we obtain a POSLIMSUP C that is equivalent to B (and hence A). The result follows. ■

Lemma 11. ASLIMSUP *is reducible to* POSLIMSUP.

Proof. Consider a ASLIMSUP A and let the weights of A be $v_1 < v_2 \ldots < v_l$. For $1 \leq i \leq l$ consider the ASBW obtained from A with the set of state with reward at least v_i as the Büchi states. It follows from the results of [1] that ASBW is reducible to POSBW. Let B_i be the POSBW that is equivalent to A_i. Let C_i be the automaton such that all Büchi states of B_i is assigned weight v_i and all other states are assigned v_1. Consider the automata C that goes with equal probability to the starting states of C_i, for $1 \leq i \leq l$, and we interpret C as a POSLIMSUP. Consider a word w, and let $A(w) = v_j$ for some $1 \leq j \leq l$, i.e., given w, the set of states with reward at least v_j is visited infinitely often with probability 1 in A. Hence the POSBW B_i accepts w with positive probability, and since C chooses C_i with positive probability, it follows that given w, in C the weight v_j is visited infinitely often with positive probability, i.e., $C(w) \geq v_j$. Moreover, given w, for all $v_k > v_l$,

the set of states with weight at least v_k is visited infinitely often with probability 0 in A. Hence for all $k > j$, the automata B_k accepts w with probability 0. Thus $C(w) < v_k$ for all $v_k > v_j$. Hence $C(w) = A(w)$ and thus AsLimSup is reducible to PosLimSup. ■

Lemma 12. AsLimSup *is not reducible to* NLimSup.

Proof. It follows from [1] that for $0 < \lambda < 1$ the following language L_λ can be expressed by a AsBW and hence by AsLimSup:

$$L_\lambda = \{a^{k_1}ba^{k_2}ba^{k_3}b\ldots \mid k_1, k_2, \cdots \in \mathbb{N}_{\geq 1}. \prod_{i=1}^{\infty}(1-\lambda^{k_i}) > 0\}.$$

It follows from argument similar to Lemma 4 that there exists $0 < \lambda < 1$ such that L_λ cannot be expressed by a NLimSup. Hence the result follows. ■

Theorem 4. AsLimSup *and* NLimSup *are incomparable in expressive power, and* PosLimSup *is more expressive than* AsLimSup *and* NLimSup.

Lemma 13. PosCW *is reducible to* PosBW.

Proof. Let $A = \langle Q, q_I, \Sigma, \delta, C\rangle$ be a PosCW with the set $C \subseteq Q$ of accepting states. We construct a PosBW $\overline{A}$ as follows:

1. The set of states is $Q \cup \overline{Q}$ where $\overline{Q} = \{\overline{q} \mid q \in Q\}$ is a copy of the states in Q;
2. q_I is the initial state;
3. The transition function is as follows, for all $\sigma \in \Sigma$:
   (a) for all states $q, q' \in Q$, we have $\overline{\delta}(q, \sigma, q') = \overline{\delta}(q, \sigma, \overline{q'}) = \frac{1}{2} \cdot \delta(q, \sigma, q')$, i.e., the state q' and its copy $\overline{q'}$ are reached with half of the original transition probability;
   (b) the states $\overline{q} \in \overline{Q}$ such that $q \notin C$ are absorbing states (i.e., $\overline{\delta}(\overline{q}, \sigma, \overline{q}) = 1$);
   (c) for all states $q \in C$ and $q' \in Q$, we have $\overline{\delta}(\overline{q}, \sigma, \overline{q'}) = \delta(q, \sigma, q')$, i.e., the transition function in the copy automaton follows that of A for states that are copy of the accepting states.
4. The set of accepting states is $\overline{C} = \{\overline{q} \in \overline{Q} \mid q \in C\}$.

We now show that the language of the PosCW A and the language of PosBW $\overline{A}$ coincides. Consider a word w such that $A(w) = 1$. Let α be the probability that given the word w eventually always states in C are visited in A, and since $A(w) = 1$ we have $\alpha > 0$. In other words, as limit k tends to ∞, the probability that after k steps only states in C are visited is α. Hence there exists k_0 such that the probability that after k_0 steps only states in C are visited is at least $\frac{\alpha}{2}$. In the automaton $\overline{A}$, the probability to reach states of $\overline{Q}$ after k_0 steps has probability $p = 1 - \frac{1}{2^{k_0}} > 0$. Hence with positive probability (at least $p \cdot \frac{\alpha}{2}$) the automaton visits infinitely often the states of $\overline{C}$, and hence $\overline{A}(w) = 1$. Observe that since every state in $\overline{Q} \setminus \overline{C}$ is absorbing and non-accepting), it follows that if we consider an accepting run $\overline{A}$, then the run must eventually always visits states in $\overline{C}$ (i.e., the copy of the accepting states C). Hence it follows that for a given word w, if $\overline{A}(w) = 1$, then with positive probability eventually always states in C are visited in A. Thus $A(w) = 1$, and the result follows. ■

