Probabilistic Weighted Automata

Krishnendu Chatterjee¹, Laurent Doyen², and Thomas A. Henzinger¹,³

¹ IST Austria (Institute of Science and Technology Austria)
² CNRS, Cachan, France
³ EPFL, 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 [19, 23]. 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 [5, 6]. 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 [12, 16]. 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 \( \text{Sup}, \text{LimSup}, \text{LimInf}, \) 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 [4, 5].

In this paper, we consider probabilistic weighted automata as generators 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). This simple definition combines in a general model the natural quantitative extensions of logics and automata [13, 14, 7], and the probabilistic models of automata for which boolean properties have been well studied [21, 3, 2]. Note that 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 \( \text{LimSup} \) or \( \text{LimInf} \), respectively). While quantitative objectives are standard in the branching-time context of stochastic games [22, 15, 17, 5, 10, 18], we are not aware of any model combining probabilities and weights in the linear-time context of words and languages, though such a model is very natural for the specification of quantitative properties. Consider the specification of two types of communication channels given in Fig. 1. One has low cost (sending costs 1 unit) and low reliability (a failure occurs in 10% of the case and entails an increased cost for the operation), while the second is expensive (sending costs 5 units), but the reliability is high (though the cost of a failure is prohibitive). In the figure, we omit the self-loops with cost 0 in state \( q_0 \) and \( q'_0 \) over \( \text{ack} \), and in \( q_1, q_2, q'_1, q'_2 \) over \( \text{send} \). Natural questions can be formulated in this framework, such as whether the average-cost of every word \( w \in \{\text{send, ack}\}^\omega \) is really smaller in the low-cost channel, or to construct a probabilistic weighted automaton that assigns the minimum of the average-cost of the two types of channels. In this paper, we attempt a comprehensive study of such fundamental questions, about the expressive power, closure properties, and decision problems for probabilistic weighted automata.

First, we compare the expressiveness of the various classes of probabilistic and nondeterministic weighted automata over infinite words. For \( \text{LimSup}, \text{LimInf}, \) 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. We also define the complement \( L^c \) of a quantitative language \( L \) by \( L^c(w) = 1 - L(w) \) for all words \( w \).\(^4\) Note that maximum and minimum are in fact the operation of least upper bound and greatest lower bound for the pointwise natural order on quantitative languages (where \( L_1 \leq L_2 \) if and only if \( L_1(w) \leq L_2(w) \) for all words \( w \)). Therefore, they also provide natural generalization of the classical union and intersection operations of boolean 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 \( \text{LimSup} \)-automata under positive semantics and \( \text{LimInf} \)-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 \( \text{LimInf} \)-automata with positive semantics can be reduced to \( \text{LimInf} \)-automata with almost-sure semantics and to \( \text{LimSup} \)-automata with positive semantics; and because they are not closed under complement, the \( \text{LimSup} \)-automata with almost-sure semantics and \( \text{LimInf} \)-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 \([1, 8]\), we establish some decidability and undecidability results for \( \text{Sup} \), \( \text{LimSup} \), and \( \text{LimInf} \) automata; in particular, emptiness and universality are undecidable for \( \text{LimSup} \)-automata with positive semantics and for \( \text{LimInf} \)-automata with almost-sure semantics, while the question is open for the emptiness of \( \text{LimInf} \)-automata with positive semantics and for the universality of \( \text{LimSup} \)-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 \( \Sigma \) 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.

\(^4\) One can define \( L^c(w) = k - L(w) \) for any constant \( k \) without changing the results of this paper.
Value functions. We consider the following value functions \( \text{Val} : \mathbb{Q}^\omega \to \mathbb{R} \) to define quantitative languages. Given an infinite sequence \( v = v_0v_1 \ldots \) of rational numbers, define

\[
\begin{align*}
\text{Sup}(v) &= \sup\{v_n \mid n \geq 0\}; \\
\text{LimSup}(v) &= \limsup_{n \to \infty} v_n = \lim sup\{v_i \mid i \geq n\}; \\
\text{LimInf}(v) &= \liminf_{n \to \infty} v_n = \lim inf\{v_i \mid i \geq n\}; \\
\text{LimAvg}(v) &= \liminf_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} v_i; \\
\text{For } 0 < \lambda < 1, \text{Disc}_\lambda(v) &= \sum_{i=0}^{\infty} \lambda^i \cdot v_i;
\end{align*}
\]