Lemma 14. POSLIMINF *is reducible to* POSLIMSUP*, and* ASLIMSUP *is reducible to* ASLIMINF.

Proof. We present the proof that POSLIMINF is reducible to POSLIMSUP, the other proof being similar. Let A be a POSLIMINF, and let V be the set of weights that appear in A. For each $v \in V$, it is easy to construct a POSCW A_v whose (boolean) language is the set of words w such that $L_A(w) \geq v$, by declaring to be accepting the states with weight at least v. We then construct for each $v \in V$ a POSBW $\overline{A}_v$ that accepts the language of A_v (such a POSBW can be constructed by Lemma 13). Finally, assuming that $V = \{v_1, \ldots, v_n\}$ with $v_1 < v_2 < \cdots < v_n$, we construct the POSLIMSUP B_i for $i = 1, 2, \ldots, n$ where B_i is obtained from $\overline{A}_{v_i}$ by assigning weight v_i to each accepting states, and v_1 to all the other states. The POSLIMSUP that expresses the language of A is $\max_{i=1,2\ldots,n} B_i$ and since POSLIMSUP is closed under $\max$ (see Lemma 16), the result follows. ■

Lemma 15. ASLIMINF *and* POSLIMSUP *are reducible to each other;* ASLIMSUP *and* POSLIMINF *have incomparable expressive power.*

Proof. This result is an easy consequence of the fact that an automaton interpreted as ASLIMINF defines the complement of the language of the same automaton interpreted as POSLIMSUP (and similarly for ASLIMSUP and POSLIMINF), and from the fact that ASLIMINF and POSLIMSUP are closed under complement, while ASLIMSUP and POSLIMINF are not (see Lemma 21 and 22). ■

3.5 Probabilistic Disc-automata

For probabilistic discounted-sum automata, the following result establishes equivalence of the non-deterministic and the positive semantics, and the equivalence of the universal and the almost-sure semantics.

Theorem 5. *The following assertions hold: (a)* NDISC *and* POSDISC *are reducible to each other; (b)* UDISC *and* ASDISC *are reducible to each other.*

Proof. (a) We first prove that NDISC is reducible to POSDISC. Let $A = \langle Q, \rho_I, \Sigma, \delta_A, \gamma \rangle$ be a NDISC, and let $v_{\min}, v_{\max}$ be its minimal and maximal weights respectively. Consider the POSDISC $B = \langle Q, \rho_I, \Sigma, \delta_B, \gamma \rangle$ where $\delta_B(q, \sigma)$ is the uniform distribution over the set of states q' such that $(q, \sigma, \{q'\}) \in \delta_A$. Let $r = q_0\sigma_1 q_1 \sigma_2 \ldots$ be a run of A (over $w = \sigma_1\sigma_2\ldots$) with value η. For all $\epsilon > 0$, we show that $\mathbb{P}^B(\{r \in \mathsf{Run}^B(w) \mid \mathsf{Val}(\gamma(r)) \geq \eta - \epsilon\}) > 0\}$. Let $n \in \mathbb{N}$ such that $\frac{\lambda^n}{1-\lambda} \cdot (v_{\max} - v_{\min}) \leq \epsilon$, and let $r_n = q_0\sigma_1 q_1 \sigma_2 \ldots \sigma_n q_n$. The discounted sum of the weights in r_n is at least $\eta - \frac{\lambda^n}{1-\lambda} \cdot (v_{\max})$. The probability of the set of runs over w that are continuations of r_n is positive, and the value of all these runs is at least $\eta - \frac{\lambda^n}{1-\lambda} \cdot (v_{\max} - v_{\min})$, and therefore at least $\eta - \epsilon$. This shows that $L_B(w) \geq \eta$, and thus $L_B(w) \geq L_A(w)$. Note that $L_B(w) \leq L_A(w)$ since there is no run in A (nor in B) over w with value greater than $L_A(w)$. Hence $L_B = L_A$.