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 \( A = (Q, \rho, \Sigma, \delta, \gamma) \) where:

- \( Q \) is a finite set of states;
- \( \rho \in \mathcal{D}(Q) \) is the initial distribution;
- \( \Sigma \) 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 \( \rho(q) > 0 \), and \( (q, \sigma, q') \) is an edge of \( A \) if \( \delta(q, \sigma)(q') > 0 \). The automaton \( A \) is deterministic if \( \rho(q_i) = 1 \) 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 \( w = \sigma_1\sigma_2 \ldots \) is a finite (resp. infinite) sequence \( r = q_0\sigma_1q_1\sigma_2 \ldots \) 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 \( \gamma(r) = v_0v_1 \ldots \) 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 \( r = q_0\sigma_1q_1\sigma_2 \ldots \sigma_kq_k \) over a finite word \( w = \sigma_1 \ldots \sigma_k \) 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 \( \text{Val} : \mathbb{Q}^\omega \to \mathbb{R} \), we say that the probabilistic \( \text{Val} \)-automaton \( A \) generates the quantitative languages defined for all words \( w \in \Sigma^\omega \) by \( L^A_\text{Val}(w) = \sup\{\eta \mid \mathbb{P}^A([r \in \text{Run}^A(w) \mid \text{Val}(\gamma(r)) \geq \eta]) = 1\} \) under the almost-sure semantics, and \( L^A_\text{max}(w) = \sup\{\eta \mid \mathbb{P}^A([r \in \text{Run}^A(w) \mid \text{Val}(\gamma(r)) \geq \eta]) > 0\} \) under the positive semantics. For non-probabilistic automata, the value of a word is either the maximal value of the runs (i.e., \( L^A_\text{max}(w) = \sup\{\text{Val}(\gamma(r)) \mid r \in \text{Run}^A(w)\} \) 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 coBuchi 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 $\equiv$ 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 $L, L' : \Sigma^\omega \rightarrow \mathbb{R}$, 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 [8].

**Remark.** We sometimes use automata with weight functions $\gamma : Q \rightarrow Q$ 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. 2s, 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. \( \text{PosSup} \) and \( \text{Assup} \) are reducible to \( \text{DSup} \).

Proof. It is easy to see that \( \text{PosSup} \)-automata define the same language when interpreted as \( \text{NSup} \)-automata, and the same holds for \( \text{Assup} \) and \( \text{USup} \). The result then follows from [7, Theorem 9].

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 \( \delta : S \to D(S) \). 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 \( \alpha \), 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 [20].

Lemma 1. Let \( A \) be a probabilistic weighted automata with alphabet \( \Sigma = \{a, b\} \). Consider the Markov chain arising of \( A \) on input \( b^\omega \) (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 $\beta$ 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 $k_\epsilon$ 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 $l_\epsilon$ 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 $\Sigma$ 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.
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. 3):
   (a) States and weight function. The set of states of the automaton is \( \{q_0, q_1, \text{sink}\} \), with \( q_0 \) as the starting state. The weight function \( \gamma \) is as follows: \( \gamma(q_0) = \gamma(\text{sink}) = 0 \) 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 \( \text{sink} \) with probability 1; and
      (iii) from \( \text{sink} \) state the next state is \( \text{sink} \) 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 \( b^{\omega} \)), 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 \( b^{\omega} \) 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.
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 $\text{POS}L^\text{IM}A\text{VG}$. Consider a $\text{POS}L^\text{IM}A\text{VG}$ $A$ and consider the $b$-Markov chain arising from $A$ under the input $b^\omega$. 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 $\text{ASL}^\text{IM}A\text{VG}$. The automaton $A$ is as follows (see Fig. 4):

(a) States and weight function. The set of states are $\{q_0, \text{sink}\}$ with $q_0$ as the starting state. The weight function is as follows: $\gamma(q_0) = 0$ and $\gamma(\text{sink}) = 1$.