Now, we prove that POSDISC is reducible to NDISC. Given a POSDISC $B = \langle Q, \rho_I, \Sigma, \delta_B, \gamma \rangle$, we construct a NDISC $A = \langle Q, \rho_I, \Sigma, \delta_A, \gamma \rangle$ where $(q, \sigma, \{q'\}) \in \delta_A$ if and only if $\delta_B(q, \sigma)(q') >$

		max	min	comp.	sum	emptiness	universality
>0	POSSUP	√	√	×	√	√	√
	POSLIMSUP	√	√	√	√	×	×
	POSLIMINF	√	√	×	√	?	√
	POSLIMAVG	√	×	×	?	?	?
	POSDISC	√	×	×	√	√	? (1)
almost-sure	ASSUP	√	√	×	√	√	√
	ASLIMSUP	√	√	×	√	√	?
	ASLIMINF	√	√	√	√	×	×
	ASLIMAVG	×	√	×	×	?	?
	ASDISC	×	√	×	√	? (1)	√

The universality problem for NDISC can be reduced to (1). It is not known whether this problem is decidable.

Table 1. Closure properties and decidability of the emptiness and universality problems.

0, for all $q, q' \in Q$, $\sigma \in \Sigma$. By analogous arguments as in the first part of the proof, it is easy to see that $L_B = L_A$.

(b) It is easy to see that the complement of the quantitative language defined by a UDISC (resp. ASDISC) can be defined by a NDISC (resp. POSDISC). Then, the result follows from Part $a)$ (essentially, given a UDISC, we obtain easily an NDISC for the complement, then an equivalent POSDISC, and finally a ASDISC for the complement of the complement, i.e., the original quantitative language). ■

Note that a by-product of this proof is that the language of a POSDISC does not depend on the precise values of the probabilities, but only on whether they are positive or not.

4 Closure Properties of Probabilistic Weighted Automata

We consider the closure properties of the probabilistic weighted automata under the operations $\max$, $\min$, complement, and sum. The results are presented in Table 1.

4.1 Closure under $\max$ and $\min$

Lemma 16 (Closure by initial non-determinism). POSLIMSUP, POSLIMINF *and* POSLIMAVG *is closed under* $\max$*; and* ASLIMSUP, ASLIMINF *and* ASLIMAVG *is closed under* $\min$.

Proof. Given two automata A_1 and A_2 consider the automata A obtained by initial non-deterministic choice of A_1 and A_2. Formally, let q_1 and q_2 be the initial states of A_1 and A_2, respectively, then in A we add an initial state q_0 and the transition from q_0 is as follows: for $\sigma \in \Sigma$, consider the set $Q_\sigma = \{q \in Q_1 \cup Q_2 \mid \delta_1(q_1, \sigma)(q) > 0 \text{ or } \delta_2(q_2, \sigma)(q) > 0\}$. From q_0, for input letter σ, the successors are from Q_σ each with probability $1/|Q_\sigma|$. If A_1 and A_2 are POSLIMSUP (resp. POSLIMINF, POSLIMAVG), then A is a POSLIMSUP (resp. POSLIMINF, POSLIMAVG) such that $A = \max\{A_1, A_2\}$. Similarly, if A_1 and A_2 are ASLIMSUP (resp. ASLIMINF, ASLIMAVG), then A is a ASLIMSUP (resp. ASLIMINF, ASLIMAVG) such that $A = \min\{A_1, A_2\}$. ■

Lemma 17 (Closure by synchronized product). ASLIMSUP *is closed under* $\max$ *and* POSLIMINF *is closed under* $\min$.

Proof. We present the proof that ASLIMSUP is closed under $\max$. Let A_1 and A_2 be two probabilistic weighted automata with weight function γ_1 and γ_2, respectively. Let A be the usual synchronized product of A_1 and A_2 with weight function γ such that $\gamma((s_1, s_2)) = \max\{\gamma_1(s_1), \gamma_2(s_2)\}$. Given a path $\pi = ((s_0^1, s_0^2), (s_1^1, s_1^2), \ldots)$ in A we denote by $\pi \restriction 1$ the path in A_1 that is the projection of the first component of π and we use similar notation for $\pi \restriction 2$. Consider a word w, let $\max\{A_1(w), A_2(w)\} = v$. We consider the following two cases to show that $A(w) = v$.

1. W.l.o.g. let the maximum be achieved by A_1, i.e., $A_1(w) = v$. Let B_i^v be the set of states s_i in A_i such that weight of s_i is at least v. Since $A_1(w) = v$, given the word w, in A_1 the event Büchi(B_1^v) holds with probability 1. Consider the following set of paths in A

$$\Pi^v = \{\pi \mid (\pi \restriction 1) \in \text{Büchi}(B_1^v)\}.$$

   Since given w, the event Büchi(B_1^v) holds with probability 1 in A_1, it follows that given w, the event Π^v holds with probability 1 in A. The γ function ensures that every path $\pi \in \Pi^v$ visits weights of value at least v infinitely often. Hence $A(w) \geq v$.
2. Consider a weight value $v' > v$. Let C_i^v be the set of states s_i in A_i such that the weight of s_i is less than v'. Given the word w, since $A_i(w) < v'$, it follows that probability of the event coBüchi(C_i^v) in A_i, given the word w, is positive. Hence given the word w, the probability of the event coBüchi$(C_1^v \times C_2^v))$ is positive in A. It follows that $A(w) < v'$.

The result follows. If A_1 and A_2 are POSLIMINF, and in A we assign weights such that every state in A has the minimum weight of its component states, and we consider A as a POSLIMINF, then $A = \min\{A_1, A_2\}$. The proof is similar to the result for ASLIMSUP. ■

Lemma 18. POSLIMSUP *is closed under* $\min$ *and* ASLIMINF *is closed under* $\max$.

Proof. Let A_1 and A_2 be two POSLIMSUP. We construct a POSLIMSUP A such that $A = \min\{A_1, A_2\}$. Let V_i be the set of weights that appear in A_i (for $i = 1, 2$), and let $V = V_1 \cup V_2$ and let v_1 be the least value in V. For each weight $v \in V_1 \cup V_2 = \{v_1, \ldots, v_k\}$, consider the POSBW A_i^v that is obtained from A_i by considering all states with weight at least v as accepting states. Since POSBW is closed under intersection(by the results of [2]), we can construct a POSBW A_{12}^v that is the intersection of A_1^v and A_2^v, i.e. $A_{12}^v = A_1^v \cap A_2^v$. We construct a POSLIMSUP B_{12}^v from A_{12}^v by assigning weights v to the accepting states of A_{12}^v and the minimum weight v_1 to all other states. Consider a word w, and we consider the following cases.

1. If $\min\{A_1(w), A_2(w)\} = v$, then for all $v' \in V$ such that $v' \leq v$ we have $A_{12}^{v'}(w) = 1$, (i.e., the POSBW $A_{12}^{v'}$ accepts w).
2. If $A_{12}^v(w) = 1$, then $A_1(w) \geq v$ and $A_2(w) \geq v$, i.e., $\min\{A_1(w), A_2(w)\} \geq v$.

It follows from above that $\min\{A_1, A_2\} = \max_{v \in V} B_{12}^v$. Since POSLIMSUP is closed under $\max$ (by initial non-determinism), it follows that POSLIMSUP is closed under $\min$. The proof of closure of ASLIMINF under $\max$ is similar. ■

The closure properties of LimAvg-automata in the positive semantics rely on the following lemma.

Lemma 19. *Consider the alphabet $\Sigma = \{a, b\}$, and consider the languages L_a and L_b that assigns the long-run average number of a's and b's, respectively. Then the following assertions hold.*

1. *There is no* POSLIMAVG *for the language $L_m = \min\{L_a, L_b\}$.*
2. *There is no* POSLIMAVG *for the language $L^* = 1 - \max\{L_a, L_b\}$.*

Proof. To obtain a contradiction, assume that there exists a POSLIMAVG A (for either L_m or L^*). We first claim that if we consider the a-Markov or the b-Markov chain of A, then there must be either an a-closed recurrent set or a b-closed recurrent set C that is reachable in A such that the expected sum of the weights in C is positive. Otherwise, if for all a-closed recurrent sets and b-closed recurrent sets we have that the expected sum of the weights is zero or negative, then we fool the automaton as follows. By Lemma 1, it follows that there exists a j such that $A((a^j \cdot b^j)^\omega) < 1/2$, however, $L_m(w) = L^*(w) = \frac{1}{2}$, i.e., we have a contradiction. W.l.o.g., we assume that there is an a-closed recurrent set C such that expected sum of weights of C is positive. Then we present the following word w: a finite word w_C to reach the cycle C, followed by a^ω; the answer of the automaton is positive, *i.e.*, $L_A(w) > 0$, while $L_m(w) = L^*(w) = 0$. Hence the result follows. ■

Lemma 20. POSLIMAVG *is not closed under* $\min$ *and* ASLIMAVG *is not closed under* $\max$.