(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 $\text{POS}L^\text{IM}A\text{VG}$, $\text{POS}L^\text{IM}S\text{UP}$ and $\text{POS}L^\text{IM}I\text{NF}$, but not by $\text{NL}^\text{IM}A\text{VG}$, $\text{NL}^\text{IM}S\text{UP}$ or $\text{NL}^\text{IM}I\text{NF}$.

**Proof.** Consider an automaton $A$ as follows (see Fig. 5):

---

Fig. 4. An $\text{ASL}^\text{IM}A\text{VG}$ for Lemma 3.

Fig. 5. A probabilistic weighted automaton ($\text{POS}L^\text{IM}A\text{VG}$, $\text{POS}L^\text{IM}S\text{UP}$, or $\text{POS}L^\text{IM}I\text{NF}$) for Lemma 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:
   (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 PosLimitAvg, PosLimitSup, or PosLimitInf, then it accepts the following language:

\[
L_z = \{ a^{k_1}ba^{k_2}ba^{k_3}b \cdots | k_1, k_2, \cdots \in \mathbb{N}_{\geq 1} \cdot \prod_{i=1}^{\infty} (1 - \frac{1}{2^n}) > 0 \} \cup (a \cup b)^* \cdot a^w;
\]

i.e., \( A(w) = 1 \) if \( w \in L_z \) and \( A(w) = 0 \) if \( w \notin L_z \); the above claim follows easily from the argument following Lemma 5 of [2]. We now show that \( L_z \) cannot be expressed as NLLimitAvg, NLLimitSup or NLLimitInf. Consider a non-deterministic automaton \( A \). Suppose there is a cycle \( C \) in \( A \) such that average of the rewards in \( C \) is positive, and \( C \) is formed by a word that contains a \( b \). If no such cycle exists, then clearly \( A \) cannot express \( L_z \) as there exists word for which \( L_z(w) = 1 \) such that \( w \) contains infinitely many \( b \)'s. 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 \( w_C = a_{i+1}ba_{i+2}b \ldots a_{i+k}b \), and hence there exists a finite word \( w_R \) (that reaches the cycle) such that \( \gamma(w_R \cdot w_C^0) > 0 \). This contradicts that \( A \) is an automaton to express \( L_z \) as \( L_z(w_R \cdot w_C^0) = 0 \). Simply exchanging the average reward of the cycle by the maximum reward (resp. minimum reward) shows that \( L_z \) is not expressible by a NLLimitSup (resp. NLLimitInf).

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

**Theorem 2.** ASLimitAvg is incomparable in expressive power with PosLimitAvg and NLLimitAvg, and NLLimitAvg cannot express all languages expressible by PosLimitAvg.

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

### 3.3 Probabilistic LimInf-automata

**Lemma 5.** NLLimitInf is reducible to both ASLimitInf and PosLimitInf.

**Proof.** It was shown in [7] that NLLimitInf is reducible to DLLimitInf. Since DLLimitInf are special cases of ASLimitInf and PosLimitInf the result follows.

**Lemma 6.** The language \( L_1 \) is expressible by an ASLimitInf, but cannot be expressed as a NLLimitInf or a PosLimitInf.
Proof. It was shown in [7] that the language $L_I$ is not expressible by $\text{NLIMINF}$. If we consider the automaton $A$ of Lemma 3 and interpret it as an $\text{ASLIMINF}$, then the automaton $A$ expresses the language $L_I$. The proof of the fact that $\text{POSIMINF}$ 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.** $\text{POSIMINF}$ is reducible to $\text{ASLIMINF}$.