Proof. The result for POSLIMAVG follows from Lemma 19. We now show that ASLIMAVG is not closed under $\max$. Consider the alphabet $\Sigma = \{a, b\}$ and the quantitative languages L_a and L_b that assign the value of long-run average number of a's and b's, respectively. There exists DLIMAVG (and hence ASLIMAVG) for L_a and L_b. We show that $L_m = \max(L_a, L_b)$ cannot be expressed by an ASLIMAVG. By contradiction, assume that A is an ASLIMAVG with set of states Q that defines L_m. Consider any a-closed recurrent C in A. The expected limit-average of the weights of the recurrent set must be 1, as if we consider the word $w^* = w_C \cdot a^\omega$ where w_C is a finite word to reach C, the value of w^* in L_m is 1. Hence, the limit-average of the weights of all the reachable a-closed recurrent set C in A is 1.

Given $\epsilon > 0$, there exists j_ϵ such that the following properties hold:

1. from any state of A, given the word a^{j_ϵ} with probability $1-\epsilon$ an a-closed recurrent set is reached (by property 1 for Markov chains);
2. once an a-closed recurrent set is reached, given the word a^{j_ϵ}, (as a consequence of property 2 for Markov chains) we can show that the following properties hold: (a) the expected average of the weights is at least $j_\epsilon \cdot (1-\epsilon)$, and (b) the probability distribution of the states is with ϵ of the probability distribution of the states for the word $a^{2 \cdot j_\epsilon}$ (this holds as the probability distribution of states on words a^j converges to the probability distribution of states on the word a^ω).

Let $\beta > 1$ be a number that is greater than the absolute maximum value of weights in A. We chose $\epsilon > 0$ such that $\epsilon < \frac{1}{40 \cdot \beta}$. Let $j = 2 \cdot j_\epsilon$ (such that j_ϵ satisfies the properties above). Consider the word $(a^j \cdot b^{3j})^\omega$ and the answer by A must be $\frac{3}{4}$, as $L_m((a^j \cdot b^{3j})^\omega) = \frac{3}{4}$. Consider the word $\widehat{w} = (a^{2j} \cdot b^{3j})^\omega$ and consider a closed recurrent set in the Markov chain obtain from A on $\widehat{w}$. We obtain the following lower bound on the expected limit-average of the weights: (a) with probability at least $1 - \epsilon$, after $j/2$ steps, a-closed recurrent sets are reached; (b) the expected average of the weights for the segment between a^j and a^{2j} is at least $j \cdot (1-\epsilon)$; and (c) the difference in probability

distribution of the states after a^j and a^{2j} is at most ϵ. Since the limit-average of the weights of $(a^j \cdot b^{3j})^\omega$ is $\frac{3}{4}$, the lower bound on the limit-average of the weights is as follows

$$\begin{aligned}(1-3\cdot\epsilon)\cdot(\tfrac{3\cdot j+j\cdot(1-\epsilon)}{5j}) - 3\cdot\epsilon\cdot\beta &= (1-\epsilon)(\tfrac{4}{5}-\tfrac{\epsilon}{5}) - 3\cdot\epsilon\cdot\beta \\ &\geq \tfrac{4}{5} - \epsilon - 3\cdot\epsilon\cdot\beta \\ &\geq \tfrac{4}{5} - 4\cdot\epsilon\cdot\beta \\ &\geq \tfrac{4}{5} - \tfrac{1}{10} \\ &\geq \tfrac{7}{10} > \tfrac{3}{5}.\end{aligned}$$

It follows that $A((a^{2j}\cdot b^{3j})^\omega) > \frac{3}{5}$. This contradicts that A expresses L_m. ■

4.2 Closure under complement

Lemma 21. POSLIMSUP *and* ASLIMINF *are closed under complement.*

Proof. We first present the proof for POSLIMSUP. Let A be a POSLIMSUP, and let V be the set of weights that appear in A. For each $v \in V$, it is easy to construct a POSBW A_v whose (boolean) language is the set of words w such that $L_A(w) \geq v$, by declaring to be accepting the states with weight at least v. We then construct for each $v \in V$ a POSBW $\bar{A}_v$ (with accepting states) that accepts the (boolean) complement of the language accepted by A_v (such a POSBW can be constructed since POSBW is closed under complementation by the results of [1]). Finally, assuming that $V = \{v_1, \ldots, v_n\}$ with $v_1 < v_2 < \cdots < v_n$, we construct the POSLIMSUP B_i for $i = 2, \ldots, n$ where B_i is obtained from $\bar{A}_{v_i}$ by assigning weight $-v_{i-1}$ to each accepting states, and $-v_n$ to all the other states. The complement of L_A is then $\max\{L_{B_2}, \ldots, L_{B_n}\}$ which is accepted by a POSLIMSUP (since POSLIMSUP is closed under $\max$). The result for ASLIMINF is similar and it uses the closure of ASCW under complementation which can be easily proved from the closure under complementation of POSBW. ■

Lemma 22. ASLIMSUP *and* POSLIMINF *are not closed under complement.*

Proof. It follows from Lemma 8 that the language L_F of finitely many a's is not expressible by an ASLIMSUP, whereas the complement L_I of infinitely many a's is expressible as a DBW and hence as a ASLIMSUP. It follows from Lemma 6 that language L_I is not expressible as an POSLIMINF, whereas its complement L_F is expressible by a DCW and hence a POSLIMINF. ■

Lemma 23. POSLIMAVG *and* ASLIMAVG *are not closed under complement.*

Proof. The fact that POSLIMAVG is not closed under complement follows from Lemma 19. We now show that ASLIMAVG is not closed under complement. Consider the DLIMAVG A over alphabet $\Sigma = \{a, b\}$ that consists of a single self-loop state with weight 1 for a and 0 for b. Notice that $A(w.a^\omega) = 1$ and $A(w.b^\omega) = 0$ for all $w \in \Sigma^*$. To obtain a contradiction, assume that there exists a ASLIMAVG B such that $B = 1 - A$. For all finite words $w \in \Sigma^*$, let $B(w)$ be the expected average

weight of the finite run of B over w. Fix $0 < \epsilon < \frac{1}{2}$. For all finite words w, there exists a number n_w such that the average number of a's in $w.b^{n_w}$ is at most ϵ, and there exists a number m_w such that $B(w.a^{m_w}) \leq \epsilon$ (since $B(w.a^\omega) = 0$). Hence, we can construct a word $w = b^{n_1}a^{m_1}b^{n_2}a^{m_2}\ldots$ such that $A(w) \leq \epsilon$ and $B(w) \leq \epsilon$. Since $B = 1 - A$, this implies that $1 \leq 2\epsilon$, a contradiction. ■

4.3 Closure under sum

Lemma 24. POSLIMSUP *and* ASLIMSUP *are closed under sum.*

Proof. Given two POSLIMSUP (resp. ASLIMSUP) A_1 and A_2, we construct a POSLIMSUP (resp. ASLIMSUP) A for the sum of their languages as follows. For a pair (v_1, v_2) of weights (v_i in A_i, for $i = 1, 2$), consider a copy of the synchronized product of A_1 and A_2. We attach a bit b whose range is $\{1, 2\}$ to each state to remember that we expect A_b to visit the guessed weight v_b. Whenever this occurs, the bit b is set to $3 - b$, and the weight of the state is $v_1 + v_2$. All other states (*i.e.* when b is unchanged) have weight $\min\{v_1 + v_2 \mid v_1 \in V_1 \wedge v_2 \in V_2\}$. Let the automata constructed be $A_{(v_1,v_2)}$. Then $A = \max_{(v_1,v_2)} A_{(v_1,v_2)}$. Since POSLIMSUP (resp. ASLIMSUP) is closed under $\max$ the result follows. ■