Proof. Let $A$ be a $\text{POSIMINF}$ and we construct a $\text{ASLIMINF}$ $B$ 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 $\text{PosCW} A^v$ 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 $\text{PosCW}$ is reducible to $\text{ASCW}$ (it was shown in [1] that $\text{ASBW}$ is reducible to $\text{PosBW}$ and it follows easily that dually $\text{PosCW}$ is reducible to $\text{ASCW}$). Let $D^v$ be an $\text{ASCW}$ that is equivalent to $A^v$. We construct a $\text{POSIMINF}$ $B^v$ 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 $\text{PosCW} A^{v'}$ and the $\text{ASCW} D^{v'}$ 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 $\text{ASLIMINF}$ is closed under $\max$ (Lemma 18) and hence we can construct an $\text{ASLIMINF}$ $B$ such that $B = \max_{v \in V} B^v$. Thus the result follows.

**Theorem 3.** We have the following strict inclusion

$$\text{NLIMINF} \subset \text{POSIMINF} \subset \text{ASLIMINF}$$

Proof. The fact that $\text{NLIMINF}$ is reducible to $\text{POSIMINF}$ follows from Lemma 5, and the fact the $\text{POSIMINF}$ is not reducible to $\text{NLIMINF}$ follows from Lemma 4. The fact that $\text{POSIMINF}$ is reducible to $\text{ASLIMINF}$ follows from Lemma 7 and the fact that $\text{ASLIMINF}$ is not reducible to $\text{POSIMINF}$ follows from Lemma 6.

3.4 Probabilistic LimSup-automata

**Lemma 8.** $\text{NLIMSUP}$ and $\text{POSIMSUP}$ are not reducible to $\text{ASLIMSUP}$.

Proof. The language $L_F$ of finitely many $a$’s can be expressed as a non-deterministic Büchi automata, and hence as a $\text{NLIMSUP}$. We will show that $\text{NLIMSUP}$ is reducible to $\text{POSIMSUP}$. It follows that $L_F$ is expressible as $\text{NLIMSUP}$ and $\text{POSIMSUP}$. The proof of the fact that $\text{ASLIMSUP}$ 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 $C$. 

11
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 [11] 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 $v_k > v_j$, 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 $L_\lambda$ can be expressed by a ASBW and hence by ASLIMSUP:

$$L_\lambda = \{a^{k_1}ba^{k_2}ba^{k_3}b \ldots | k_1, k_2, \ldots \in \mathbb{N}_{\geq 1}. \prod_{i=1}^{\infty} (1 - \lambda^{k_i}) > 0\}.$$
It follows from argument similar to Lemma 4 that there exists $0 < \lambda < 1$ such that $L_\lambda$ 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 $A = (Q, q_I, \Sigma, \delta, C)$ be a PosCW with the set $C \subseteq Q$ of accepting states. We construct a PosBW $A'$ 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 $\overline{\delta}(\overline{q}, \sigma, \overline{q'}) = \overline{\delta}(\overline{q}, \sigma, q') = \frac{1}{2} \cdot \delta(q, \sigma, q')$, 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 $q \not\in C$ 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 $\overline{\delta}(\overline{q}, \sigma, q') = \delta(q, \sigma, q')$, 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 $A'$ coincides. Consider a word $w$ such that $A(w) = 1$. Let $\alpha$ 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 $\infty$, the probability that after $k$ steps only states in $C$ are visited is $\alpha$. 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 $A'$, the probability to reach states of $\overline{C}$ 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 $A'(w) = 1$. Observe that since every state in $\overline{C} \setminus C$ is absorbing and non-accepting, it follows that if we consider an accepting run $A$, 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 $A'(w) = 1$, 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 $A_v$ 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 $A_{v_i}$ 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. \textsc{asliminf} and \textsc{poslimsup} are reducible to each other; \textsc{aslimsup} and \textsc{posliminf} have incomparable expressive power.

Proof. This result is an easy consequence of the fact that an automaton interpreted as \textsc{asliminf} defines the complement of the language of the same automaton interpreted as \textsc{poslimsup} (and similarly for \textsc{aslimsup} and \textsc{posliminf}), and from the fact that \textsc{asliminf} and \textsc{poslimsup} are closed under complement, while \textsc{aslimsup} and \textsc{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) \textsc{ndisc} and \textsc{posdisc} are reducible to each other; (b) \textsc{udisc} and \textsc{asdisc} are reducible to each other.

Proof. (a) We first prove that \textsc{ndisc} is reducible to \textsc{posdisc}. Let $A = (Q, \rho_1, \Sigma, \delta_A, \gamma)$ be a \textsc{ndisc}, and let $v_{\text{min}}, v_{\text{max}}$ be its minimal and maximal weights respectively. Consider the \textsc{posdisc} $B = (Q, \rho_1, \Sigma, \delta_B, \gamma)$ where $\delta_B(q, \sigma)$ is the uniform distribution over the set of states $q'$ such that $(q, \sigma, \{q'\}) \in \delta_A$. Let $r = q_0\sigma_1\sigma_2\ldots$ be a run of $A$ (over $w = \sigma_1\sigma_2\ldots$) with value $\eta$. For all $\epsilon > 0$, we show that $\mathbb{P}^B(\{r \in \text{Run}_B(w) \mid \text{Val}(\gamma(r)) \geq \eta - \epsilon\}) > 0$. Let $n \in \mathbb{N}$ such that $\frac{\sqrt[n]{\mathbb{E}[\text{Val}(\gamma(r))]} - \mathbb{E}[\text{Val}(\gamma(r))]}{\mathbb{E}[\text{Val}(\gamma(r))]} \leq \epsilon$, and let $r_n = q_0\sigma_1\sigma_2\ldots\sigma_nq_n$. The discounted sum of the weights in $r_n$ is at least $\eta - \frac{n}{1+\lambda} \cdot (v_{\text{max}} - v_{\text{min}})$. 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 $\eta - \frac{n}{1+\lambda} \cdot (v_{\text{max}} - v_{\text{min}})$, and therefore at least $\eta - \epsilon$. This shows that $L_B(w) \geq \eta$, and thus $L_B(w) \geq L^\text{sc}_B(w)$. 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 \textsc{posdisc} is reducible to \textsc{ndisc}. Given a \textsc{posdisc} $B = (Q, \rho_1, \Sigma, \delta_B, \gamma)$, we construct a \textsc{ndisc} $A = (Q, \rho_1, \Sigma, \delta_A, \gamma)$ where $(q, \sigma, \{q'\}) \in \delta_A$ if and only if $\delta_B(q, \sigma)(q') > 0$, 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 \textsc{udisc} (resp. \textsc{asdisc}) can be defined by a \textsc{ndisc} (resp. \textsc{posdisc}). Then, the result follows from Part a) (essentially, given a \textsc{udisc}, we obtain easily an \textsc{ndisc} for the complement, then an equivalent \textsc{posdisc}, and finally a \textsc{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 \textsc{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.
Table 1. Closure properties and decidability of the emptiness and universality problems.

|            | max | min | comp | sum | emptiness | universality |
|------------|-----|-----|------|-----|-----------|--------------|
| POS SUP    | ✓   | ✓   | ×    | ✓   | ✓         | ✓            |
| POS LIM SUP| ✓   | ✓   | ✓    | ✓   | ✓         | ✓            |
| POS LIM INF| ✓   | ✓   | ×    | ✓   | ✓         | ✓            |
| POS LIM AVG| ✓   | ×   | ×    | ✓   | ✓         | ? (1)        |
| POS DISC   | ×   | ×   | ×    | ✓   | ✓         | ? (1)        |

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

4.1 Closure under max and min

Lemma 16 (Closure by initial non-determinism). **POS LIM SUP, POS LIM INF and POS LIM AVG is closed under max; and AS LIM SUP, AS LIM INF and AS LIM AVG 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 $\sigma$, the successors are from $Q_\sigma$ each with probability $1/|Q_\sigma|$. If $A_1$ and $A_2$ are POS LIM SUP (resp. POS LIM INF, POS LIM AVG), then $A$ is a POS LIM SUP (resp. POS LIM INF, POS LIM AVG) such that $A = \max \{ A_1, A_2 \}$. Similarly, if $A_1$ and $A_2$ are AS LIM SUP (resp. AS LIM INF, AS LIM AVG), then $A$ is a AS LIM SUP (resp. AS LIM INF, AS LIM AVG) such that $A = \min \{ A_1, A_2 \}$. □