Lemma 25. POSLIMINF *and* ASLIMINF *are closed under sum.*

Proof. Given two POSLIMINF (resp. ASLIMINF) A_1 and A_2, we construct a POSLIMINF (resp. ASLIMINF) A for the sum of their languages as follows. For $i = 1, 2$, let V_i be the set of weights that appear in A_i. Let $v_{\min} = \min\{v_1 + v_2 \mid v_1 \in V_1 \wedge v_2 \in V_2\}$. For $v_1 \in V_1$ and $v_2 \in V_2$, for $i = 1, 2$, consider the POSCW (resp. ASCW) A_{v_i} obtained from A_i by making all states with weights at least v_i as accepting states. Let $A_{(v_1,v_2)}$ be the POSCW (resp. ASCW) such that $A_{(v_1,v_2)} = A_{v_1} \cap A_{v_2}$: such an POSCW (resp. ASCW) exists since POSCW (resp. ASCW) is closed under intersection. In other words, for a word w we have $A_{(v_1,v_2)}(w) = 1$ iff $A_1(w) \geq v_1$ and $A_2(w) \geq v_2$. Let $\overline{A}_{(v_1,v_2)}$ be the POSLIMINF (resp. ASLIMINF) obtained from $A_{(v_1,v_2)}$ by assigning weight $v_1 + v_2$ to all accepting states and weight $v_{\min}$ to all other states. Then the automaton for the sum of A_1 and A_2 (denoted as $A_1 + A_2$) is $\max_{(v_1,v_2)\in V_1\times V_2} \overline{A}_{(v_1,v_2)}$. Since POSLIMINF (resp. ASLIMINF) is closed under $\max$ the result follows. ■

Lemma 26. ASLIMAVG *is not closed under sum.*

Proof. Consider the alphabet $\Sigma = \{a, b\}$, and consider the DLIMAVG-definable languages L_a and L_b that assigns to each word w the long-run average number of a's and b's in w respectively. Let $L_+ = L_a + L_b$. We show that L_+ is not expressible by ASLIMAVG. Assume towards contradiction that L_+ is defined by an ASLIMAVG A with set of states Q (we assume w.l.o.g that every state in Q is reachable). Let $\beta > 1$ be greater than the maximum absolute value of the weights in A.

First, we claim that from every state $q \in Q$, if we consider the automaton A_q with q as starting state then $A_q(a^\omega) = 1$: this follows since if we consider a finite word w_q to reach q, then $L_+(w_q \cdot a^\omega) = 1$ and hence $A(w_q \cdot a^\omega) = 1$. It follows that from any state q, as k tends to ∞, the expected

average of the weights converges almost-surely to 1. This implies if we consider the a-Markov chain arising from A, then from any state q, for all closed recurrent set C of states reachable from q, the expected average of the weights of C is 1. Hence for every $\gamma > 0$ there exists a natural number k_0^γ such that from any state q, for all $k > k_0^\gamma$ given the word a^k the expected average of the weights is at least $\frac{1}{2}$ with probability $1 - \gamma$ (this is because we can chose long enough k such that the closed recurrent states are reached with probability $1-\gamma$ by property 1 for Markov chains, and then the long enough sequence ensures that the expected average approaches 1 by property 2 for Markov chains), and for the first k_0^γ steps the expected average of the weights is at least $-\beta$. The same result holds if we consider as input a sequence of b's instead of a's.

Consider the word w generated inductively by the following procedure: (a) w_0 is the empty word; (b) we generate w_{i+1} from w_i as follows: (i) the sequence of letters added to w_i to obtain w_{i+1} is at least i; (ii) first we generate a long enough sequence w'_{i+1} of a's after w_i such that the average number of b's in $w_i \cdot w'_{i+1}$ falls below $\frac{1}{i}$; (iii) then generate a long enough sequence w''_{i+1} of b's such that the average number of a's in $w_i \cdot w'_{i+1} \cdot w''_{i+1}$ falls below $\frac{1}{i}$; (iv) the word $w_{i+1} = w_i \cdot w'_{i+1} \cdot w''_{i+1}$. The word w is the limit of these sequences. For $\gamma > 0$, consider $i \geq 6 \cdot k_0^\gamma \cdot \beta$ (where k_0^γ satisfies the properties described above for γ). By construction for $i > 6 \cdot k_0^\gamma \cdot \beta$, the length of w_i is at least $6 \cdot k_0 \cdot \beta$, and hence it follows that in the segment constructed between w_i and w_{i+1}, for all $|w_i| \leq \ell \leq |w_{i+1}|$ with probability at least $1 - \gamma$ the expected average of the weights is at least

$$\frac{\frac{\ell - k_0^\gamma}{2} - k_0^\gamma \cdot \beta}{\ell} \geq \frac{1}{2} - \frac{2 \cdot k_0^\gamma \cdot \beta}{\ell} \geq \frac{1}{2} - \frac{1}{3} \geq \frac{1}{6}.$$

Hence for all $\gamma > 0$, the expected average of the weights is at least $\frac{1}{6}$ with probability at least $1 - \gamma$. Since this holds for all $\gamma > 0$, it follows that the expected average of the weights is at least $\frac{1}{6}$ almost-surely, (i.e., $A(w) \geq \frac{1}{6}$). We have $L_a(w) = L_b(w) = 0$ and thus $L_+(w) = 0$, while $A(w) \geq \frac{1}{6}$. Thus we have a contradiction. ■

Lemma 27. POSDISC *and* ASDISC *are closed under sum.*

Proof. The result for POSDISC follows from Theorem 5 and the fact that NDISC and UDISC are closed under sum (which is easy to prove using a synchronized product of automata where the weight of a joint transition is the sum of the weights of the corresponding transitions. ■

Open question. Whether POSLIMAVG is closed under sum remains open.