Lemma 17 (Closure by synchronized product). **AS LIM SUP is closed under max and POS LIM INF is closed under min.**

**Proof.** We present the proof that AS LIM SUP is closed under max. Let $A_1$ and $A_2$ be two probabilistic weighted automata with weight function $\gamma_1$ and $\gamma_2$, respectively. Let $A$ be the usual synchronized product of $A_1$ and $A_2$ with weight function $\gamma$ such that $\gamma((s_1, s_2)) = \max\{\gamma_1(s_1), \gamma_2(s_2)\}$. Given a path $\pi = (s_0^1, s_1^2, \ldots)$ in $A$ we denote by $\pi \uparrow 1$ the path in $A_1$ that is the projection of the first component of $\pi$ and we use similar notation for $\pi \uparrow 2$. 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($B_i^v$) holds with probability 1. Consider the following set of paths in $A$ $H^v = \{ \pi \mid (\pi \uparrow 1) \in \text{Büchi}(B_i^v) \}$. 

15
Since given $w$, the event $\text{Büchi}(B_1^v)$ holds with probability 1 in $A_1$, it follows that given $w$, the event $\Pi^v$ holds with probability 1 in $A$. The $\gamma$ 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 $C_1^{v'}$ be the set of states $s_i$ in $A_1$ such that the weight of $s_i$ is less than $v'$. Given the word $w$, since $A_1(w) < v'$, it follows that probability of the event $\text{coBüchi}(C_1^{v'})$ in $A_1$, given the word $w$, is positive. Hence given the word $w$, the probability of the event $\text{coBüchi}(C_1^{v'} \times C_2^{v'})$ is positive in $A$. It follows that $A(w) < v'$.

The result follows. If $A_1$ and $A_2$ are $\text{POSlimINF}$, 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 $\text{POSlimINF}$, then $A = \min\{A_1, A_2\}$. The proof is similar to the result for $\text{ASlimSUP}$.

**Lemma 18.** $\text{POSlimSUP}$ is closed under $\min$ and $\text{ASlimINF}$ is closed under $\max$.

**Proof.** Let $A_1$ and $A_2$ be two $\text{POSlimSUP}$. We construct a $\text{POSlimSUP}$ $A$ 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 $\text{POSBW}$ $A_i^v$ that is obtained from $A_i$ by considering all states with weight at least $v$ as accepting states. Since $\text{POSBW}$ is closed under intersection(by the results of [2]), we can construct a $\text{POSBW}$ $A_{12}^v$ 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 $\text{POSlimSUP}$ $B_{12}^v$ 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 $\text{POSBW}$ $A_{12}^v$ 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 $\text{POSlimSUP}$ is closed under $\max$ (by initial non-determinism), it follows that $\text{POSlimSUP}$ is closed under $\max$. The proof of closure of $\text{ASlimINF}$ 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 $\text{POSlimAVG}$ for the language $L_m = \min\{L_a, L_b\}$.
2. There is no $\text{POSlimAVG}$ for the language $L^* = 1 - \max\{L_a, L_b\}$.

**Proof.** To obtain a contradiction, assume that there exists a $\text{POSlimAVG} A$ (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
Lemma 20. \textit{poslimavg} is not closed under \textit{min} and \textit{aslimavg} is not closed under \textit{max.}

\textbf{Proof.} The result for \textit{poslimavg} follows from Lemma 19. We now show that \textit{aslimavg} is not closed under \textit{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 $\textit{DLIMAVG}$ (and hence $\textit{ASLIMAVG}$) for $L_a$ and $L_b$. We show that $L_m = \max(L_a, L_b)$ cannot be expressed by an $\textit{ASLIMAVG}$. By contradiction, assume that $A$ is an $\textit{ASLIMAVG}$ 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 $w^* = w_C \cdot a$ 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 $j_\epsilon$ such that the following properties hold:

1. from any state of $A$, given the word $a^{j_\epsilon}$ 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 $a^{j_\epsilon}$, (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 $\epsilon$ 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 $a^\omega$).

Let $\beta > 1$ be a number that is greater than the absolute maximum value of weights in $A$. We chose $\epsilon > 0$ such that $\frac{\epsilon}{1 - \epsilon - \epsilon} > \frac{1}{\beta}$. Let $j = 2 \cdot j_\epsilon$ (such that $j_\epsilon$ 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 $\tilde{w} = (a^{2j} \cdot b^{3j})^\omega$ and consider a closed recurrent set in the Markov chain obtain from $A$ on $\tilde{w}$. 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 $\epsilon$. 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

$$
(1 - 3 \cdot \epsilon) \cdot \left( \frac{3}{2} \cdot \left(1 - \frac{1}{2} \cdot \epsilon \right) \right) - 3 \cdot \epsilon \cdot \beta = (1 - \epsilon) \left( \frac{3}{2} \cdot \left(1 - \frac{1}{2} \cdot \epsilon \right) \right) - 3 \cdot \epsilon \cdot \beta \\
\geq \frac{4}{5} - 3 \cdot \epsilon \cdot \beta \\
\geq \frac{4}{5} - 4 \cdot \epsilon \cdot \beta \\
\geq \frac{4}{5} - \frac{1}{10} \\
\geq \frac{7}{10} > \frac{3}{5}.
$$

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

\hfill \blacksquare
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** $A_v$ (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 $A_{v_i}$ by assigning weight $-v_{i-1}$ to each accepting state, 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 $A(w.a^n) = 1$ and $A(w.b^n) = 0$ for all $w \in \Sigma^*$. To obtain a contradiction, assume that there exists an **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 $w.b^n$ is at most $\epsilon$, and there exists a number $m_w$ such that $B(w.a^m) \leq \epsilon$ (since $B(w.a^n) = 0$). Hence, we can construct a word $w = b^m.a^m_1.b^m.a^m_2 \cdots$ 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 \(A_0\) 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 \(\min\{v_1 + v_2 \mid v_1 \in V_1 \land v_2 \in V_2\}\). Let the automata constructed be \(A(v_1, v_2)\). Then \(A = \max(v_1, v_2) \cdot A(v_1, v_2)\). Since \(\text{POSlimSup} (\text{resp.}\ \text{AslimSup})\) is closed under \(\max\) the result follows.

\[\text{Lemma 25.} \ \text{POSlimInf and AslimInf are closed under sum.}\]

\[\text{Proof.}\] Given two \(\text{POSlimInf}\) (resp. \(\text{AslimInf}\)) \(A_1\) and \(A_2\), we construct a \(\text{POSlimInf}\) (resp. \(\text{AslimInf}\)) \(A\) 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 \(v_{\min} = \min\{v_1 + v_2 \mid v_1 \in V_1 \land v_2 \in V_2\}\). For \(v_1 \in V_1\) and \(v_2 \in V_2\), for \(i = 1, 2\), consider the \(\text{POS} \\text{CW}\) (resp. \(\text{AS} \\text{CW}\)) \(A_q\) 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 \(\text{POS} \\text{CW}\) (resp. \(\text{AS} \\text{CW}\)) such that \(A(v_1, v_2) = A_{v_1} \cap A_{v_2}\) such an \(\text{POS} \\text{CW}\) (resp. \(\text{AS} \\text{CW}\)) exists since \(\text{POS} \\text{CW}\) (resp. \(\text{AS} \\text{CW}\)) 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 \(\text{POSlimInf}\) (resp. \(\text{AslimInf}\)) 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 \(\text{POSlimInf}\) (resp. \(\text{AslimInf}\)) is closed under \(\max\) the result follows.