5 Decision Problems for Probabilistic Weighted Automata

We conclude the paper with some decidability and undecidability results for classical decision problems about quantitative languages (see Table 1). Most of them are direct corollaries of the results in [1]. Given a weighted automaton A and a rational number $\nu \in \mathbb{Q}$, the *quantitative emptiness problem* asks whether there exists a word $w \in \Sigma^\omega$ such that $L_A(w) \geq \nu$, and the *quantitative universality problem* asks whether $L_A(w) \geq \nu$ for all words $w \in \Sigma^\omega$.

Theorem 6. *The emptiness and universality problems for* POSSUP *and* ASSUP *are decidable.*

Proof. By Theorem 1, these problems reduce to emptiness of DSUP which is decidable ([7, Theorem 1]).

The following theorems are trivial corollaries of [1, Theorem 2].

Theorem 7. *The emptiness problem for* POSLIMSUP *and the universality problem for* ASLIMSUP *are undecidable.*

It is easy to obtain the following result as a straightforward generalization of [1, Theorem 6].

Theorem 8. *The emptiness problem for* ASLIMSUP *and the universality problem for* POSLIMINF *are decidable.*

The following result is a particular case of [1, Corollary 3].

Theorem 9. *The emptiness problem for* ASLIMINF *and the universality problem for* POSLIMSUP *are undecidable.*

Finally, by Theorem 5 and the decidability of emptiness for NDISC, we get the following result.

Theorem 10. *The emptiness problem for* POSDISC *and the universality problem for* ASDISC *are decidable.*

Note that by Theorem 5, the universality problem for NDISC (which is not know to be decidable) can be reduced to the universality problem for POSDISC and to the emptiness problem for ASDISC.

References

1. C. Baier, N. Bertrand, and M. Größer. On decision problems for probabilistic Büchi automata. In *Proc. of FoSSaCS*, LNCS 4962, pages 287–301. Springer, 2008.
2. C. Baier and M. Größer. Recognizing omega-regular languages with probabilistic automata. In *Proc. of LICS: Logic in Comp. Science*, pages 137–146. IEEE, 2005.
3. A. Chakrabarti, K. Chatterjee, T. A. Henzinger, O. Kupferman, and R. Majumdar. Verifying quantitative properties using bound functions. In *Proc. of CHARME*, LNCS 3725, pages 50–64. Springer, 2005.
4. A. Chakrabarti, L. de Alfaro, T. A. Henzinger, and M. Stoelinga. Resource interfaces. In *Proc. of EMSOFT*, LNCS 2855, pages 117–133. Springer, 2003.
5. K. Chatterjee, L. de Alfaro, M. Faella, T.A. Henzinger, R. Majumdar, and M. Stoelinga. Compositional quantitative reasoning. In *Proc. of QEST*, pages 179–188. IEEE Computer Society Press, 2006.
6. K. Chatterjee, L. Doyen, and T. A. Henzinger. Expressiveness and closure properties for quantitative languages, 2008. http://infoscience.epfl.ch/record/116423.
7. K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. In *Proc. of CSL: Computer Science Logic*, LNCS 5213, pages 385–400. Springer-Verlag, 2008.
8. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.
9. K. Culik II and J. Karhumäki. Finite automata computing real functions. *SIAM J. Comput.*, 23(4):789–814, 1994.
10. Z. Ésik and W. Kuich. An algebraic generalization of omega-regular languages. In *Proc. of MFCS*, LNCS 3153, pages 648–659. Springer, 2004.
11. E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games*, Lecture Notes in Computer Science 2500. Springer-Verlag, 2002.
12. J. G. Kemeny, J. L. Snell, and A. W. Knapp. *Denumerable Markov Chains*. D. Van Nostrand Company, 1966.
13. M. Y. Vardi. An automata-theoretic approach to linear temporal logic. In *Proc. of Banff Higher Order Workshop*, LNCS 1043, pages 238–266. Springer, 1995.