\[\text{Lemma 26.} \ \text{AslimAvg is not closed under sum.}\]

\[\text{Proof.}\] Consider the alphabet \(\Sigma = \{a, b\}\), and consider the \(\text{DLlimAvg}\)-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 \(\text{AslimAvg}\). Assume towards contradiction that \(L_+\) is defined by an \(\text{AslimAvg}\) \(A\) 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 \(\infty\), the expected average of the weights converges almost-surely to 1. This implies if we consider the \(\alpha\)-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 \(k_0\) such that from any state \(q\), for all \(k > k_0\) 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 \(k_0\) 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 \(\frac{1}{2}\); (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 \(\frac{1}{2}\); (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 $k_0^\gamma$ satisfies the properties described above for $\gamma$). By construction for $i > 6 \cdot k_0^\gamma \cdot \beta$, the length of $w_i$ is at least $6 \cdot k_0^\gamma \cdot \beta$, 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

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

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 POSSUP and the universality problem for ASSUP are undecidable.

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

Theorem 8. The emptiness problem for ASSUP and the universality problem for POSSUP are decidable.

Theorem 9. The emptiness problem for POSSUP and the universality problem for ASSUP are decidable.
Proof (Sketch). We sketch the main ideas of the proof that emptiness of coB"uchi automata in positive semantics is achievable in EXPTIME and with exponential memory. The proof extends easily to POSLIMINF and to the universality problem for AsLIMSUP.

Emptiness of coB"uchi automata in positive semantics can be viewed as deciding the existence of a blind positive-winning strategy in a stochastic game with coB"uchi objective. It follows from the results of [9] that this problem can be decomposed into positive winning for safety and reachability objectives.

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

Theorem 10. 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 11. 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 known 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. Vincent D. Blondel and Vincent Canterini. Undecidable problems for probabilistic automata of fixed dimension. Theory Comput. Syst., 36(3):231–245, 2003.
4. 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.
5. A. Chakrabarti, L. de Alfaro, T. A. Henzinger, and M. Stoelinga. Resource interfaces. In Proc. of EMSOFT, LNCS 2855, pages 117–133. Springer, 2003.
6. 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.
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. K. Chatterjee, L. Doyen, and T. A. Henzinger. Expressiveness and closure properties for quantitative languages. In Proc. of LICS: Logic in Computer Science. IEEE Comp. Soc. Press, 2009. To appear.
9. K. Chatterjee, L. Doyen, and T. A. Henzinger. Qualitative analysis of partially-observable Markov decision processes. In CoRR: 0909.1645, 2009.
10. K. Chatterjee, M. Jurdziński, and T.A. Henzinger. Quantitative stochastic parity games. In Proc. of SODA: Symposium on Discrete Algorithms, pages 114–123. ACM Press, 2004.
11. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. Journal of the ACM, 42(4):857–907, 1995.
12. K. Culik II and J. Karhumäki. Finite automata computing real functions. SIAM J. Comput., 23(4):789–814, 1994.
13. L. de Alfaro, M. Faella, and M. Stoelinga. Linear and branching metrics for quantitative transition systems. In Proc. of ICALP: Automata, Languages and Programming, LNCS 3142, pages 97–109. Springer, 2004.
14. M. Droste and D. Kuske. Skew and infinitary formal power series. *Theor. Comput. Sci.*, 366(3):199–227, 2006.
15. A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *Int. Journal of Game Theory*, 8(2):109–113, 1979.
16. Z. Ésik and W. Kuich. An algebraic generalization of omega-regular languages. In *Proc. of MFCS*, LNCS 3153, pages 648–659. Springer, 2004.
17. J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, 1997.
18. H. Gimbert. *Jeux positionnels*. PhD thesis, Université Paris 7, 2006.
19. E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games*, Lecture Notes in Computer Science 2500. Springer-Verlag, 2002.
20. J. G. Kemeny, J. L. Snell, and A. W. Knapp. *Denumerable Markov Chains*. D. Van Nostrand Company, 1966.
21. Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.
22. L. S. Shapley. Stochastic games. In *Proc. of the National Academy of Science USA*, volume 39, pages 1095–1100, 1953.
23. 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.